

EVOLUTIONARY ALGORITHMS IN DESIGN

T. Stanković, M. Stošić and D. Marjanović

Keywords: evolutionary computation, evolutionary design, genetic algorithms

1. Introduction

The evolutionary algorithms have come far from being just a drawing board idea or a mind concept. They have found applications as a problem solvers and process backbones in many intriguing fields such as artificial intelligence, informatics (genetic programming), mathematics, artificial life-base creatures, evolvable hardware etc. In engineering domains evolutionary algorithms have during the last fifteen years proven themselves as powerful multiple-objective optimisation tools (MOGA – Multiple-objective genetic algorithms, TOD – topological optimum design, FLP – floor facility layout etc.) but in design theory there are much less published applications [S. Vajna 2003; M. M. Andreasen, 2005] and their full potential there remains still very much hidden. A subgroup in field of evolutionary algorithms, genetic algorithms, are very applicable for describing of processes appearing within real world engineering design and the reasons why is that so will be discussed further in next chapters. At the end a case study will be presented where a functionality of a concept is reached by an optimisation process.

2. Motivation for applying genetic algorithms

When applying genetic algorithms as the problem solving techniques then we're thinking in terms of search algorithms that are, in order to find solution, mimicking the notions from the evolution of the natural organisms. That implies that the solution or the set of solutions to the given problem evolves in time from the feasible solution population by the principle of the survival of the fittest.

The merit of feasibility, or how good the potential solution solves the given problem, is being introduced in a form of the fitness function which then acts as an evolutionary guide. So, the standard engineering optimisation problem of maximizing the fitness function is being put up before the solver algorithm. From various literatures [D. E. Goldberg, 2003; Wood, K.L. and Otto, K.N., 1999; S. Wolfram, 2001] the problem of maximising the fitness function is the same as the problem of satisfying the list of constraints. It is obvious that here exists a resemblance between the algorithm and the design process. Design process itself can be described as the process of satisfying the list of input constraints, or process of moulding of the conflicting requirements [S. Wolfram, 2001; J. Andersson, 2000].

“In every generation, a new set of artificial creatures are created using bits and peaces of the fittest of the old; an occasional new part is tried for good measure” [P. Bentley, 1999]. From lather can be formulated the first of the reasons why to apply evolutionary computation rather than some other computation technique to a particular design stage. As it was shown the search process itself is done by mixing up parts of individual solutions which results in the speed up of the overall process (crossover operator in EA literature), and with the addition of amount of stochastic or randomness (mutation operator in EA literature) the algorithm property of escaping local extreme unfolds. This property of the genetic algorithms is adopted from the natural evolution and it is known in the EA

community [D. E. Goldberg, 2003; P. Bentley, 1999] as the adaptation property. Furthermore, because of its non gradient origins the algorithm is immune to the issues regarding the discontinuities of the search space and the derivability of the fitness function, which is the second very important property of the algorithm. Search spaces built upon customer needs and lists of requirements (like material strength and fatigue, vibrations, aesthetics etc.) in design processes are often discontinuous, the fitness functions are non derivable and multimodal and thus difficult to describe in a form of a calculus based algorithm within the scope of continuous mathematics.

Innovation or findings of new ideas in human design processes is often introduced by various methods and techniques like brainstorming for instance [Wood, K.L. and Otto, K.N., 1999], which is really an attempt to produce an unbiased human performed heuristic search. The same principle exists in natural evolution and in that way also in genetic algorithms by mutation operators [D. E. Goldberg, 2003; P. Bentley, 1999] which stochastically alter feasible solutions. Until proved otherwise the evolutionary algorithms are as the natural evolution is still unbiased systems.

If go further there is an analogy between evolution of the natural organisms and the human design process. Human designs have also evolved during the centuries, as the organisms did, and moreover when devising a new idea the designer is creating it by combination of some previous solutions which are in core the identical principles as the recombination of the genetic material of the living organisms [D. E. Goldberg, 2003].

It seems that the genetic algorithms are very robust problem solving methods, which by imitating the principles from the evolution of natural organisms are in a way also quite similar to human design processes.

3. Evolutionary design

A human conducted design process is defined in literature as a set of technical activities within a product development process which often include the development of new concepts, embodiment and often other non managerial activities [Wood, K.L. and Otto, K.N., 1999]. From that the definition of the evolutionary design immediately follows as a design process that is based on the evolutionary computation. That kind of an iterative process is obviously best and only (nature is excluded) done by the computers because of their ability to process vast amounts of data in reasonable amounts of time.

In literature [D. E. Goldberg, 2003] the field of evolutionary design is described as a mixture of evolutionary biology, computer science, and design. More so, if go further, the field of integral evolutionary design becomes very interesting for design science as it combines evolutionary design optimisation and creative evolutionary design [(D. E. Goldberg, 2003]. The two keywords optimisation and creative and their interloping are indeed the property of every good human design processes. As the natural evolution shapes the living organisms thus enabling them to coexist in their environments, so is the designer shaping and adjusting its designs enabling their existence in constraint bounded design space.

Optimisation is by default a process of making better the existent which, if put in engineering framework, means that the process input data consists of a design model that needs to be improved over a set of design variables. Again, in natural world the optimisation itself is embodied in the core of the evolution, because the natural organisms are in a fact by optimisation improving over time [S. Wolfram, 2001]. That implies that a complete new good solution can emerge as the result of constant improving. If comparing that to the real world enterprises, the sole purpose of powerful design optimisation techniques is the reduction of the product production price which in end results in more competitive product being put to market thus creating an edge over the competition.

There is a one very astonishing phenomenon in the evolution of the living organisms; it is their ability to adapt to the rapid changes in their environments. From design science literature [Wood, K.L. and Otto, K.N., 1999] it's known that adaptive design is a process of adapting the existent subsystem to the requirements of a new product. Adaptive design is more and more present in today's product development because of rapid changes in the market induced by product customization needs. Such designs processes bear risks because of speed in which they need to be done but if conducted carefully they can produce significant gains resulting maybe in creating a complete new product subsystem. From the evolution of the natural organisms' point of view, the adaptation process was and still is a

risky business giving the endless opportunities to the ones who manage to adapt. Evolutionary algorithms are in fact doing the same thing by adapting the potential problem solutions through stochastic operators to the search environment.

Within the creative evolutionary design according to literature [D. E. Goldberg, 2003] two fields exist; the field of evolutionary design on conceptual level which goal is the development of innovative designs concepts by evaluating fitness according to resulting functionality, and the field of generic evolutionary design which is a search for new design morphologies but still regarding the functionality. Morphology or form development is commonly related to the initial stages of design processes in which the performance and overall future functionality (or purpose) of the product is depending mainly on the physical properties of its form. That is often a case in product development processes in the industries like shipbuilding, aerospace industries etc.

Function analysis of the future product is very important part of product development [G. Pahl, W. Beitz 1988] and is done primarily to speed up the overall process, incorporate customer needs and wishes as new product features, and if possible to gain from the knowledge obtained from previous projects. And again, the evolution of natural organisms and genetic algorithms by mimicking it are doing the same by reusing the genetic materials of its previous successful designs in order to create new better designs.

When applying genetic algorithms in design the question regarding heuristics arises. For instance, is the derived design really concept the optimal one, is the result of optimisation the optimal one? The author's thoughts about these kinds of questions regarding his practical engineering backgrounds are that these questions are irrelevant as long the computation process produces good and useful results. The notions about proving that the produced solution is really the global optimum in significantly large search space often surpass the boundaries of common engineering practice.

So there is no doubt that much resemblance exists between evolutionary design [D. E. Goldberg, 2003; P. Bentley, 1999] and the design methodologies described in engineering literature [Wood, K.L. and Otto, K.N., 1999], so why not thrive on it more extensively in the modern product development processes?

4. Case study

The goal was to represent the process of conceptual design as an optimisation process, where the goal of fitness function is the obtaining of functionality by following a set of rules. The result of the conceptualisation would be represented in an early embodiment stage as a solid model assembly in Dassault Systems CATIA V5R16 software. By the inspiration of A-Design [M. Campbell, 1998] for this case study a weighting mechanism was chosen as an example. The difference between A-design [M. Campbell, 1998] and this case study is that here presented algorithm relies predominantly on genetic algorithm and that only one population (population of designs) is being maintained and evolved.

Chromosomes are formed from set of predefined components. Further more to reduce the search space the selection methods for chromosome generation and the methods for performing crossovers are selected accordingly how well they generate concepts. Genetic algorithm was chosen as a drive algorithm because of reasons explained in previous chapters and because of its mapping properties (chapter 4.3).

4.1 Components

The components are for the sake of simplicity nonparametric and can be interconnected at their interfaces. Each of components has one drive and one driven interface, where the driven interface can connect to two components simultaneously but one of these must be elastic component. However, the types of the components that can be interconnected at interfaces differs accordingly to the component properties, the type of motion that they can transmit that is, so within one interface there can be multiple interface points at which the connection could be established. So, the interface points are divided accordingly their purpose as drive interface points (P), driven interface points (G) and interface points reserved for elastic elements (O), and accordingly the type of motion they are transferring.

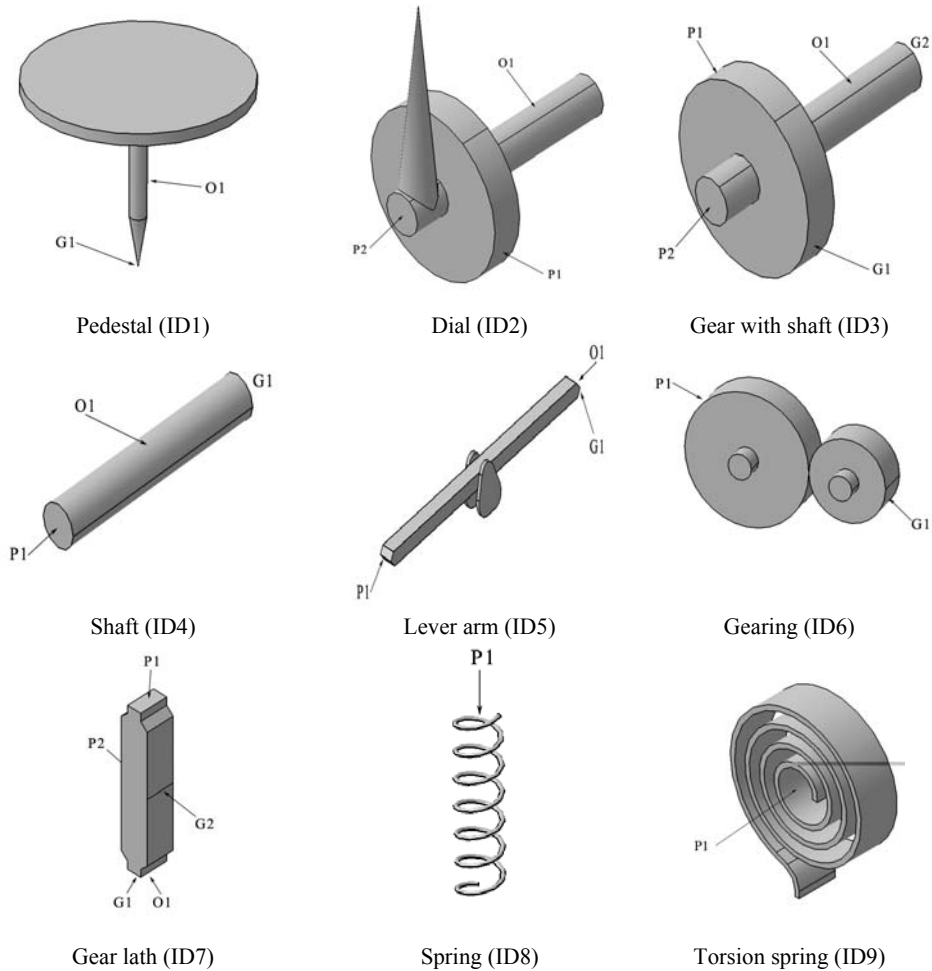


Figure 1. Overview of nonparametric components

The first and the last components of heuristically generated chromosomes are always the same - pedestal and dial (figure 1). If we take for an example Gear with shaft (figure 1), then for the input interface we have two drive interface points (P1) and (P2), one corresponding to gear and one to the shaft. So for (P1) the drive components could be gearing, gear with shaft, gear lath and dial and for (P2) the drive components could be shaft, gear with shaft and dial. Driven components for (G1) are gearing, gear with shaft and gear lath and for (G2) we have shaft, gear with shaft and dial. The (O1) interface is reserved for spring.

4.2 Genotype – generating “G” methods

The genotype is created first by random selection of components with the constraint that the first component is always a pedestal and the last one is always a dial. Afterwards the random selection of spring places is performed. The length of chromosome is random and it varies within user defined boundaries from minimum 3 components to about 40.

In figure 2 a genotype is shown constituting of interconnected components (ID), and to them connected springs marked with (O). There are three methods – “G” methods for generating the genotype one not favouring any of components, second method favouring gears and third favouring

levers. Later to induce the randomness these three methods produce in every generation “new blood” offspring by measuring how well their creations achieved functionally. So the genotype carries a signature of a method that created it.

1 4 3 5 4 4 6 7 3 3 4 5 5 4 2
O O O O

Figure 2. Genotype representation

4.3 Phenotype

The existence of mapping function in genetic algorithms was found to be very suitable for this case study. So if the genotype would be considered as a concept then its phenotype would be early embodiment which corresponds with the literature definitions of phenotype as real world representation of the genetic material carried in genotype [P. Bentley, 1999; D. E. Goldberg, 2003]. To be evaluated the genotype is mapped into phenotype through mapping function. Phenotype of chromosome in figure 2 is presented in table 1.

Table 1. Phenotype representation

1	“”	1	“1.O1.O1.P1”
2	“4.G1.3.P2”	2	“4.O1.O2.P1”
3	“”	3	“3.O1.O2.P1”
4	“”	4	“5.O1.O1.P1”
5	“4.G1.4.P1”		
6	“”		
7	“6.G1.7.P2”		
8	“7.G2.3.P1”		
9	“3.G1.3.P1”		
10	“3.G2.4.P1”		
11	“”		
12	“5.G1.5.P1”		
13	“”		
14	“4.G1.2.P2”		

The mapping function is searching if there can be established a connection between components by checking it with the knowledge base of good connections [M. Campbell, 1998]. So for the strings number 7 and 8 if taken for example have proper connection established by driven interface point “7.P2” connected to the “7.G2” drive interface point. The “7” is referring to the type of component of gene number 8.

4.4 Evaluation and Selection

Evaluation of chromosomes is done on their phenotype representations by applying the following formula:

$$fitness = \frac{\text{number of successful connections}}{\text{number of possible connections}}$$

Two parents are selected for breeding by applying roulette wheel selection. There is no special mating pool so all of the chromosomes have the possibility in creating offspring. The effect of cloning is prevented with parent identity check.

4.5 Crossover –“C” methods and Mutation

Each two parents that are selected for mating produce two offspring, but one of them always dies by selection of a random coin toss. This was done to enhance the randomness. The chromosomes are not identical in size so the crossover point is determined randomly on a chromosome with shorter genotype. There're three types of crossovers – “C” methods: one point, two point and three point crossovers (figure 3).

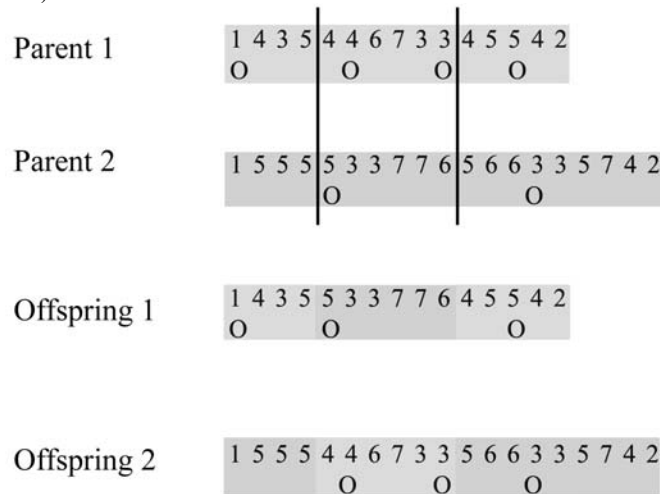


Figure 3. Two point crossover

Which type of crossover is to be performed is at first defined with same probability of $p = 0.33$. Later in process the probability alters by measuring how well their creations have achieved functionally. So the genotype carries the signature of crossover method that created it. Again we have three types of mutations: the mutation of components, the mutation of springs and combined mutation of the both. The selection of method is performed pure stochastically with probability of 0.33. On figure 4 the third type of mutation is presented, where a mutation of component is done to a randomly altering the gene value thus replacing the components. The mutation of spring is done by random exchange of its place.

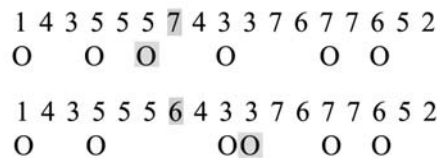


Figure 4. Mutation example

4.6 Algorithm and visualisation

The initial population generated is user defined and it usually consists of 20 – 30 chromosomes. The structure of algorithm is shown below:

Start

Initialisation – generation of population with equal start probability for all “G” methods

Evaluation – of initial population

Loop

Selection – roulette wheel selection of parents

Matting – crossover with equal start probability for all “C” methods
Mutation – mutation of offspring with 1% probability with equal probability of mutation method selection through whole process
Evaluation – with mapping of offspring

- Putting the functional concepts in temporary functional concepts collection (when fitness is equal 1.00).
- Putting the functional concepts that are not identical in final functional concepts collection.

New generation – because of **elitism** 3 concepts make to new generation by random selection from final functional concepts collection. Further more first 5 non-functional concepts are also allowed to live on. To induce randomness user defined number of completely new chromosomes is directly generated by “G” methods and added to population. Rest of free spaces is populated with the best offspring members.

Evaluation – of new generation

- Updating the probabilities for selection of “C” and “G” methods accordingly the fitness of chromosomes they created.

Visualisation - of final functional concepts (user option)

Repeat until the condition is satisfied
Visualisation - of final functional concepts (user option)
End

The phenotypes were visualised in Dassault Systems CATIA V5R16 software (figure 5).

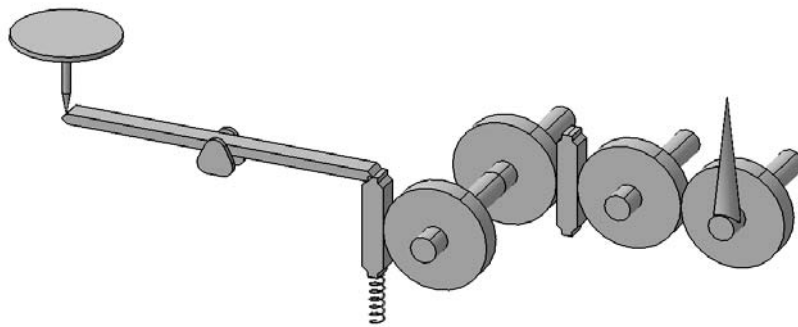


Figure 5. Phenotype visualization

5. Conclusion and future work

In the case study it was shown that the process of conceptual design could be presented as an optimisation process where by following the set of rules for proper connecting and by utilising the inventiveness of evolutionary computation it is possible to create a fully functional weighting mechanism. More so if taken into account the definition of human conducted design process as the process of satisfying the list of input constraints, or process of moulding of the conflicting requirements [S. Wolfram, 2001; J. Andersson, 2000] which is the same as the problem of maximising the fitness function [D. E. Goldberg, 2003; Wood, K.L. and Otto, K.N., 1999; S. Wolfram, 2001] then it was shown that there truly exists resemblance between evolutionary design process and the human performed design process. That implies that evolutionary algorithms and especially genetic algorithm present suitable basis for creation of design tools which should aid designer in conception phases of design processes.

Future work will include the development of automated tools for conceptual designs with parametric components including energy, material and signal flow concepts [G. Pahl, W. Beitz, 1988], topological optimisation tools, genetic evolutionary design tools etc.

References

- Z. Fan, M. M. Andreasen, J. Wang, E. D. Goodman (2005) Towards an Evolvable Chromosome Model for Interactive Computer Design Support, 177.45, ICED 05*
- David E. Goldberg (2003) Genetic Algorithms in Search Optimization and Machine Learning, Addison Wesley Longman, Inc. 25th printing*
- S. Clement, A. Jordan, S. Vajna (2003) The Autogenetic Design Theory – an Evolutionary View of the Design Process, ICED 03 Stockholm, August 19-21*
- J. Andersson (2000) A survey of multi-objective optimization in engineering design, Technical Report: LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, Linköping, Sweden*
- Wood, K.L. and Otto, K.N., Product Design (1999) Techniques in Reverse Engineering, Systematic design, and New Product Development, Prentice-Hall, NY*
- Steven Wolfram (2001) A New Kind of Science, Wolfram Media*
- Peter Bentley (1999) Evolutionary Design by Computers, Morgan Kaufmann*
- G. Pahl, W. Beitz (1988) Engineering Design - A Systematic Approach, Springer Verlag*
- M. Campbell, J. Cagan and K. Kotovsky (1998) A-Design: Theory and Implementation of an Adaptive, Agent-based Method of Conceptual design, Artificial Intelligence in Design '98, Kluwer Academic Publishers, Netherlands*

Dipl. Ing. Tino Stanković Naval Architect
Faculty of Mechanical Engineering and Naval Architecture
Product Development and Design Chair
I. Lučića 5, 10000 Zagreb, Croatia
Tel.: (+385 1)6168 369
Fax.: (+385 1)6168 284
Email: tino.stankovic@fsb.hr
URL: <http://www.cadlab.fsb.hr>