Automated Concept Evolution

Anne Defaweux
Vrije Universiteit Brussel
adefaweu@vub.ac.be

Tobias Grosche
University of Mannheim
grosche@bwl.uni-mannheim.de

Maria Karapatsiou
University of Sheffield
m.karapatsiou@shef.ac.uk

Alberto Moraglio
University of Essex
amoragn@essex.ac.uk

Alex Shenfield
University of Sheffield
a.shenfield@shef.ac.uk

August 30, 2003

Abstract

The Human Based Genetic Algorithm is an extension of the field of Interactive Evolutionary Computation where, in addition to fitness and selection, the user performs all the other genetic operators. The definition of operators is left intentionally loose to stimulate the user's creativity in the evolutionary process. This framework naturally extends to cooperative environments and has great potentiality in fields like marketing, industrial design, creative writing. This paper presents a full design, web-oriented implementation and controlled experiments for story-telling and creation of marketing slogans. This approach raises questions about the balance between humans and machines in creativity, whether the technique is scalable, and what other areas these techniques can be applied to.

1 Introduction

The field of genetic algorithms (GAs) for mathematically defined optimisation problems is well established. Conventional GAs are used in many fields, from scheduling problems to function optimisation. However there is increasing interest in using GA type methods for more ill-defined problems such as design exploration. These problems cannot be easily tackled by conventional GAs due to the subjective nature of their evaluation functions. This has led to growing interest in the use of humans for the evaluation of potential solutions.

This paper starts with an overview of the subject area and goes on to introduce an evolutionary algorithm that can be used by a distributed group of interacting people to produce the solution to a creative problem. We then present the results of our experiments and go on to draw some conclusions about the effectiveness of this approach. Finally we conclude with some suggestions for further work.

2 Overview of the Subject

2.1 Interactive Evolutionary Computation

Interactive evolutionary computation (IEC) is an evolutionary optimisation technique in which the fitness function of the system is replaced by a human user. IEC can be defined as ”the technology that EC optimises the target systems based on subjective human evaluation as fitness values for system outputs” [Tag01]. In IEC the user evaluates the individuals of the population subjectively. This fitness is used to select the parents for the next generation. The next generation is then produced from these parents by the algorithm, using crossover and mutation.

Interactive evolutionary computation is being used in many diverse application areas [Tag01]. These application areas range from automated creation of music [Gra95] to automobile design [Bil94]. The first research area of IEC was the bio-morphs of Dawkins [Daw86]. Other areas of IEC research include facial image generation for criminal suspect recognition [CaJ91], knowledge acquisition and data mining [TeI96], and robotics [Doz01].

An extension of IEC called the Human Based Genetic Algorithm (HBGA) [Kos01][Kos00] has been proposed, in which selection and reproduction are both performed by human operators. In Kosorukoff's implementation, one set of users would perform selection, another set of users would perform mutation, and another set of users would perform crossover.

A version of the HBGA was used in the Free Knowledge Exchange (FKE) project for collaborative web-based problem-solving. The FKE project evolves strings of natural language to arrive at better solutions to problems submitted by its participants [Kos02].

In the FKE system, the population consists of a list of problems created by participants. The system starts with an empty population, to which the first ideas proposed by participants are added. The fitness of each problem in the population is calculated according to the number of participants interested in solving this problem. A problem's fitness determines the probability of its selection, so problems with higher fitness (i.e. in which many people are interested) appear more frequently.

Five problems are presented at a time with the most popular solutions to the problem following each. The user is then invited to either agree with an existing solution (thus increasing the fitness of that solution), or to propose their own answer. Crossover and mutation in the FKE is implicitly encouraged by showing the existing answers, thus allowing the user to use parts of different solutions in their answer (crossover) or to change a solution (mutation). In order to make it easier for people to share their ideas, participants contributions are anonymous.

2.2 Knowledge Management

Parallels can be drawn between organisational genetic algorithms like the HBGA and knowledge management techniques such as random creativity techniques, brainstorming, and ideas banks [Kos00]. Organisational genetic algorithms can also be seen as techniques to enable innovation [Gol99] both automatically (e.g. the use of Genetic Programming to design patentable electric circuits) or by assisting humans in the creative process [Sim91].

Random creativity techniques exist to inspire creative exploration of the solution space [RaM89]. These include random word techniques - in which a random word is used to generate new associations and ideas, and random heuristic techniques - in which heuristics from other fields are adapted to solve a problem. These techniques enable the user to approach their problem from a different angle. This is analogous to mutation.

Brainstorming is a commonly used business approach with similar goals as the random creativity techniques. One definition of brainstorming is "a conference technique by which a group attempts to find a solution for a specific problem by amassing all the ideas spontaneously by its members" [Osb93]. The introduction of computer assisted brainstorming [Hol84] provides the ability to brainstorm asynchronously, to stop and resume easily, and to automate the note taking process. This is analogous to recombination.

Ideas banks use the evolutionary technique of selection to evaluate socially innovative ideas. The main idea behind the Global Ideas Bank (www.globalideasbank.org) is to collect mix and match ideas to achieve "more successful and humane ways of doing things" [Eno98]. This is analogous to selection.

3 Model

This work was inspired by the work of Terry Fogarty attempting to automate a method which explored human and computer interaction in order to support creativity.[Fog03]

3.1 Description of the Algorithm

1. A problem is defined
2. Participants create an initial population by suggesting solutions
3. The next generation is then created using the processes of replication, crossover, mutation and creation
4. If a final solution has not been found the procedure is repeated from step 3 with the new population, else the process is stopped

3.2 Program Architecture

To run our experiment we built an interactive environment. At each time step, a participant has the opportunity to select individuals among the population in order to generate the new population. The selected individuals can be recombined, mutated or replicated to form the next generation. As there are multiple participants for each run they have to be able to interact in a distributed way.

We implemented this through a web interface. When a participant enters the process, they are presented with the list of the individuals which make up the population. They can recombine individuals (using crossover), mutate individuals, or replicate individuals to form the next generation. They could also create new individuals. Since our experiment was text based, it consisted of a button to replicate an individual and a text box allowing the user to create a new individual through a crossover, mutation or creation operator. The data was processed by a PHP script on the web server which added the new individuals to the population. These individuals thus formed the next generation. All the simulation data was kept in a MySQL data base.

The system was synchronised, so that progress to the next generation could only happen once all the participants had finished. Furthermore, we wished the selection process to happen without communication between participants; this was enforced by allowing each user to see only their own choices at each generation. The contribution of the other users was only known once the generation was completed and the individuals become part of the new selection pool.

3.3 Fitness

Our approach to the calculation of fitness can be compared to that of biology. Fitness is the chance of selection. In biology, the computation of the fitness of an individual is done by comparing the offspring of one parent to the offspring of another parent [Mich99]. It is the philosophy of ”the more offspring you have, the greatest fitness you get”. Fitness is therefore an afterthought which captures the outcome of the selection process. Evolutionary Biology also stresses that individuals are being selected on the quality of their phenotype and not on their genotype [Lew74]. In our implementation, we can observe the same approach. We do not have explicit fitness, the fitness is an implicit consequence of the selection process. Individuals (that is in our case ideas) are being selected by the user for reproduction in terms of their semantics (which can be addressed in biological term as their phenotype) but the genetic operators operate on their syntax to produce new individuals characterised by a new syntax (which can be compared to the biological genotype) with possibly a new meaning.

3.4 Genetic Operators

The individuals in the population are ideas. They were represented as strings. The same idea can be represented by many different sentences. This makes it sometimes difficult to define crossover or mutation in a meaningful way or to automate the process. We made the choice to leave the use of the operators to the users. They have the opportunity to recombine or mutate previous ideas manually. We made this choice because we can expect a human user to identify the equivalent sentences even if they are syntactically different. However, we intend the participants in the evolutionary process to use genetic operators. Each operation has to be tagged as creation, mutation or recombination by the user. This tag however is the user’s responsibility. We expect the person using the automated process to play the game. Further approaches should introduce control mechanisms to ensure that a crossover matches the definition. Let us therefore provide the reader with our definition of such operators.

- Replication: This operator is very restrictive. It reproduces an exact copy of the selected individual. No altering of the individual is allowed.

- Mutation: This operator copies an individual, but allows modification of it. This change can alter the syntax or the semantics of the sentence. We propose mutation to happen in a non disruptive way, that is the offspring remains closely related to the parent.

  Example (taken from our experimental simulation):

Parent:	first contact from aliens, gang of drunk (?) people rob star ship ...
Offspring:	first contact from aliens, gang of drunk (?) people rob star ship ... Alien ask the police to help them find the ship in the neighbouring galaxy

  In this example we can see that the previous idea has been extended with a new idea. Semantically, the offspring remains related to its parent.

- Crossover: Crossover is the recombination of two or more individuals in order to form a new individual. Since the individuals of our population consist of text based ideas, we can not always recombine them without some adaptation of the sentences. However, we can talk of a crossover between two or more individuals when the new individual combines elements of all the parents.

  Example (taken from our experimental simulation):

Parent 1:	Political thriller. Terrorist group kidnap George Bush. Car chase
Parent 2:	Group of students meet at a summer school to write a story together, have to play roles but hate each other, find it really boring until things begin to happen.
Offspring:	Group of students meet at a summer school to write a story together, have to play roles but hate each other, find it really boring until they met a group of terrorists who kidnap George Bush

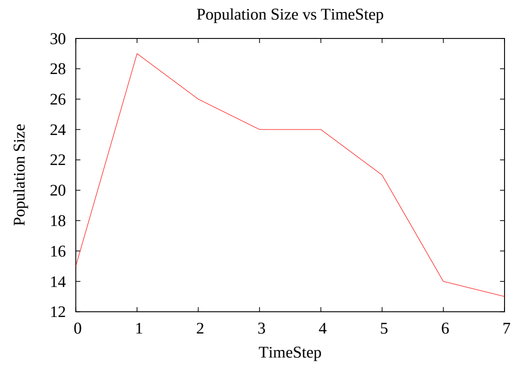

Figure 1: Evolution of the population size in the third run

- Creation: This operator is needed in the initialisation of the process, it allows the participants to create totally new individuals.

4 Experiments

4.1 Experimental Setup

Our experiment was set up to observe the method suggested in the previous section. Our intentions were to test whether our approach of the interactive GA was viable. Two issues we wanted to explore were the unpredictability of human interactions and the infinite search space of potential ideas. We observed whether the population of ideas would converge through the generations towards one commonly accepted idea. If so, we could conclude that the system would emerge a solution for all participants.

We asked for feedback from the participants to evaluate their degree of satisfaction with the solution. We also looked at the final solution produced to see whether it had been propagated through all the generations or whether it was the result of the participant's creativity. When all the participants accept the solution as being of high quality we can conclude that the run was a success.

We observed the sensitivity of our method to two main parameters. The first parameter was the problem posed to the users. We wanted to see whether the outcome of the run was altered if the problem was different. In our first run we posed the problem of creating a story to seven people. In the other two runs we posed the problem of finding a catchy slogan to advertise the university of Parma.

The second parameter we investigated was the selective pressure. Selective pressure on the process was introduced by limiting the time that the users had to generate their contribution to the next generation. Two out of our three runs have no selective pressure, while in the third run the selective pressure was increased throughout the generations. We propose that when selective pressure is used the participants will tend to propagate their favourite ideas and to take less time to explore the idea space. In contrast, using low or non-existent selective pressures will result in a slower convergence of the algorithm. This can be observed by looking at the speed of convergence.

Our experiment was conducted during the EvoNet Summer school 2003 held in Parma, Italy. We made three runs involving seven, four and three persons respectively. In the following table we summarise the parameters for each run.

	Run 1	Run 2	Run 3
Group Size	7	4	3
Topic	Invent Story	Promote Parma University	Promote Parma University
Time constrains	No	No	Yes

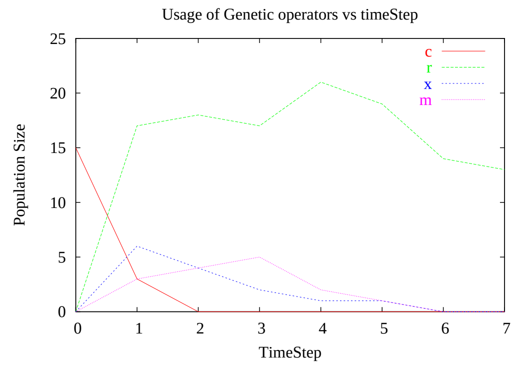

Figure 2: Evolution of the operators use in the third run

4.2 Experimental Results and Conclusions

Our first observation was that all of the runs showed convergence (see Figure:1). After an initial exploration of idea space, which is characterised by an increase of the population size, we can observe a tendency to converge towards a small set of solutions. This result shows that our algorithm succeeds in regulating the population size using ideas of phenotypic selection inspired by nature.

The second result concerns the use of the genetic operators (see Figure:2). After the initial diversification of the population (due to the use of crossover, mutation, and creation), we see an increase in the use of the replication operator leading to convergence. This distribution of diversity is similar to that which you would see in a conventional GA. This result occurred in all the runs of the experiment.

The last result we would like to mention here concerns the user feedback from our participants. After each run the participants were asked to evaluate the result(s) produced. The satisfaction with the result(s) was highest in those runs performed without selective pressure. However, even with selective pressure the users expressed a high degree of satisfaction with the result(s) produced. We can also note that the final solutions in our runs were not present in the initial population.

We can conclude from these results that the human based genetic algorithm system introduced in this paper performs in a similar way to conventional GAs. Another conclusion that can be drawn is that the self-regulation of the population size introduced here is effective, with small group sizes at least. We feel that the system stimulates creativity in the users, whilst also resulting in a consensus developing about the solution to the problem.

5 Further Work

The results presented in this paper were obtained with small group sizes, due to time constraints. To generalise on these results, we propose a new set of experiments using larger groups. Our software implementation allows this due to the distributed nature of the web based system. This could increase the statistical significance of the results obtained.

The operation of the system within large groups is also another interesting area of investigation. One problem that would need to be addressed using large group sizes is how to constrain the population. In group sizes larger than 20 we propose that some method of ranking may be necessary to prevent the user being presented with too greater number of individuals.

Another potential area of investigation is the type of creative task that the system is used to perform. At present our system allows for only text based ideas to form the population, but this could be extended to graphical designs such as fashion designs, architectural designs, and design of urban areas.

In this paper a specific scenario was proposed in which the user has a big role in the evolutionary process, with the computer offering a support medium. Investigation could also focus on whether this is the best balance of human/computer interaction for the support of creativity.

References

[Bil94] Biles, J. A. GenJam: A genetic algorithm for generating jazz solos. In Int. Computer Music Conf. (ICMC'94). Aarhus, Denmark. pp 131-137. 1994

[CaJ91] Caldwell, C. and Johnston, V. S. Tracking a criminal suspect through "f ace-space" with a genetic algorithm. In 4th Int. Conf. on Genetic Algorithms (IC GA'91). San Diego, CA, US. pp 416-421. CA, USA: Morgan Kaufmann Publishers. 1991

[Mich99] Darwinian Dynamics, by Richard E. Michod Princeton University Press 1999

[Tag01] Takagi, H. Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. Proceedings IEEE vol89,no9,pp1275-1296. 2001

[Gra95] Graf, J. Interactive evolutionary algorithms in design. In Int. Conf. on Artificial Neural Nets and Genetic Algorithms. Ales, France. pp 227-230. 1995

[Daw86] Dawkins, R. The Blind Watchmaker. Essex: Longman. 1986

[TeI96] Terano, T. and Ishino, Y. Knowledge acquisition from questionnaire data using simulated breeding and inductive learning methods. In Expert Systems With Applications, vol. 11, no. 4. pp 507-518. 1996

[Doz01] Dozier, G. Evolving robot behaviour via interactive evolutionary computation: From real-world to simulation. In 16th ACM Symposium on Applied Computing (SAC2001). pp 340-344. 2001

[Kos01] Kosorukoff A. Human based genetic algorithm. IlliGAL report no. 2001004. 2001.

[Kos00] Kosorukoff A. Human Based Genetic Algorithm, http://www.hbga.org 2000.

[Kos02] Kosorukoff A. and Goldberg D. Evolutionary Computation as a Form of Organization, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002), Morgan Kaufmann, pp 965-972. 2002.

[Lew74] R.C Lewontin, The Genetic basis of evolutionary change Columbia University Press, NY,1974

[Eno98] An incremental strategy for changing the world. www.globalideasbank.org/wbi/WBI-1.HTML 1998.

[RaM89] Ray M. and Myers R, Creativity in business, Doubleday, NY. 1989

[Osb93] Osborn A.F, Applied Imagination: Principles and Procedures of Creative Problem-Solving.Creative Education Foundation. 1993.

[Hol84] Hollander S, Computer Assisted Creativity and the Policy Process, Masters Thesis, Thayer School of Engineering, 1984

[Gol99] Goldberg D. The race, the turtle and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity. In evolutionary design by Computers, Morgan Kaufmann, San Mateo, CA. 1999.

[Sim91] Sims K. Artificial Evolution for Computer Graphics. Computer Graphics, 25(4), pp. 319-328. July 1991.

[Fog03] Fogarty T. Automated Concept Evolution. Evonet Summer School presentation 2003. (unpublished)