POLITECNICO
MILANO 1863

Philosophical analysis of the role of genetic algorithms in creative applications

ENERLI Damiano
ID: 838120
damiano.enerli@mail.polimi.it

July 14, 2016

Abstract

In this paper I discuss the importance from a philosophical point of view of computational methods, namely genetic algorithms, in their application to creative environments. I also briefly refer to conceptual blending theory and explore how its implementation can be compared to genetic algorithms, underlining differences and similarities and analyzing the role of the procedures in the scope of computational creativity. I will argue that the creative process can be partially formalized with these methods but that this formalization just represents a high potential tool completing a process originated out of the machine.

1 Introduction

My work is developed within computational creativity field, a multidisciplinary area of study that crosses Artificial Intelligence and the arts, based on the will to understand and replicate or augment human creativity through ad hoc computer programs. I'm going to debate if the formalization of creativity is possible and to what extent with respect to two kinds of computational approaches. The structure of the paper is organized as follows.

Since creativity is a highly ambiguous term, I will dedicate section 3 to identify it, building a comfortable framework for my analysis and relating it to the more general environment we call intelligence. Computational creativity is a huge research field, so I will just concentrate the analysis on genetic algorithms, with a brief reference to conceptual blending algorithms, describing in section 4 how they are employed in the setting and which are the main points of their performances. I will show examples to argue for the creative status of the outputs these methods provide, also trying to find some common patterns between the two in order to see if the creative process has some distinctive characteristics that can help us to identify and formalize it. In section 5 I present my concrete stance about the position of the algorithms at issue, arguing that they represent a partial success in computational creativity but that their role is still highly subjugated to the system designer. I will analyze theses and objections similar to the classical arguments used in A.I. debate, re-elaborating them in the context of computational creativity. Finally section 6 concludes the paper, summarizing the analysis and leaving some open questions about the issue.

Next section is just a necessary passage to briefly introduce the basic knowledge about the concepts this paper is focused on: genetic algorithms and conceptual blending theory.

2 Background

2.1 Genetic Algorithms

Genetic Algorithms (GAs) are a specific instance of evolutionary computing techniques, a branch of Artificial Intelligence where the key idea is to exploit some concepts from Darwinian theory of evolution and apply them to computational problems, mostly optimization ones. The basics of evolutionary computation was sketched, among the others, by (Turing, 1950): in the last section Turing exposes his visionary idea about auto-programming machines that evolve by combination of computer programs into child machines in a cycle aimed to reach human intelligence in an automatic way, starting from a learning software instead of a complex one explicitly designed to resemble the mind.

In particular, a GA is an optimization process that produces a population of individuals in the domain space of a given function and combines them according to a model of biological evolution in order to find the global optimum of the function. The idea (not realistic in every context) behind this is that a combination of good individuals produces better ones, where the concept of goodness is measured in terms of the performance with respect to the given function we want to maximize. In the general case, individuals are simple bit strings manipulated by three operators:

- Mutation: random changes of some bits in the individuals.
- Recombination or crossover: fusion of two individuals into one child, where the particular way of mixing two bit strings is implementation dependent.
- Selection: passing from a generation to the next one only the best elements survive, according to some fixed criteria.

From the starting population, another called offspring is obtained. Bad individuals are discarded and the procedure is repeated in cycle until some kind of convergence is reached, i.e. most individuals in the final population are equivalent to the best solution.

2.2 Conceptual Blending

Conceptual blending is an attempt to formalize in a theory the subconscious process of the blending of structures from two or more mental spaces, projected into a new space which inherits aspects from the input ones but also shows a new autonomous structure. One of the first formulations of the theory is described in (Tunner and Fauconnier, 1995), where the authors refer to the term "mental space" to mean a relative small concept, whose structure is often recruited from different domains, a brick in the knowledge of the subject.

Conceptual blending is a cognitive operation that applies to everyday life, involved in reasoning, imagination, linguistic expressions. Computational implementations of this processes are employed in frame-based systems (where a frame is intended to model a unit of knowledge, a concept described by attributes and predicates) to obtain creative results in some specific settings, like metaphors creation (where a source space is partially mapped or "blended" onto a target one to obtain an impressive phrase to express a concept). These algorithms work by combining two or more frames in one blended space that inherits predicates and attributes selected from the input spaces according to some policy, but also has new features that originates from the combination of the heterogeneous inputs. The result is a standalone concept and is not intended to give information about the source spaces.

3 Creativity and Thinking

Creativity is strictly related to intelligence. Before discussing this sentence I should point out what creativity and intelligence are meant to be in the scope of this paper, and how we can say that a behavior belongs to the former and to the latter. We know that answers to these questions are difficult and blurred, so I'm just going to recall some issues about creativity and the standard way we can classify a formal procedure as creative. To reduce the complexity of the concept, (Newell, Shaw and Simon, 1959) assumed a restricted point of view and gave a definition in terms of criteria about creativity in the context of problem solving. I list here the four rules they identified to help us to label a problem-solving program as creative or non-creative:

- **novelty** of the product of the thinking (for the thinker alone or for his whole culture),

- **unconventionality** of the thinking process,
- **persistence** of the process and high motivation that it requires,
- **imprecision** of the initial definition of the problem, that requires the ability to formulate the problem before solving it.

This definition seems to be clear, yet its application causes some problems, because the distinction between creative and non-creative is not always as sharp as we could expect and a method does not have to satisfy all criteria at the same time to obtain the label of creative. But what I want to underline is the key concept emerging from the statements: to evaluate the creativity in a problem solving environment, we can do more than considering just the solving process; we have to give importance at all the components, i.e. the problem itself and the results obtained, beyond the procedure. This fact complicates the situation because it introduces borderline cases hard to judge: let's consider for example trying to solve a problem with a known approach never used for it. This may produce unseen positive results we can reasonably label as creative because of their surprising novelty.

During the paper I will refer to these criteria as a valid starting point to reason about the potential of the algorithms I mentioned in the introduction.

Resuming the theme I introduced with the beginning sentence of this section, I want to put at the attention of the reader the parallelism between creative thinking and intelligence in the general sense, intended as the ability to think, and underline how it is difficult some times to distinguish creativity from what we retain to be standard reasoning. For example (Newell, Shaw and Simon, 1959) argue that creative problem solving does not need its own theory distinct from the general problem solving one because the former is just a peculiar instance of the latter that occurs when the problem to solve presents specific features as a high difficulty and novelty, that enforce the solver to adopt a kind of reasoning characterized by an high degree of freedom.

One could object that creativity is not just related with problem solving, so I suggest to analyze if there is a difference between this kind of approach and creativity as we intend it in artistic frameworks for example. We would not say that a piece of art, let's say a painting, is the solution to a problem. We'd rather talk about inspiration, internal need to represent something, to communicate. This does not prevent us to model these necessities as a kind of problem. What is really difficult to collocate is the illumination that triggers this needs. This obstacle could be due to our lack of knowledge or to an intrinsic feature of the illumination itself. If we had a precise knowledge about inspirations, we could easily try to formalize the process, but for now I postpone the discussion to the end of this section and focus on the problem solving setting.

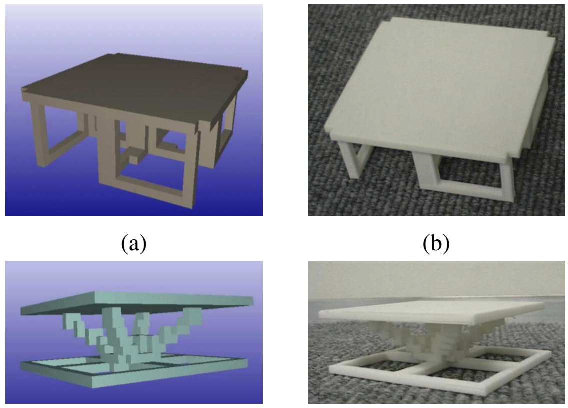
Problem solving can be seen as solution discovery and there is a well known difference between discovery and invention. Columbus discovered America, someone invented the wheel. But are we really capable of invention in the sense of creation, or do we just take hints from the world, and find solutions to non-existing-before (or not yet taken into account) problems? In other words, did someone really created the wheel out of nowhere, or did he "just" applied creative thinking on how to solve the problem of transporting things looking at the world around him? The answer to this specific case seems to be obviously the latter, but still we could never deny the essence of that invention, degrading it to the status of discovery, because an object with that specific function never existed before. Thus I suggest to consider, along with discovery and invention, a third concept of creation, meaning that a thought (of any kind) springs from the mind as an autonomous concept. So we may ask if creativity is an expression of intelligence, a kind of unconventional reasoning or if humans are actually capable of creation as defined above. The source of thoughts is a controversial debate and reminds me of the theological objection in (Turing, 1950) but I'm not interested in discussing here if a creative thought is or not an exclusive function of the soul. The purpose of the dissertation is to show how creativity and general thinking are divided by a really fine line and thus in the attempt of their implementation they're exposed to the same critics, also depending on the point of view assumed by the analysis.

The scope of this paper lies outside the kind of creation described above and is closer to the point of view on creativity exposed by (Simon, Langley and Bradshaw, 1981): they brilliantly expose the theory that scientific discovery is strictly concerned with problem solving and take advantage of an example computer program, called BACON.4, capable of re-discovering through problem solving techniques some important theories like Kepler's laws, starting from a bunch of data. In the paper they build up a clear distinction between strong and weak methods in scientific development: strong methods are used in well-known domains, consist of powerful techniques applied in a systematic way and lie in the the domain of "normal" science; weak methods have uncertain results, proceed by trial and error and are a distinctive sign of scientific inquiry, because they're applied to unexplored domains where ad hoc methods are not available, by definition. I will use these terms later in the discussion to compare humans' ways of proceeding with machines' one.

4 Applications and results of the algorithms to creative environments

A GA, as said, is an optimization procedure: how can it be applied in the computational creativity setting? And, more generally, can optimization be considered creative? In section 3 we've seen how to appreciate creativity in a problem solving context. Here I argue the answer to the second question is yes, at least partially, but first I have to show an approach to reply to the first question.

There are lot of examples of applications of these algorithms in design, where people have to search for arrangements of structures with mathematical constraints given by the functionality of the designed object: the design of the shape of a train respecting aerodynamic equations is just one of this cases. As I explained GAs proceed by manipulating bit strings we call individuals. The main point to apply this kind of algorithms for example in design, is the meaning we give to individuals: they can just represent numbers or we can set up a suitable mapping between the sequence of bits and a structure in the physical world, encoding somehow the constraints the structure is subject to. This enables the transposition of the design process to a problem of search in the space of the representations. A great effort in this sense is represented by (Hornby, 2003), where the concept of generative representation, against the non-generative one is introduced. Since the domain space could be huge and full of useless solutions, the idea is to facilitate the search process exploiting hierarchical reuse of organizational units. A generative representation is one in which encoded design can reuse elements of its encoding in the translation to an actual design. This allows to achieve plausible results without affecting the automatic generation of individuals performed by the evolutionary algorithm. An example by this work is the generation of novel designs for a table shown in picture 1.



(a) (b) (c) (d)

Figure 1: Evolved tables in simulation and reality

Back to the second question, I asked if an optimization process could actually be creative. Now I'm going to exploit the parallelism between the biological evolution and the way a GA works to support my thesis. Would the reader say that nature is creative? We don't know why life exists, but since it does, all the organisms try to preserve themselves as an intrinsic instinct and here comes the natural selection. It's not difficult to imagine a big change occurred in the past to the world's climate: living organisms that fitted to the environment during the years, suddenly become obsolete. Evolution is the process that produces individuals that fit to the new world. Here I see a problem (survive to the changes), a solving approach (working by mutation, recombination and selection) and a peculiar solution never seen before (the new organisms that fits to the new environment). Focusing on the results, we can spot the creativity of the nature in every day life (shapes, movements, colors of living organisms) as the continuous stages of an optimization process, the one carried out by the nature itself, maximizing a fitness function (the likelihood of surviving) under some constraints imposed by the environment.

With this argument I'm not trying to say that every optimization process should be considered creative, but I'm arguing that in some particular conditions search for the optimum may produce something new and unexpected.

One could object that GA approach is just a simplified model of the evolution. I completely agree with this and I'm not claiming GAs have a fully creative behaviour by themselves, but I argue that the same reasoning I applied to evolution results applies as well to the table example: I'm pretty sure the shapes of the final solution were a pleasant surprise for the author, otherwise if we could imagine the hundreds of possible shapes before the execution of the GA we would have never looked at the problem of design automation.

Another objection I want to discuss is one that could be formulated like this: GAs (and evolution) are just using randomness to find the optimum of some function, this is absolutely not creative. There is for sure a random component in the algorithm, but this does not mean the whole process is blind. As (Goldberg, 1999) points out there is an high creative potential hide in GAs because of their structure: while the three operators they adopt are completely useless if considered singularly, together they are a source of continuous improvement (mutation + selection) and innovation (recombination + selection). There is no creativity in a static world, and this is why we need random changes and random combinations of individuals. Selection is the key concept to make the process goal driven, in some sense: by letting only the fittest survive, the algorithm is running with the purpose of producing fitter and fitter individuals.

One great example of implementation of GAs in a context we surely consider creative is given by the research field of John R. Koza, one of the fathers of the genetic programming paradigm, which is an extension of GAs where individuals correspond to computer programs, in a setting that is very close to the already cited one of auto-programming machines from Turing. Koza is actually involved in studies for the use of genetic programming as an automated inventor: a machine for creating new and useful patentable inventions. His website[1] is rich of what they call "human-competitive" examples of invention in the field of electronic components, produced with methods generally described in (Koza, Keane and Streeter, 2003).

So we've seen how GAs, which are basically a problem solver, can be applied in some creative settings. Another approach to computational creativity I want to briefly talk about is the implementation of conceptual blending. In (Li, Zook, Davis and Riedl, 2012) they show the results of their system which is producing gadgets in fiction for a more general A.I. purpose, stories generation. The system works by combining two structures representing concepts from the real world to obtain a gadget in this specific case which has attributes and predicates obtained through a projection from the source spaces. They describe the importance of three procedure not clearly defined by previous implementations: (1) the selection of input spaces, (2) the mechanism for projection and (3) a sufficiency condition. I want to focus on the third one: in the example the generation of the gadget has to be the solution to a particular problem raised by the course of the story: they need something weird fact to happen and the system has to find a gadget to make it possible. It starts with blending spaces selected with some policy until the gadget has the properties required by the context.

A context is highly needed in this framework and in conceptual blending in general because the blended space can assume different connotations depending on the features we consider. For example we are able to understand the meaning of a blended linguistic expression like a metaphor only if we can contextualize it, otherwise we don't know the aspects that expression is trying to bring to our attention.

5 Discussion on the philosophical role of GAs

GAs and conceptual blending algorithms are on two different levels. They were thought with completely different purposes: optimization for GAs and direct encoding of a creative model for the others; still I want to highlight and discuss some common aspects.

Both algorithms have a combination procedure: the former are meant to combine elements from the same kind in a process that resembles organisms reproduction; the latter are trying to unify structures that represent heterogeneous concepts in the real world. Here I spot two natures of creativity, an innovation-driven one and a generative one. In both algorithms the concept of goal is of crucial importance. In a GA the goal is defined as the maximization of a fitness function, while in conceptual blending implementation I mentioned,

[1] http://www.genetic-programming.com/inventionmachine.html

the goal is expressed by means of the needs of a context the final result is required to satisfy.

If we consider for example the arts, it is difficult to imagine human creativity originates with a precise objective. We'd rather prefer to talk about inspiration. For scientific discovery seems to be the same: "Methodologists of science sometimes hint that the fundamentality of a piece of scientific work is almost inversely proportional to the clarity of vision with which it can be planned" (Simon, Langley and Bradshaw, 1981, 5). This is quite reasonable because we think about creativity as some non-traditional way to operate. If we set a clear objective function the course of action becomes bounded in a sense.

In other words, I am considering again Lady Lovelace's objection: how could a machine operate out of the box, if an algorithm is a scheme itself by definition? I think this is a fundamental point in my discussion. I indirectly refused a slightly different formulation of this objection (which declaims a machine cannot take us by surprise) in the previous section, showing how an algorithm can produce unexpected results, but the objection in this new form seems to be irreproachable. If we consider the process, randomness is not sufficient to say a machine is behaving creatively because it is just part of an algorithm, defined by encoded instructions like others. Our machine is just doing what it is supposed to do, even if we're not able to predict its output.

I support the fact that what is exposed in section 4 is just valid at the final result level. If we remember the 4 criteria for creativity, they involved also the process and the problem formulation. I'm not stating a creative result is not enough to call creative who or what produced it. I argue that a creative input from the external is necessary to achieve remarkable results with a machine, and the key of this input is in the formulation of the problem. GA is just a way of operating, a standard process that resembles trial and error procedures (trying solution, discarding bad ones and combining good ones), typical of innovative processes, but there could be other algorithms suitable for this environment, implementing what we called "weak methods". My thesis is that the formulation of a problem in computational terms represents the essence of creativity in the contexts I exposed. I think the main concept is perfectly expressed by these words: "We humans seem to reserve the word creative as a category that goes beyond innovative, but in what way? I would suggest that the word creativity is reserved for people and things that are able to transfer knowledge from one domain to another" (Goldberg, 1999, 7). That's exactly what I mean: if we want to implement a system for automatic design of a table we have to be able to transfer the physical domain into the computational one. We need knowledge about both domains and we have to think in a non traditional way to overlap them in a new land, a cross domain, where an abstract bit sequence can actually assume a meaning in the physical world.

That is the space of the representations and it seems to be the same concept expressed by conceptual blending theory. (Tunner and Fauconnier, 1995) call it the third space in the "many-space model": a space where two domains are blended together to form something that is in the same time more and less than the sum of the source spaces.

Conceptual blending is clearly a way our brains work when producing something creative. But an implementation to automatize it, like the one I described in previous section, is highly dependent on the semantic of the context: we have to understand it and give it to the machine.

An essential theme I have to deal with is the role of the designer of a system like the described ones. In particular considering the context of the previously cited inventor machine makes me bring to surface a curious question: to whom (or what) should we bestow the property of a patent developed like that? It would be not so easy to pay the license fees to a machine so the common sense should suggest us the programmer deserves the profit, in the same way we reason with respect to standard software. The matter here seems to be subtler because we usually don't talk about software as an inventor. In the traditional paradigm we are the users and the machine has no active role in the human-computer interaction. In computational creativity the setting is overturned. They talk about the machine as the subject. Maybe it's just a way to turn on the news and wreak new havoc in the already highly debated Artificial Intelligence world. Or maybe they're actually claiming the role of machines is changing. My opinion is that the kind of procedures at issue has really something different with respect to traditional algorithms, because it introduces autonomous way for development of solutions that gives the algorithms an intermediate position between human and his tools. GAs set up simulations in a development environment that resembles the natural world, modeling a paradigm of innovation. Conceptual blending models the way humans establish links between different domains, which is in my opinion one of the greatest expressions of intelligence. I think we have good tracks to follow but we shouldn't forget they are just models, closed in a box, and humans are still the essence and the interpreter of the meaning of the results these models produce.

Another way to look at the issue is to recall the argument of consciousness against the strong artificial intelligence, stating it in a suitable form for the scope of this paper: a machine could never be aware of the fact it invented something. That's why we have to map a real world problem to a fitness function to maximize when using GAs, and a context dependent stopping criteria when implementing a conceptual blending framework. Humans have to fix the problem because machines are not really able to understand when they reach a creative result. This argument is a very strong one, since the concept of consciousness is quite difficult to point out. Humans could be guided as well by some sort of objective function in their creative works, even in arts we can imagine a subtle function to be optimized by the subject, like the need to express oneself. Our knowledge about human processes is very far from being complete, and we can't precisely state where all our ideas come from. Maybe we'll never be able to tell, so we have to exploit as much as we can what we're able to do that machines are not, and vice versa.

Machines take us by surprise because they reason in a way which is quite unnatural for us, they can produce things we may not be able to imagine, but they are forced to work with mathematical abstractions humans are able to produce. So we could take advantage of this diversity instead of trying to avoid it.

6 Conclusion

I expressed my point of view about the relations between creativity, optimization and problem solving. I described two kinds of approaches to computational creativity setting in order to extrapolate basic operations common in the creative process, showing examples of their application and discussing the philosophical role of the machine in each context. I supported the thesis that creativity can be at least partially described as manipulation of pieces of information, thus formalized in an algorithm and encoded in a machine. Finally I argued that If we are able to formulate problems and apply suitable mappings between machine's language and the real world, we can exploit these powerful algorithms in order to obtain results and innovations that humans alone could hardly imagine, but on the other hand machines are no more than calculators without highly creative designers.

Summarizing my analysis I conclude that machines with respect to the state of the art are lying in a limbo between the status of mere tools and the one of creative entities and I invite the reader to wonder if there are actual reasons to talk about "human-competitive results" while considering the product of a human-computer process. In other words, maybe we don't need human-resembling machines, able to inspire themselves, but we should exploit the immense power given by the combination of the efforts.

This is not meant to be an exhortation to leave the studies aimed to understand and encode mental processes. We have a lot to learn about the mind and we don't know if it is wholly susceptible of formalization, but in the meantime we can continuously improve our machines and make available always more powerful tools for creativity and all other aspects of life. If an insuperable border line between the essence of a tool and the consciousness really exists, our seek for knowledge will maybe bring it to light.

References

Goldberg, D. E. (1999). Genetic and evolutionary algorithms in the real world. *Urbana, 51*, 61801.

Hadamard, J. (1954). *An essay on the psychology of invention in the mathematical field.* Courier Corporation.

Hornby, G. S. (2003). *Generative representations for evolutionary design automation* (Doctoral dissertation, Brandeis University).

Koza, J. R., Keane, M. A., & Streeter, M. J. (2003). Evolving inventions. *Scientific American, 288*(2), 52-59.

Li, B., Zook, A., Davis, N., & Riedl, M. O. (2012). Goal-driven conceptual blending: A computational approach for creativity. In *International Conference on Computational Creativity* (Vol. 10).

Newell, A., Shaw, J. C., & Simon, H. A. (1959). *The processes of creative thinking.* Santa Monica, CA: Rand Corporation.

Simon, H. A., Langley, P. W., & Bradshaw, G. L. (1981). Scientific discovery as problem solving. *Synthese, 47*(1), 1-27.

Tunner, M., & Fauconnier, G. (1995). Conceptual integration and formal expression. *Metaphor and Symbol, 10*(3), 183-204.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*(236), 433-460.

www.genetic-programming.com/johnkoza.html (last access July 2016).