DOI: http://dx.doi.org/10.7551/978-0-262-32621-6-ch146

Computation and Scientific Discovery? A Bio-inspired Approach

Ioan Muntean[1]

[1] The *Reilly Center for Science, Technology and Values*, University of Notre Dame, Notre Dame, IN 46556
imuntean@nd.edu

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022

Abstract

Philosophers argue that scientific discovery is far from being a rule-following procedure with a general logic: More likely it incorporates creativity and autonomy of the scientist, and probably luck. Others think that discovery can be automatized by some computational process. Based on a concrete example of Schmidt and Lipson Schmidt and Lipson (2009), I argue that the bottom-up discovery is computable and that both aspects of creativity and autonomy can be incorporated. The bio-inspired evolutionary computation (genetic algorithms) are the most promising tool in this respect. The paper tackles the epistemology of applying a evolutionary computational and genetic algorithms, to the process of discovering laws of nature, invariants or symmetries from collections of data. Here i focus on more general aspects of the epistemology of evolutionary computation when applied to knowledge discovery. These two topics: computational techniques applied in science and scientific discovery taken separately are both controversial enough to raise suspicions in philosophy of science. The majority of philosophers of science would look with a jaundiced eye to both and ask whether there is anything new to say about discovery and computers in science. This paper is a first stab to the philosophical richness of computational techniques applied to the context of discovery. I discuss the prospect of using this type of computation to discover laws of nature, invariants or symmetries and appraise their role in future scientific discoveries.

Is scientific discovery an algorithmic process?

I argue in this paper for a deeper connection between bio-inspired computation and the process of scientific discovery. Based on new concrete results of Schmidt and Lipson 2009, I infer here some epistemological consequences for using evolutionary computation in scientific discovery.

Knowledge is central to virtually all advanced forms of life; discovery and learning characterize us as a species as well as other higher order animals. We discover in order to survive and adapt. Science is just another specific form of knowledge in which data and experiments play a fundamental role in conjecturing hypotheses about the world. If discovery is probably intrinsically linked to our evolution as a whole, scientific discovery played a central role only in the evolution of humanity in the last four centuries or so (a good turning point is the work of Francis Bacon and its influence during the "Scientific Revolution").

How do we infer laws and generalizations from data? How do we discover new models and theories? Are creativity and autonomy of scientists major cognitive faculties that define and shape science, or, on the contrary, is scientific discovery just a process of following rules, methods and algorithms? The nature of scientific discovery, together with, arguably, artistic creativity, moral decision making and religious experience are among those faculties that define us as humans better than anything else.

These fundamental questions about the nature of scientific discovery are germane to the discussion of artificial scientific discovery. As I link the process of discovery to human life as a species, it is germane to investigate philosophically the paths to an artificial process of scientific discovery. Can we create machines that would perform activities deemed by many as "human-only"?

The broader scope of this paper is to investigate the possibility of a cooperation between the human scientist and the artificial discoverer. I based my argument on a specific

Two approaches to scientific discovery

For the purpose of this paper, the scientific endeavor can be divided between the context of discovery and the context of justification. The distinction can be traced to H. Reichenbach's early works but it is very clearly expressed in Reichenbach (1949). After introducing the infamous distinction, Reichenbach discussed the reliability of a logic and epistemology of discovery. Epistemology is a rational reconstruction of a thought process. In a common interpretation, there is no epistemology of discovery, which is basically a subjective and irrational process: P. Duhem, E. Mach, K. Popper, R. Carnap, C. Hempel, or R. Brainwaite for different reasons deemed discovery as irrelevant when compared to the context of justification. The iconoclastic view of scientific discovery as a "happy guess" or "mystic presentiment" is discussed in Koestler (1959). In a different key, M. Curd and Th. Nickles interpreted Reichenbach's discovery-justification distinction as not excluding an epistemology of discovery. There is an epistemology of discovery, with or without a logic of discovery. So epistemology is much a broader area than logic in this specific framework.

For both these contexts it is relevant to ask this question: is science based on deductive logic, induction or on heuristics? A similar question can be asked about the nature of discovery: is scientific discovery algorithmic, nearly algorithmic or, on the contrary, is it non-discursive, not re-constructible, non-reproducible, singular, a "Eureka"-like mental episode? Is discovery merely a psychological process with no epistemological significance (when compared to the process of justification, for example)?

There are perhaps two main programs in the philosophy of scientific discovery. First, there is a strong program aiming to formulate a general logic for scientific discovery, to encompass all scientific discoveries under one formalism Simon (1973); Hanson (1958). The connection proposed by Langley, Simon, Bradshaw and Zytkow (Langley et al., 1987) between discovery and the heuristic search procedure falls under this strong program. But this strong program felt in disgrace for several reasons and was replaced with a weaker program that gives up the idea of a formal and general *logic* of scientific discovery and tackles the *epistemological* aspects of particular discoveries Nickles (1980b,a); Meheus and Nickles (2009).[1] Here epistemology can be both descriptive and normative and more attention is paid to non-formal and non-logical epistemological aspects of discovery: heuristics, search, risky generalizations, etc. This weak program is more sensitive to the specific conditions of the discovery and of the specific nature of the discoverer. One can ask two questions:

> **(1)** *How do individual scientists, with their limited cognitive faculty, discover new scientific theories? By following a set of rules or by sheer creativity?*
>
> **(2)** *How new theories can be discovered by scientists aided by computers, by Artificial Intelligence systems, or any system other than individual scientists?*

The descriptive epistemology of scientific discovery can answer **(1)** by a careful analysis carried within history of science. Here the discoverer is an individual—the lone genius of Kant, or any scientist experiencing the "Eureka" moment of discovery. We face here a "dilemma of explanation" if we have a theory about scientific discovery as algorithmic Nickles (1980b); Wartofsky (1980):

> **(3)** DILEMMA OF ALGORITHMIC EXPLANATION*: The dilemma is then: either the theory succeeds, and the concept of discovery is explained away, or reductively eliminated—or the theory fails, and discovery remains unexplained.*

I emphasize here the novelty of question **(2)**. First, it does not have a complete answer in the history of science, because the computer-aided scientific discovery or discoveries made by large teams of scientists have a shorter history–when compared to scientific discoveries made by individuals. When the discoverer is a collaborative team, a whole scientific communities, a team working with computers, or a set of computational processes, or all these working together, rationality and creativity may well have radically opposite meanings. The answer to **(1)** does not entail an answer to **(2)**, and vice-versa. Communities, computers or other entities may discover scientific laws, patterns, or theories by an altogether different mechanism than human scientists do, with or without explaining away creativity.

This paper aims to answer **(2)** and show in what sense there is "a third way" in Wartofsky's dilemma **(3)**. The way in which computers and artificial intelligence are used in science may elucidate the normative part of this epistemological approach, but we do not need to equate computational techniques with rational agents, machines, number crunching devices, etc. I do not identify rationality with logic, irrationality with creativity, or machines with logic and creativity with humans only. When used in the scientific discovery, the computational technique comprehends several elements such as: creativity, rule-following procedures, logic etc. I think there is something interesting for philosophers to study about discovery and about computation, taken separately or when computation is directly applied to scientific discovery.

The skeptic against computers used in areas in which human knowledge reigns may raise important questions: Are current computational techniques versatile enough to reproduce, and eventually enhance, the process of scientific discovery? If so, which type of computation is the most promising? And moreover, is this process going to slowly replace humans with machines, even in the process of discover? I reckon that all these questions are attractive from a philosophy of science point of view. It is even more contentious whether a computational process can discover solutions to problems that humans (alone) cannot discover.

In focusing on the epistemology of scientific discovery and the possibility of its algorithmic reconstruction, the current approach is more local and partial: I focus on a specific bottom-up approach to discovery: inferring invariants and laws of nature from large sets of data, and on a specific type of computation: the evolutionary computation implemented by genetic algorithms.

The philosophy of computation in science follows the debates on the relation between data, phenomena, models and theories. For the purpose of my analysis, two contexts of computational science are relevant, both inspired by recent discussions on applying science/applied science Morrison (2006); Bod (2006); Boon (2006). (a) The computational technique starts from a scientific theory and move towards the data: here computation is the *application* of a theory or a "top-down" approach. Or (b), computation is a *heuristic* tool that starts from data and builds a theory in a "bottom-up" approach. Each of these two approaches may have their own specific computational turns: computational techniques used in one may or may not be as revolutionary as they seem in the other. Differentiating these two contexts may help the philosopher argue for the novelty of the epistemological aspects of (b) when compared to (a).

Evolutionary Computation and the Bottom-up Approach to Theory-building

On different occasions, philosophers and scientists alike pointed out to a major difference among two types of scientific reasoning (Th. Kuhn, L. Laudan, among others). On one hand, one has the rule-based reasoning in which new theories or models are inferred from a set of rules. The system of abstract rules is used to solve problems. The rules in general are content-neutral and in the ideal situation they can be applied to virtually any new set of data. On the other hand, one witnesses case-based reasoning in science. Th. Kuhn and K. Popper asked incessantly: is science applied by following rules? Exemplars are solutions to previous problems that scientist learn during their scientific education and solve future puzzles based on an "acquired similarity"Kuhn (1962).

[1]For reasons why the strong program failed, see Curd (1980); Laudan (1980).

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022

Scientists try to make a new phenomenon fit to one or more previous phenomena.

A relevant step forward is to show that neither science, nor computation can be reduced to a succession of rule-following procedures. If we restrict computers to rule-following, then there is little chance, if any, that computational techniques can be useful in scientific discovery. Some philosophers of science have analyzed computation as heuristics device in the *discovery* of new theories. Here concrete results are less notable than in (a). Computer scientists try to use algorithms to discover laws of nature, invariants or patterns in data at least since the 1970s: the most known are the packages DENTRAL, EURISKO, GLAUBER, STAHL and BACON Simon et al. (1981); Mitchell (1997); Waltz and Buchanan (2009). They are designed for a theory-building procedure, when the scientists have little or no idea about how the theory is supposed to look like Keller (2003); Galison (1996); Langley (1979); Barberousse et al. (2007); Pennock (2000, 2007). There is a similarity between the Case-Based reasoning suggested by Kuhn and similar AI techniques used in problem-solving Bod (2006). A case-based procedure always retrieves cases whose problem is similar to the problem being solved. The procedure discussed is data-oriented as opposed to rule-based processing. Computers mimic frequently the process of learning, which is not completely based on rules. According to Bod, data-oriented procedures in computers are similar to the way scientists explain new phenomena "by maximizing derivational similarity between the new phenomenon and previously derived phenomena" Bod (2006).

Therefore, neither scientists nor computers follow strict rules, but reuse previous results in order to solve new problems. For Bod, previous patterns of derivations are learned and accumulated, not phenomena in themselves. Rules are always present, but they are complemented with corrections, normalizations, exemplars derivations, adjustments, all stored and reused from previous cases. In context (b), in the data-oriented discovery process, something else is needed than rule-following procesures. This takes us a step towards answering **(1)** and solving dilemma **(3)**. As P. Langley *et al.*, P. Thagard (1998) and L. Darden (1998) have argued, bringing in computation into the discussion on scientific discovery should majorly boost philosopher's interest in discovery. But, as my argument goes, the nature of computation plays a central role in dismissing **(3)** as a false dilemma and answering **(2)**. I show that once we move to a new type of computation, **(3)** is based on some false assumptions if we give up the very restrictive concept of algorithm and adopt a general concept of computation.

Based on the concrete case study (Schmidt and Lipson, 2009), I show in what sense creativity and rationality can in fact go hand in hand in the case of genetic algorithms applied to scientific discovery. The answer lies in the artificial life metaphor used by Schmidt and Lipson. Computational results in this context are still rare, but as my argument goes, this case cuts deeper into the computational epistemology. More concretely, in the following two sections I address these questions:

> **(4)** *What are the epistemological consequences of using evolutionary computation in scientific discovery?*
>
> **(5)** *Is evolutionary computation the appropriate type of computation for the process of discovery?*

Evolutionary Computation

Roughly speaking, computer algorithms were born based on three distinct analogies: algorithms as "formal proofs", algorithms as "learning processes" and algorithms as "searching procedures for optimality". The latter inspired the area of evolutionary computation, as the paradigm for optimality is an organism optimally adapted to its environment.

How is "search" related to "life"? In the 1930s, S. Wright (1932) interpreted a biological species as a system that evolves in time by exploring a multi-peaked landscape heuristic of optimal solutions to a "fitness problem" . The operation of optimization of search which is typically performed by an algorithm can mimic a living organism that over a long period of evolution fits the environment. On the other hand the process of adaptation and evolution is not smooth.

Organisms are subjected to *random* mutations, too. Taken the biomimetic strategy on step forward: Is it a good idea to add randomness to algorithms? There are several types of *stochastic* algorithms each of them being more or less *biomimetic* in their nature. Biomimetic strategies are widely used in robotics and artificial intelligence, but they are almost ignored by philosophers.[2] Are they useful when applied to scientific discovery?

After a serendipitous proposal by A. Turing in the early 1950s, Evolutionary Computation (*EC*) was rediscovered and reinvented at least ten times before the 1980s (Fogel, 1998). The milestone is J. Holland's work (1975). Following Turing and von Neumann, Holland was able to see the potential of using the knowledge on natural adaptation process to improving search techniques and applied the principles of natural selection directly to problem-solving algorithms. One fundamental difference, not available in Turing's time, is that selection occurs better at the level of population, not at the level of individuals.

The elements of a genetic algorithm

Genetic algorithms are iterative procedures of *searching* for the optimal solution to a problem P. They are based on the metaphor of biological processes in which organisms: (a) *non-consciously* adapt to the "environment" P and (b) are selected by a *supraindividual* mechanism such as selection.[3] The question is whether we can generate algorithms in the same way organisms are created through evolution.

Genetic algorithms start from a given number of initial individuals randomly distributed in a given space, called the initial population. The genetic algorithm transforms individuals, each with an associated value of fitness, into a new generation by using the principles of survival-of-the-fittest, reproduction of the fittest and sexual recombination and mutation. Similar to Wright's landscape, the genetic algorithm finds "the most suitable" or the "best so far" solution to the problem by breeding individuals over a number of generations.

The procedure can be stopped by a termination condition: when the sought-for level of optimality is reached or when all

[2] On the concept of biomimetics, see Srensen (2004); Muntean and Wright (2007).

[3] I take here algorithms as abstract, mathematical objects, whereas programs as their concrete instantiation on a machine. A sensitive difference is between genetic algorithms, genetic programming and genetic strategies. See Jong (2006).

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022

the solutions converge to one candidate. The fitness function estimates the fitness to breeding of individuals in accordance with the principle of survival and reproduction of the fittest:

- Better individuals are more likely to be selected than inferior individuals.
- Reselection is allowed.
- Selection is stochastic.

The genetic algorithm ends with a *termination condition* that can be the satisfying of a success predicate or completing a maximum number of steps. The success predicate depends on the user's choice and can be deemed as a pragmatic criterion. The winner is designated at the "best-so-far" individual as the result of the run.

Here is an abstract implementation of a genetic algorithm:

```
[1] produce an initial population of
               individuals

    [2] WHILE `termination' not met do
       [3] evaluate the fitness of all
               individuals

       [4] select fitter individuals for
               reproduction
       [5] produce new individuals

       [6] generate a new population by
       inserting some new
       good individuals and
       by discarding some
       `bad' individuals

       [7] mutate some individuals

    [8] ENDWHILE

[9] Call the individual(s) which satisfy
       the `termination' condition
       the ``best-fit-so-far''
```

Case study: (Schmidt and Lipson, 2009): Distilling laws and invariants

To show that "algorithmic explanation" and "creativity" are *not* mutually exclusive in **(3)**, I use as an example of computation applied directly to science the result reported in *Nature* (Schmidt and Lipson, 2009). M. Schmidt and H. Lipson have showed how symbolic regression based on evolutionary programming can be used in discovering *natural*, *non-trivial* and *meaningful* invariants in physics.[4] Their algorithm searches over the infinite possible ways of modeling data to find the best and most useful expression available given (i) a set of data; (ii) a termination condition and (iii) a set of evolutionary path. It starts with a set of individuals which can be equations, models and scientific heuristic methods of search—not necessary mathematical objects. Each individual is tested against a bank of experimental data. Many individuals do not make

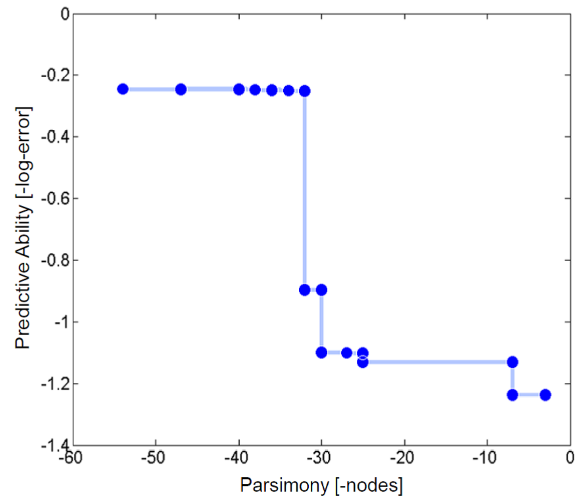

Figure 1: The Pareto front with two "cliffs".(Schmidt and Lipson, 2009, supplementary online materials)

sense mathematically or do not meet some consistency criteria, so they are discharged. Some may fit the data better than others. The software saves these individuals for "breeding", cross-combining a 'father' with a 'mother'. It is claimed that over hundreds of thousands of generations, some extremely fit individuals emerge.

Schmidt and Lipson approached scientific discovery as being data-driven. They started from a set of measured, uninterpreted set of data representing the position, velocity and acceleration of a lab experiment or a virtual system (generated by another algorithm). The method used, the "symbolic regression", is not new at all, but here the program searches for both the form and the parameters of an equation that model a given set of experimental data. They have discovered not only analytic functions from empirical data, but structures which are highly relevant to physical sciences: Hamiltonians, Lagrangians, laws of conservation, symmetries, and other invariants.

Schmidt and Lipson adopted the balance between two objectives: the predictive power and the complexity/parsimony of each candidate. By calculating the "Pareto front" of the dependence predictive ability versus parsimony, Schmidt& Lipson found that there are two cliffs where predictive ability jumps rapidly at some relatively small increase in complexity.

The Epistemology of Discovery with Evolutionary Algorithms: Risks and Advantages

One knee-jerk reaction to applying computation to science is: what is so philosophical about (yet) another tool used by scientists? Although we are nowhere near an "end of computation", the philosopher would not directly infer from its success, its epistemological relevance. Many scientific tools are successful in science, but philosophically inept, and *vice versa*. Although not yet successful, I claim that this case study is worth of a philosophical scrutiny as it sheds some

[4]The package is EUREQA, a software based on evolutionary algorithms Lab (2009).

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022

light on some concepts such as: creativity, rule-following, knowledge production, etc. The procedure addresses some very general epistemological issues of scientific discovery. The knowledge-production in this case study uncovers interesting aspects of the scientific discovery. I frame the following epistemic "aspects" both as problems and as novel features of the scientific discovery based on evolutionary algorithms. The direct application of evolutionary computation to scientific discovery shows how productive bio-inspired algorithms can be. The most attractive feature of evolutionary computation is its ability to "explore" the logical space of solutions, even those which remains unconceived to the mind of the scientist. But the whole process is not totally automatized and the algorithm is not fully autonomous. The human scientist imposes her own meta-rules on the algorithm. On the other hand, because every solution is a model better or worse adapted to data, the bio-mimetic aspect of this example is clear: scientific models adapt to the data and create populations of solutions such that each individual contributes to the adaptation function of the population. After running the algorithm as suggested by Schmidt and Lipson, the scientist is able to explore the "tip of the iceberg", i.e. the best adapted in so far individual from a multitude of previous generations of solutions. The unconceived alternative models, although not direcly present in the final solution did influence it if they were part of the intermediate generations of solutions. I relay the epistemological aspects of the genetic algorithms used in scientific discovery to the various aspects of artificial life. A stronger connection, not endorsed here, would connect knowledge in general to evolution, the are being the evolutionary epistemology. The main part of my argument is that the face of scientific discovery "as we know it" may change radically once evolutionary computation is involved in the process of discovery. I list here several aspects of this "upward epistemology" that is still nascent but very enticing philosophically.

Stochasticity versus scrutability of solutions

Genetic algorithms can be stochastic or not, depending on the mutation operator occurring in step (7) or by selecting the individuals for reproduction in step (5) (in Table 1). An algorithm becomes deterministic if exactly one parent is *identically* reproduced or if two parents are combined without adding or losing information based solely on their fitness. Genetic algorithms are stochastic in two major respects: both the operation of selection and reproduction are random. That means the results (offspring) are not direct results of the input data (the parents).

The crossover operator takes two individuals, the parents, and produces two new individuals, the offspring, by swapping substrings of the parents. Randomly choosing two parents to mate or randomly deleting or adding information from the parents will make the algorithm stochastic. Mutation is a background redistribution of strings to prevent premature convergence to *local* optima.

Weak individuals may survive "by luck" and fit individuals may not be drawn to reproduce. The advantage of a random mutation is that at least some populations, ideally a few only, could escape the traps which deterministic methods may be captured by, and end up with an unexpected and novel result. For very complex problems, this biomimetic procedure can output results which are definitely not accessible to deterministic algorithms if a delicate balance between the mechanism of selection that decrease variation and those that increase variation (mutation) has been achieved.

Because the scientist can control this mutation operator and its frequency, the output of such a discovery algorithm is not traceable by humans. At the limit, the solution of such an algorithm may be inscrutable to humans. It is also the case that for any run, because of the stochastic element, the best individuals are not guaranteed to be selected, and the worst are not eliminated. One can say that the algorithm favors the best and marginalizes the unfit. The selection is not entirely "greedy" in the search space. We do not need to associate creativity to such a random process. As I show before, human element is not totally eliminated in this case. The creativity is blind in this case, similar to mutation in biological populations.

Rules, laws and metarules

The evolutionary algorithms do not follow a set of rules in respect of the discovery of new laws or invariants. As the case study suggests, the process of discovery is here ruled by the metarules of evolution as well as the method used to decide about the fitness function and the termination condition.

For simple laws and invariants, genetic algorithms are easily outrun but Turing machines. But given the complexity of current science, deterministic algorithms may well be worn out as aiding tools to optimality. Although this may sound speculative, let us assume that science evolved toward increasingly complex representations. Maybe the good-old-days of simple, beautiful laws of nature are gone. What if were not going to encounter beautiful laws such as:

$$F = ma;\, F = k\frac{m_1 m_2}{r^2};\, E = mc^2;\, R_{ij} - \frac{1}{2}g_{ij}R = 8\pi G T_{ij}$$

anywhere down the road? For the time being, weve been lucky enough that our best laws of nature could have been fit on a "T-shirt", as it were. How do we discover more and more complex laws of nature? We are limited by conceivability and our limited resources to recognize patterns and regularities may become overtaken by the increasingly complex set of data. Time in which we could deduct laws from phenomena without any epistemic extenders may be over. More and more complex data are collected. Cosmologists, neuroscientists, sociologists, political scientists do not have the luxury to infer their laws from laws as simple as Newton's or Einstein's. What if, from now on, the would-be laws of nature wont fit even a football banner? We need to brace up for more and more complex scientific representations...

Social science, biology, suggest that we may want to drop completely the ideal of laws of nature in their simplest and purest form. In some historical cases, pre-existing theories and the accompanying mathematics were not "already there" when a major discovery in science occurred: contrast this with the received view on the "unreasonable effectiveness of mathematics". We may even need to reconsider the concept of universal laws of nature, existing independent of the way we collect and simulate data.

Now, here is a brighter perspective. Even if the good old days are bygone, there are new ways of coping with increasing complexity in the form of invariants, regularities, laws of nature and alike. Distributive knowledge in science is a tempting idea. Science made by communities of scientists, labs, research programs may steadily replace science

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022

made by individuals. The other possible path suggested by Humphreys is a collaborative work between computers and humans. Maybe we have to face the fact that science is getting closer to the limits of our knowledge, we as limited individual brains. Philosophically put, science is getting closer to the conceivability limit of possibilities.

Triviality versus meaningless

In Schmidt and Lipson's approach, there is a problem of triviality and meaninglessness of solutions. For almost any set of empirical data there are uncountable invariants or conserved quantities, some of them being trivial, some being meaningless. The main task in this case is to find a non-trivial invariant of the system that also can be interpreted as having a meaning. Schmidt and Lipson proposed a criterion based on decomposability: the candidate equations should predict connections between dynamics of the *subcomponents* of the system. This is done by pairing the variables and looking for natural behaviors of parts of the system. More precisely, the conservation equations should be able to predict connections among derivatives of groups of variables over time, relations that we can also readily calculate from new experimental data. Ultimately, their procedure was able to infer the *optimal* form of the double pendulum Hamiltonian by avoiding trivial and meaningless solutions. Schmidt and Lipson included a human decision maker in their algorithm who stops the search process at certain time and imposes the constraints of the symbolic regression such as: "naturalness", "interestingness" or "meaningfulness".

Interpretation versus understanding

Bootstrapping can also be used to infer laws for more complex systems. Results about simpler systems can be used to infer equations for more complex systems. From a statistical analysis, Schmidt and Lipson inferred that terms that are frequently used and are more complex have also *meaning*. For example, trigonometric terms represent potential energy, squared velocities are associated to kinetic energy. The main claim of Schmidt and Lipson is that these terms are ready for a human interpretation:

> These terms may make up an 'emergent alphabet' for describing a range of systems, which could accelerate their modeling and simplify their conceptual understanding. [. . .] The concise analytical expressions that we found are amenable to human interpretation and help to reveal the physics underlying the observed phenomenon. Many applications exist for this approach, in fields ranging from systems biology to cosmology, where theoretical gaps exist despite abundance in data.
>
> Might this process diminish the role of future scientists? Quite the contrary: Scientists may use processes such as this to help focus on interesting phenomena more rapidly and to interpret their meaning Schmidt and Lipson (2009).

The outcome of such an algorithm can help *in the future* with understanding scientific results which are not strictly speaking discovered by humans. The operation of distilling laws from data does more than generating symbols, be them complex expressions of conserved quantities or equations. Similar to numerical simulations, "the results are not automatically reliable" and more effort and human expertise is needed to decide what results are reliable and which are not (Winsberg, 2009). But in this case the computation is more than a tool or a technique because it makes the results intelligible to the human scientist and the question whether the method can be truly creative is up for grabs.

Path dependency versus global solutions

Genetic algorithms compensate some of their drawbacks by their effectiveness in global search. Remember that they maintain a population of solutions which are constantly updated with fitter new individuals and hence avoid local optima. For a certain complexity of the search space, a genetic algorithm has a better chance to find the global optimum. This changes radically the epistemological aspects of genetic algorithms. They are very efficient in solving "hard problems" where little or nothing is known about the sought-for structure and when discovering new structures trumps the process of evaluating existing knowledge.

The case study underscores well this problem of any evolutionary computation: its path dependence. Even the non-trivial and meaningful solutions are not unique! The procedure does not produce a single set of solutions, but a set of *candidates* for the analytical solutions. It is known that any complex problem has a number of local maxima in the landscape of solutions with different fitness values. At different runs of the simulations, different populations can converge to different maxima. The human discoverer will always reach only one solution, whereas a set of genetic algorithms running on the same initial population will end up with different optimal solutions. This is a direct consequence of the fact that similar to biological evolution, the process is non-deterministic. As it was recently argued, this leads to a non-modular functionality of the algorithms and hence to a limited understanding of the operations (Kuorikoski and Pyhnen, 2013). The only aspect which is etymologically accessible to the scientist is comparing results and deciding the best fit. But the way we achieved that results is inscrutable to the scientist. Previous generations and the evolution itself is in many cases too complicated to follow or alternatively, too stochastic to constitute a justification *per se*. As we cannot trace the proof of the algorithm and replicate it, this is in direct analogy with the way we can run the tape of life and every time a different rational agent will emerge as the "better-to-fit". The principles of recombination, selection, and mutation are basically "operators" in the algorithm to generate new individuals.

Turing versus non-Turing; abstraction versus implementation

This aspect is more speculative and reflects a general attitude towards computation in general. Why is evolutionary computation so special? Some theoretical results suggest that evolutionary Turing machines may are more expressive than Turing machines—at an abstract level.[5] The so-called "Turing Evolutionary machine" is more expressible than an ordinary Turing machine, and its output can converge to the output of an universal Turing Machine. More importantly, the evolutionary Turing machine can solve the TM-unsolvable halting problem using non-algorithmic means (Eberbach, 2005). Generalizing computation to a non-Turing aradigm would

[5] I will follow here mainly Eberbach (2005). See also Pudlk (2001).

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022

provide novel and unexpected epistemological results. Unlike Turing machines, the theory of Evolutionary Turing Machines is relatively unknown to the philosophical community. Eberbach has showed that evolutionary computation can be non-algorithmic, can evolve non-recursive functions and that an evolutionary Turing machine can solve the TM unsolvable halting problem of a UTM. “They are specific metaalgorithms (i.e., algorithms operating on other algorithms) with no restriction on their domain and some (rather historical) restriction on evolutionary algorithms that they have to be probabilistic, population-based, and using fitness function.” Eberbach (2005). Practical implementations of evolutionary computation are approximations of Turing machines and they are heavily restricted to time and resources of concrete implementations.

Conclusion

With its “upward epistemology”, evolutionary computation applied to discovery is a promising new tool for future scientific projects. Evolutionary computation and genetic algorithms in particular, anticipate the way scientific methodology and knowledge may look in a couple of decades. And the philosopher of science cannot wait for the foreseeable moment of the informational singularity when artificial intelligence will compete with humans. My humble philosophical prediction is that evolutionary computation, or some more “evolved” offspring of it, will be there at the “singularity” party - if there shall be any.

References

Barberousse, A., Franceschelli, S., and Imbert, C. (2007). Cellular automata, modeling, and computation.

Bod, R. (2006). Towards a general model of applying science. *International Studies in the Philosophy of Science*, 20(1):5–25.

Boon, M. (2006). How science is applied in technology. *International Studies in the Philosophy of Science*, 20(1):27–47.

Curd, M. (1980). The logic of discovery: an analysis of three approaches. In Nickles, T., editor, *Scientific discovery, logic, and rationality*, volume 56. Springer.

Darden, L. (1998). Anomaly-Driven theory redesign: Computational philosophy of science experiment. In Bynum, T. and Moor, J., editors, *The Digital Phoenix*. Blackwell, Cambridge.

Eberbach, E. (2005). Toward a theory of evolutionary computation. *BioSystems*, 82(1):1–19.

Fogel, D. B., editor (1998). *Evolutionary Computation: The Fossil Record*. Wiley-IEEE Press, 1 edition.

Galison, P. L. (1996). Computer simulations and the trading zone. In Galison, P. and Stump, D. J., editors, *The Disunity of science : boundaries, contexts, and power*. Stanford University Press, Stanford Calif.

Hanson, N. R. (1958). *Patterns of Discovery: An Inquiry Into the Conceptual Foundations of Science*. Cambridge University Press.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, second edition bradford books, 1992 edition.

Jong, K. A. d. D. (2006). *Evolutionary Computation*. MIT Press: A Bradford Book, Cambridge MA, 1st edition.

Keller, E. (2003). Models, simulation, and 'Computer experiments'. In Radder, H., editor, *The Philosophy of Scientific Experimentation*, pages 198–215. University of Pittsburgh Press.

Koestler, A. (1959). *The Sleepwalkers: A History of Man's Changing Vision of the Universe*. MacMillan, Los Angeles.

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. University Of Chicago Press, 3rd (1996) edition.

Kuorikoski, J. and Pyhnen, S. (2013). Understanding nonmodular functionality: Lessons from genetic algorithms. *Philosophy of Science*, 80(5):637–649.

Lab, C. M. (2009). Eureqa.

Langley, P. (1979). Rediscovering physics with BACON.3. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*.

Langley, P., Bradshaw, G. L., Simon, H. A., and Zytkow, J. (1987). *Scientific discovery : computational explorations of the creative processes*. MIT Press, Cambridge, Mass.

Laudan, L. (1980). Why was the logic of discovery abandoned? In Nickles, T., editor, *Scientific discovery, logic, and rationality*, volume 56. Springer.

Meheus, J. and Nickles, T., editors (2009). *Models of Discovery and Creativity*. Springer, 1st edition. edition.

Mitchell, S. D. (1997). Pragmatic laws. *Philosophy of Science*, 64(4 Supplement):S468–S479. Journal Article.

Morrison, M. (2006). Applying science and applied science: Whats the difference? *International Studies in the Philosophy of Science*, 20(1):81–91.

Muntean, I. and Wright, C. D. (2007). Autonomous agency, AI, and allostasis a biomimetic perspective. *Pragmatics & Cognition*, 15(3):485–513. Journal Article.

Nickles, T., editor (1980a). *Scientific discovery, case studies*, volume 60. D Reidel Pub Co.

Nickles, T., editor (1980b). *Scientific discovery, logic, and rationality*, volume 56. Springer.

Pennock, R. T. (2000). Can darwinian mechanisms make novel discoveries?: Learning from discoveries made by evolving neural networks. *Foundations of Science*, 5(2):225–238.

Pennock, R. T. (2007). Models, simulations, instantiations, and evidence: the case of digital evolution. *Journal of Experimental & Theoretical Artificial Intelligence*, 19(1):29–42.

Pudlk, P. (2001). Complexity theory and genetics: The computational power of crossing over. *Information and Computation*, 171(2):201–223.

Reichenbach, H. (1949). *Experience and Prediction: An Analysis of the Foundations and the Structure of Knowledge*. Literary Licensing, LLC.

Schmidt, M. and Lipson, H. (2009). Distilling Free-Form natural laws from experimental data. *Science*, 324(5923):81–85.

Simon, H. A. (1973). Does scientific discovery have a logic? *Philosophy of Science*, 40(4):471–480. ArticleType: research-article / Full publication date: Dec., 1973 / Copyright 1973 Philosophy of Science Association.

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022

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

Srensen, M. H. (2004). The genealogy of biomimetics: Half a centurys quest for dynamic IT. In Ijspeert, A. J., Murata, M., and Wakamiya, N., editors, *Biologically inspired approaches to advanced information technology*, volume 3141 of *Lecture notes in computer science*, page 496. Springer, Berlin; New York. Book, Section.

Thagard, P. (1998). Computation and the philosophy of science. In Bynum, T. and Moor, J., editors, *The Digital Phoenix*. Blackwell, Cambridge.

Waltz, D. and Buchanan, B. G. (2009). Automating science. *Science*, 324(5923):43 –44.

Wartofsky, M. (1980). Scientific judgement: Creativity and discovery in scientific thought. In Nickles, T., editor, *Scientific discovery, case studies*, volume 60. D Reidel Pub Co.

Winsberg, E. (2009). Computer simulation and the philosophy of science. *Philosophy Compass*, 4(5):835–845.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *Proc of the 6th International Congress of Genetics*, volume 1, page 356366.

Downloaded from http://direct.mit.edu/isal/proceedings-pdf/alife2014/26/892/1901333/978-0-262-32621-6-ch146.pdf by guest on 16 August 2022