Adaptation Writ Large: An Essay in Memory of John H. Holland

Stephanie Forrest
University of New Mexico
Santa Fe Institute
forrest@cs.unm.edu

Melanie Mitchell
Portland State University
Santa Fe Institute
mm@pdx.edu

1. INTRODUCTION

Professor John H. Holland passed away recently in Ann Arbor, MI, where he had been on the University of Michigan faculty for over 50 years. John, as he was known universally to his colleagues and students, leaves behind a long legacy of intellectual achievements.

John was a direct intellectual descendant of the cybernetics era, and early on was strongly influenced by the work of von Neumann, Wiener, Ashby, and Turing, all of whom viewed computation as a broad, interdisciplinary enterprise. Thus, John became an early proponent of interdisciplinary approaches to computer science and was an active evangelist of what is now called *c*omputational thinking, reaching out enthusiastically to psychologists, economists, physicists, linguists, philosophers, and pretty much anyone he came in contact with. As a result, even though he received what was arguably the world's first computer science Ph.D. in 1959, his contributions are sometimes better known outside of CS than within.

John Holland is best known for his invention of *g*enetic algorithms (GAs), a family of search and learning methods inspired by biological evolution. Since their invention in the 1960s, GAs, along with related evolutionary computation methods, have become a thriving subfield of computer science, with widespread scientific and commercial applications. While the mechanisms of GAs are well-known to much of the CS community, fewer are aware that GAs were only one offshoot of Holland's much broader motivation—to develop a general theory of adaptation in complex systems. This motivation was the driving force in all of Holland's research.

In this short essay, we sketch five key, recurrent themes of Holland's work on adaptive systems: (1) the evolutionary dynamics of "building blocks"; (2) learning via credit assignment and rule discovery; (3) the emergence of internal models; (4) the art of scientific modeling; and (5) the universal properties of complex adaptive systems. We discuss the role these themes have played in computer science, and highlight especially the ideas that we think remain relevant to today's research agendas.

2. EVOLUTIONARY DYNAMICS OF "BUILDING BLOCKS"

Holland's goal of developing a general theory of adaptation was spurred both by his early work on computer models of Hebbian learning [17] and his reading of Ronald Fisher's classic work on integrating genetics with Darwinian selection [7]. As Holland further read extensively in evolutionary biology, economics, game theory, and control theory, he came to recognize that adaptation was the most central concept in all these fields. That is, these fields all concern populations of agents that must continually obtain information from uncertain changing environments and use it in order to improve their performance with respect to those environments. Moreover, Holland recognized that in systems with *adaptive agents* of this kind, there is never a state of *e*quilibrium or a final *o*ptimum configuration. Due to the environment's "perpetual novelty" (as Holland termed it), adaptation continues forever in an open-ended way. Holland's focus on open-ended, non-equilibrium dynamics was in stark contrast with the mainstream approach (at the time) in all these fields—the belief that "solving for" stable equilibrium dynamics was the scientific goal. Holland's contrary view was that a system in stable equilibrium is essentially *d*ead.

Underlying Holland's theory of adaptation are three core ideas:

- **Populations:** Adaptation occurs in populations of individuals evolving (or learning) over time, in which statistics can be leveraged to direct population dynamics. (More on this below.)

- **Building Blocks:** In a population undergoing adaptation, individuals can be decomposed into *b*uilding blocks—sets of traits that are the evolutionary "atoms" of an individual's fitness or performance. As an example from biology, Holland gives the *K*rebs cycle, a core cellular metabolic pathway. Similarly, in neuroscience, a building block of visual perception might be a group of neurons that responds to an oriented edge; in computer science, a building block of a sorting program might be a "swap" operator.

- **Exploitation Versus Exploration:** Successful adaptation requires the right tradeoff between *e*xploitation, in which tried-and-true building blocks propagate in a population, and *e*xploration, in which existing building

> blocks are recombined or mutated in new ways. A system that focuses too much on exploitation risks never finding better individuals or not being able to adapt to changing environments. Conversely, too much focus on exploration means that the system won't leverage the successful building blocks that have already been discovered. There is some optimal balance between these two modes of processing.

Several of Holland's early papers (e.g., [10, 11]) and his influential 1975 book Adaptation in Natural and Artificial Systems [12] developed a general, formal setting in which these ideas could be expressed mathematically.

In particular, inspired by Bellman [2] and others, Holland explored the exploitation-versus-exploration tradeoff via an idealized "two-armed bandit" problem. Given a slot machine with two arms, each of which has an unknown payoff probability, how should you allocate N trials (pulls) between the arms so as to maximize your total payoff? An example of an extreme exploitation strategy would be to alternate between the arms until one of them gives a payoff, and then allocate all future trials to that arm alone. Conversely, an example of an extreme exploration strategy would be to randomly allocate the trials, irrespective of the payoff rates you obtain. Obviously, each of these strategies is flawed. What is an optimal strategy?

In [11] and [12], Holland derived an equation for such a strategy. Let A denote the arm currently observed to have the higher payoff probability, and B denote the other arm. Holland showed that the optimal strategy (the one that yields maximal payoff, or minimal loss) is for the number of trials allocated to A to grow slightly faster than an exponential function of the number of trials allocated to B. Holland then showed that this result extends naturally to the multi-armed bandit case.

In an population undergoing adaptation, the building blocks making up individuals can be compared to the arms of a multi-armed bandit. Each building block can be said to have a probability of "payoff" (i.e., contribution to a given individual's fitness). Evaluating an individual in an environment is like "pulling the arms" on a multi-armed bandit, where the "arms" correspond to each of the building blocks making up that individual. Thus, in the process of assigning a fitness to each individual in a population, adaptive evolution can be seen as implicitly sampling the many building blocks making up that collection of individuals. The average fitness over many individuals containing a particular building block gives an estimate of that building block's payoff probability.

The question of how to balance exploitation and exploration—how to optimally allocate trials to different arms based on their observed payoff—now becomes the question of how to optimally sample in the vast space of possible building blocks, based on their observed average fitness. Of course adaptive evolution deals in populations of individuals, not building blocks. There is no explicit mechanism for keeping statistics on building blocks' "observed average fitness". However, Holland's central idea here is that an approximation to optimal building-block sampling does indeed occur, as an emergent property of the population dynamics.

To show this mathematically, Holland defined an idealization of adaptive evolution: an algorithm he called a reproductive plan. A reproductive plan is a population-based stochastic process operating on bit strings, involving the "genetic" operators of reproduction, fitness-based selection, crossover, mutation, and inversion. Building blocks are formalized as schemata—patterns of bits within a string. (Here are two examples of schemas: "all strings beginning with the bits 10" or "all strings that start with the pattern 1*1", where * can be replaced by either 0 or 1).

Holland's most important result was the following: He proved mathematically that his reproductive plan results—implicitly—in a near-optimal allocation of trials to schemata, thus optimizing the exploitation versus exploration balance. Holland noted that the explicit act of assigning fitnesses to the individuals in a population was actually implicitly sampling a much larger collection of building blocks. Holland termed this implicit parallelism; the parallel leveraging of statistics from implicit sampling is a main strength of his population-based approaches to search.

It was this formal setting and resulting theorems that led to the invention of genetic algorithms, which featured stochastic population-based search and crossover as a critical operation: Crossover recombined successful building blocks and thus allowed them to be tested in new contexts. What made GAs unique among other evolution-inspired algorithms at the time were the mathematical foundations described above, including bit-string representations, the focus on recombination as a central mechanism, and a focus (at least in Holland's mind) on continual adaptation to non-stationary environments rather than optimization to static environments.

It is important to point out again that the framework Holland developed in [12] was more general than the genetic algorithm; it was created in order to develop an interdisciplinary theory of adaptation, one that would inform biology, say, as much as computer science [4]. The later, successful application of genetic algorithms to real-world optimization and learning tasks was, for Holland, just icing on the cake.

3. LEARNING VIA CREDIT ASSIGNMENT AND RULE DISCOVERY

While genetic algorithms are nowadays most often associated with models of biological evolution, in Holland's view the processes of adaptation he was studying manifested themselves in many different complex systems. One system of special interest to Holland was the human mind, and in particular, its capacity to learn via induction.

In the 1970s and 80s Holland formulated and extensively explored a particular model of decision-making, action, and inductive learning, called the classifier system (e.g., [15, 9]. In a classifier system, a population of "if-then" rules interacts with an environment, and over time is able to learn to improve its performance via both credit-assignment and rule-discovery algorithms.

Like genetic algorithms, classifier systems can be viewed both as models of adaptation and as artificial-intelligence (AI) methods. At the time Holland was developing classifier systems, the field of AI was focused on expert systems, which typically did not learn on their own, a fundamental deficiency in Holland's view: "[Expert] systems are brittle in the sense that they respond appropriately only in narrow domains, requiring substantial human intervention to compensate for even slight shifts in domain" [9]. In contrast, Holland proposed that "induction is the basic, and perhaps only, way of making large advances in this direction."

Holland's inspiration for classifier systems came from several different disciplines. Knowledge representation in the form of a population of "if-then" rules seemed like a good choice, not only because of its popularity in AI at the time but also from Holland's early work on modeling Hebbian cell assemblies: "In Hebbs view, a cell assembly makes a simple statement: If such and such an event occurs, then I will fire for a while at a high rate." [21] (p. 182).

The if-then rules, when activated, compete to post their results on a shared "message list", modeling the system's short-term memory (again inspired by Hebb's work). The competition was based on an bidding process, in which a rule's current strength affected its ability to bid highly. Successful rules were gradually strengthened (and unsuccessful rules weakened) via a credit-assignment method called the *b*ucket-brigade algorithm, in which rules gaining rewards from the environment or from other rules would pay back part of their gains to those earlier-firing "stage-setting" rules that set up the conditions for the eventual reward. The idea for the bucket-brigade algorithm was inspired by the mechanisms of Hebbian learning and by early work on learning and optimization in economics, psychology, control theory, and other fields (e.g., [2, 18]). [1]

Finally, new rules for the population are discovered by a genetic algorithm, using the strength of rules as a fitness function.

Classifier systems are fairly complex architectures, with learning occurring at multiple time scales. Nevertheless, such systems have been usefully applied to areas as diverse as poker-playing [20], control of gas pipeline transmission [8], and modeling the stock market [16]. Modern reinforcement learning methods have largely taken over as more useful approaches to real-world decision and control problems, but classifier systems can perhaps be thought of as an essential "stage-setting" method that enabled the development of later approaches.

John Holland's main interest in classifier systems was how the two learning mechanisms could work together to create subsets of rules that accurately model the environment, and thus are able to "fire" at appropriate times. An important part of such internal models is the notion of a *d*efault hierarchy—a set of rules that together form a kind of hierarchical decision structure, going from most general to most specific. For example, a general rule might state that, "If X is a bird, then X can fly", whereas a more specific one might state, "If X is a bird with small wings and a large body, then X cannot fly" [13]. When faced with a bird in the environment, both rules might compete to post their conclusion, but the more specific one would be favored if both of its conditions are met.

Holland envisioned that, with a complex environment and enough time for learning, a classifier system might learn an intricate default hierarchy of rules, and that the competition inherent in the learning and action mechanisms would allow the system to adapt to a continually changing environment without losing what it has learned in the past. Holland put it this way: "Competition among rules provides the system with a graceful way of handling perpetual novelty. When a system has strong rules that respond to a particular situation, that is the equivalent of saying that it has certain well-validated hypotheses. Offspring rules, which begin life weaker than do their parents, can win the competition and influence the system's behavior only when there are not strong rules whose conditions are satisfied—in other words when the system does not know what to do. If their actions help, they survive; if not they are soon replaced. Thus, the offspring do not interfere with the system's action in well-practiced situations but wait gracefully in the wings as hypotheses about what to do under novel circumstances." [14]. A detailed discussion of some real-world applications of classifier systems is given in [3].

[1] Of course similar credit-assignment methods have been used extensively in reinforcement learning, neural networks, and other areas of modern machine learning, but Holland was among the first to propose such methods.

4. INTERNAL MODELS AND DEFAULT HIERARCHIES

The concept of *i*nternal models is central to Holland's theory of adaptive systems. He posits that all adaptive systems create and use internal models to prosper in their environments, through anticipation. Models can be tacit and learned through evolutionary time, as in the case of bacteria swimming up a chemical gradient, or explicit and learned over a single lifespan, as in the case of cognitive systems that incorporate experience into internal rule sets through learning.

In [13], Holland and his co-authors discuss how such internal models can be learned, without supervision, by combining environmental inputs with stored knowledge. A key idea is that a model defines sets of equivalence relations over environmental states, together with a set of transition rules, which are learned over time based on environmental "rewards" (or "punishments"). Models that form valid homomorphisms with the environment allow the system to make accurate predictions. In Holland's conception, the equivalence classes are initially very general, say "moving object" and "stationary object," and over time, through experience and learning, they are specialized into more useful and precise subclasses, say "insect" and "nest". As described above, the adaptive system eventually learns a default hierarchy of rules involving both general and specific classes.

Although the idea of default hierarchies is prevalent in many knowledge representation systems (LIST A FEW?), Holland made two key contributions. The first was his emphasis of homomorphisms as a formal way to evaluate model validity. This idea that dates back to Ross Ashby's *An Introduction to Cybernetics* [1]. Holland's student, Bernard Ziegler developed this idea into a formal theory of computer modeling and simulation [22]. Even today, these early homomorphic theories of modeling stand as the most elegant approach we know of for characterizing when a model is consistent with the environment and how an intelligent agent, human or artificial, can update the model to better reflect reality.

Holland's second key contribution was describing a computational mechanism—classifier systems—by which a cognitive system could iteratively build up a detailed and hierarchical model of its environment. The key learning elements of this method, the bucket-brigade algorithm, combined with a genetic algorithm, as described above, presaged many of the ideas in modern reinforcement learning.

5. THE ART OF SCIENTIFIC MODELING

Given that Holland believed that the ability to learn and manipulate internal models was essential for any adapting system, it is no surprise that he viewed modeling as essential for scientific inquiry.

Today, we use computational models both for *prediction*—by analyzing data via statistical models—and for *understanding* how systems work—by probing the effects of hypothesized underlying mechanisms. This latter use of models was dear to Holland's heart. In his view, the key to science was understanding the mechanisms that cause a system to behave in a certain way, an aspiration that goes well beyond data fitting methods, which typically focus only on describing the aggregate behavior of a system.

For example, a statistical model that describes the boom and bust pattern of the stock market cannot address the underlying mechanisms that lead to these cycles, through the collective actions of myriad individual buy/sell decisions. Similarly, the genetic algorithms for which Holland is so famous provide a simple computational framework in which to explore the dynamics of Darwinian evolution and whether the basic mechanisms of variation, differential reproduction, and heredity are sufficient to account for the richness of our natural world.

The emphasis on exploratory models to build intuitions was an important theme of Holland's work, and he often quoted Eddington's remark on the occasion of the first experimental test of Einstein's theory of relativity: "The contemplation in natural science of a wider domain than the actual leads to a far better understanding of the actual" [6].

Holland was interested in models that explored basic principles and mechanisms, even if they didn't make specific or detailed predictions. Such models can show generically how certain behaviors could be produced. Holland pioneered a style of modeling that has come to be known as 'individual based' or 'agent based,' in which every component of a system is represented explicitly and has state, e.g., every trader in a stock market system or every cell in an immune system model. In such models, each agent has its own behavior rules, which it can update (or learn) over time. In order to capture the constraints of systems living under spatial constraints, these models are often defined over spatial structures, such as networks or simple grids.

A given agent-based model encodes a theory about the mechanisms that are relevant for producing the behavior of interest. Similar to expert systems, such models are especially useful for studying systems that don't have analytic mathematical descriptions. Agent-based models can facilitate interdisciplinary collaborations because the underlying rules can be easily communicated. The agent-based models championed by Holland were typically idealized versions of complex systems and not intended to provide detailed, domain-specific predictions. Instead they were meant to explore possible general mechanisms of complex systems and thus provide insights that might lead to more specific, detailed models. Such idealized models are akin to what Dennett has called "intuition pumps" [5].

It should be noted that Holland's view of modeling is by no means typical. For example, in a textbook on computational modeling, the authors offer the following definition: âĂIJModeling is the application of methods to analyze complex, real-world problems in order to make predictions about what might happen with various actionsâĂİ [19]. This sort of perspective completely rules out the kind of exploratory modeling that Holland was most interested in.

Some researchers dispute that models make any kind of scientific contribution: "Models are metaphors that explain the world we don't understand in terms of worlds we do. They are merely analogies, provide partial insight, stand on someone else's feet. Theories stand on their own feet, and rely on no analogies." [Emanuel Derman, 2012]. [STEPH, I COULD NOT FIND REFERENCE FOR THIS QUOTE OUTSIDE SFI VIDEO.]

[NOW WE NEED A STRONG FINISHING SENTENCE TO RESCUE JHH STYLE MODELING.]

6. COMPLEXITY

7. RELEVANCE TO MODERN CS

- Evolutionary computation
- Q Learning
- Backprop
- From Turing nomination: "Holland's machine learning system known as the Learning Classifier System (LCS), developed in the early 1980s, incorporated a reinforcement learning algorithm known as the bucket brigade for non-Markovian environments, anticipating by nearly a decade non-Markovian learning algorithms."
- Exploitation versus exploration – relevance to reinforcement learning, other parts of machine learning and optimization.
- Active learning. (Two-armed bandit problem.)
- On-line learning.

8. CONCLUSION

Introduced a few generations of students to computation in natural systems, an idea that today is better accepted. His insights were deeper and more general than what often passes for work in biomimicry, e.g., for robots.

The ideas have had huge impact and should still be a beacon for research in intelligent and complex systems

John's personality and humanity is inextricably tangled up with his intellectual contributions.

9. REFERENCES

[1] W. R. Ashby. *An Introduction to Cybernetics.* Chapman & Hall, London, 1956.

[2] Richard Ernest Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[3] L.B. Booker, D.E. Goldberg, and J.H. Holland. Classifier systems and genetic algorithms. *Artificial Intelligence*, 40(1-3):235–282, 1989.

[4] Freddy Bugge Christiansen and Marcus W. Feldman. Algorithms, genetics, and populations: The schemata theorem revisited. *Complexity*, 3(3):57–64, 1998.

[5] Daniel C. Dennett. *Elbow Room: The Varieties of Free Will Worth Wanting*. MIT Press, 1984.

[6] Arthur Eddington. *The Nature of the Physical World: Gifford Lectures, 1927*. Cambridge University Press, Cambridge, UK, 1927.

[7] R. A. Fisher. *The Genetical Theory of Natural Selection: A Complete Variorum Edition.* Oxford University Press, 1930.

[8] D.E. Goldberg. *Computer-aided gas pipeline operation using genetic algorithms and rule learning.* PhD thesis, University of Michigan, Ann Arbor, MI, 1983.

[9] J. H. Holland. Escaping brittleness: {T}he possibilities of general purpose learning algorithms applied to parallel rule-based systems. *Machine Learning*, 2:593–623, 1986.

[10] John H. Holland. Outline for a Logical Theory of Adaptive Systems. *Journal of the ACM*, 9(3):297–314, 1962.

[11] John H. Holland. Genetic Algorithms and the Optimal Allocation of Trials. *SIAM Journal on Computing*, 2(2):88–105, 1973.

[12] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* University of Michigan Press, 1975.

[13] John H. Holland. *Induction: Processes of Inference, Learning, and Discovery.* MIT Press, 1989.

[14] John H Holland. Genetic Algorithms. *Scientific American*, 267(1):44–50, 1992.

[15] John H. Holland and Judith S. Reitman. Cognitive systems based on adaptive algorithms. *ACM SIGART Bulletin*, 63:49, 1977.

[16] R.G. Palmer, W. Brian Arthur, John H. Holland, Blake LeBaron, and Paul Tayler. Artificial economic life: a simple model of a stockmarket. *Physica D: Nonlinear Phenomena*, 75(1-3):264–274, 1994.

[17] N. Rochester, J. Holland, L. Haibt, and W. Duda. Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IEEE Transactions on Information Theory*, 2(3):80–93, 1956.

[18] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[19] Angela B. Shiflet and George W. Shiflet. *Introduction to Computational Science: Modeling and Simulation for the Sciences*. Princeton University Press, 2014.

[20] Stephen Frederick Smith. *A learning system based on genetic adaptive algorithms*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, 1980.

[21] Mitchell M. Waldrop. *Complexity: The Emerging Science at the Edge of Order and Chaos*, volume 8. Simon and Schuster, 1993.

[22] Bernard Ziegler. *Theory of Modeling and Simulation.* Wiley Interscience, New York, 1976.