The Quantitative Law of Effect is a Robust Emergent Property of an Evolutionary Algorithm for Reinforcement Learning

J.J McDowell[1] and Zahra Ansari

Department of Psychology, Emory University, Atlanta, Georgia, U.S.A., 30322
jack.mcdowell@emory.edu, zansari@learnlink.emory.edu

Abstract. An evolutionary reinforcement-learning algorithm, the operation of which was not associated with an optimality condition, was instantiated in an artificial organism. The algorithm caused the organism's behavior to evolve in response to selection pressure applied by reinforcement from the environment. The resulting behavior was consistent with the well-established quantitative law of effect, which asserts that the time rate of a behavior is a hyperbolic function of the time rate of reinforcement obtained for the behavior. The high-order, steady-state, hyperbolic relationship between behavior and reinforcement exhibited by the artificial organism did not depend on specific qualitative or quantitative features of the evolutionary algorithm, and it described the organism's behavior significantly better than other, similar, function forms. This evolutionary algorithm is a good candidate for a dynamics of live behavior, and it might be a useful building block for more complex artificial organisms.

1 Background: Matching Theory and Reinforcement Learning

During the past three decades, the mathematical description of behavior-environment relationships has become an important part of the experimental analysis of behavior. Perhaps the most widely studied and successful mathematical work in behavior analysis is the family of equations known as matching theory [1]. In dozens of experiments with many species, including humans, matching theory has been shown to accurately describe the relationship between properties of behavior and properties of a variety of psychologically significant environments. The most fundamental equation of matching theory is its hyperbolic rate equation, which is often referred to as the quantitative law of effect.

As is well known, E. L. Thorndike (c. 1911) discovered the law of effect, or principle of reinforcement, in his famous puzzle-box experiments. B. F. Skinner (c. 1938) later gave the law of effect a stochastic cast by stating that positive reinforcement increased the probability of a behavior's future occurrence. In 1961, Skinner's student, R. J. Herrnstein, published an influential paper [3] in which he reported that pigeons' rates of choosing various alternatives (i.e., keys to peck) in a multi-alternative

[1] Presentation of this paper was aided by a faculty travel grant to the first author from the Institute of Comparative and International Studies, Emory University.

M. Capcarrere et al. (Eds.): ECAL 2005, LNAI 3630, pp. 413 – 422, 2005.

environment was governed with a remarkable degree of accuracy and precision by a simple algebraic equation that related the rate of key pecking on the various alternatives to the rate of reinforcement obtained for pecking on those alternatives. This equation came to be known as the matching law. From this equation, Herrnstein [4] derived the hyperbolic rate equation in 1970, and since then a great deal of experimental and mathematical research on matching theory has expanded its scope to many species, behaviors and reinforcers, and to a variety of experimental and naturalistic environments [1].

The hyperbolic rate equation, or quantitative law of effect, states how the absolute rate of a behavior, R, in a given environment is governed by the absolute rate of reinforcement, r, obtained for that behavior,

$$R = \frac{kr}{r + r_e}, \tag{1}$$

where k and r_e are parameters of the hyperbola. The parameter, k, is the y-asymptote of the hyperbola, and r_e determines its curvature, that is, how rapidly the function approaches its asymptote. As interpreted by matching theory, k is related to properties of behavior such as the amount of effort the behavior requires, and r_e is related to additional sources of reinforcement that may be available in the environment. In behavior analysis, Equation 1 is now recognized as a fundamental statement of the way reinforcement governs behavior.

An important feature of Equation 1 is that it describes behavior in the steady state, when it is in equilibrium with conditions in the environment. Each point on the hyperbola represents, for a particular behavior and a particular reinforcer, the average equilibrium response rate that is supported by an average reinforcement rate. In most experimental situations, R >> r, in other words, relatively few instances of the behavior are reinforced. The problem of how behavior gets to the steady state has been pursued vigorously, but as yet has not yielded a generally accepted mathematical dynamics. As might be supposed, one of the most popular approaches to this problem is based on optimality theory [9]. Another approach is based on linear filtering [7], and very recent work has made use of computational modeling based on an evolutionary algorithm [6]. The computational approach to behavioral dynamics, which dovetails with work on reinforcement learning in artificial life and related disciplines, is the subject of this article.

Reinforcement learning algorithms in machine learning and artificial intelligence fall into two broad categories. Algorithms in one category deal with the expected utility or value of different courses of action [5, 12]. Temporal-difference learning is an example of this type of algorithm. Utility-based algorithms have been applied to many problems, including some that are relevant to the behavior of live organisms, such as chaining [13], conditioned reinforcement [13], and multi-alternative responding that is consistent with Herrnstein's original matching equation [2]. The second category of reinforcement learning algorithms is concerned with finding the best action or policy in a particular set of circumstances [8]. These algorithms usually entail evolutionary principles. Action-based evolutionary algorithms have also been widely applied, including to problems that are relevant to the behavior of live organisms, such as foraging in multi-alternative environments, which can also be described by the original matching equation [10-11]. Virtually all of the existing utility-based and action-based reinforcement learning algorithms are designed to solve an optimality problem, that is, they work either by attempting to maximize the expected utility of a sequence of actions, or by attempting to maximize in some way the overall outcomes of an agent's actions.

The reinforcement learning algorithm that will be discussed in this article falls into the second category, although it is not a typical example of this category. It is an evolutionary algorithm that is not, however, designed to solve an optimality problem. Instead, it is simply used as the dynamic mechanism of an artificial organism's behavior. The organism's behavior evolves through a process of selection, reproduction and mutation over many generations, or time steps, where selection pressure is applied by the environment in the form of reinforcing stimuli. The behavior reaches steady states in response to constant time-averaged reinforcement rate inputs, and these steady states can be compared to the requirements of Equation 1. The questions of interest in this research are whether the behavior of an artificial organism that operates according to evolutionary principles conforms to Equation 1, and if so, whether this conformance depends on specific implementations of the evolutionary principles.

2 The Artificial Organism and Evolutionary Algorithm

In this section, the structure and operation of the artificial organism will be described, along with the components of the evolutionary algorithm that constitutes its dynamics.

2.1 The Artificial Organism

The artificial organism consists of 100 10-bit strings that represent integers ranging from 0 through 1023. This collection of bit strings constitutes the organism's repertoire of behaviors or actions. The behaviors can be sorted into classes, called operants, based on how they act upon the environment. A rat's or human's lever press in an experimental chamber, for example, is an operant defined by a switch closure. Individual members of this class include a lever press with the right limb, a lever press with the left limb, a high-force press that exceeds the force required for switch closure, and so on. Partitioning the 100 bit strings into operant classes sets the baseline structure and operation of the artificial organism. For our purposes we will define just two classes, one consisting of the 41 integers from 0 through 40, and one consisting of the remaining 983 integers. The first behavioral class will be designated the target operant, analogous to a lever press. The second behavioral class represents doing something else.

The artificial organism is initialized with 100 10-bit strings selected at random from the 1024 possible strings. The organism's behavior at each time step is determined by the relative frequencies with which the integer values of these strings fall into the different operant classes. The relative frequencies constitute the probabilities that the organism will emit a behavior from each class, and these probabilities are used to determine which operant the organism emits at each moment.

2.2 Fitness

When an operant is reinforced, it is identified as fit with respect to conditions in the environment. Two definitions of fitness will be considered. For *midpoint fitness*, the integer midpoint of the reinforced class of behavior is taken as the fitness criterion, that is, it represents the fittest individual behavior. For *specific individual fitness*, the fitness criterion is the integer value of a specific individual behavior selected from the reinforced class, based on the relative frequencies of the individual members of that class. In both cases, the fitness of each of the 100 bit strings that constitute the organism's behavioral repertoire is defined as the absolute value of the difference between that bit string's integer value and the fitness criterion. Note that this method of defining fitness means that lower fitness values are associated with fitter individual behaviors.

2.3 Parents

Following a reinforcement, parents are chosen for mating on the basis of their fitness by selecting fitness values from a uniform fitness density function,

$$p(x) = \frac{1}{2\mu} \qquad \text{for} \quad 0 \le \mathrm{x} \le 2\mu\,, \tag{2}$$

a linear fitness density function,

$$p(x) = -\frac{2}{9\mu^2}x + \frac{2}{3\mu} \qquad \text{for} \quad 0 \le x \le 3\mu\,, \tag{3}$$

or an exponential fitness density function,

$$p(x) = \frac{1}{\mu}e^{-\frac{1}{\mu}x}\,. \tag{4}$$

For all functions, *p(x)* is the probability density associated with a fitness value, *x*, and μ is the mean of the density function. These fitness density functions are completely determined by their means. They associate higher probability densities with lower fitness values, and hence with fitter individual behaviors. A general method for constructing functions of this type is given in [6].

Following a reinforcement, a father behavior is chosen from the repertoire by drawing a fitness value at random from one of the fitness density functions, and then searching the organism's repertoire for a behavior with that fitness. If none is found, then another fitness value is drawn at random from the fitness density function, and so on, until a father behavior is found. A distinct mother behavior is obtained in the same way.

In the event that reinforcement does not occur at a given time step, parents are selected at random from the organism's repertoire. In either case, 100 sets of parents are chosen, each of which produces one child behavior. The resulting set of 100 child behaviors then replaces the artificial organism's behavioral repertoire, and a behavior from this repertoire is chosen for emission using the method described earlier.

2.4 Reproduction

Two types of reproduction will be considered. In *bitwise* reproduction, each bit in a child's bit string is set equal to the corresponding bit either from the father's bit string or from the mother's bit string, each with a probability of 0.5. In *crossover* reproduction, the parents' bit strings are sliced at a random location and then combined by crossing over. One of the resulting bit strings is chosen at random as the child.

2.5 Mutation

After a new generation of behaviors has been produced, a fixed percentage of the behaviors undergoes mutation, that is, the behaviors' integer values are changed. The individual behaviors that undergo mutation are chosen at random from the organism's repertoire. Three methods of mutation will be considered. In *Gaussian* mutation, the integer value of the chosen behavior is taken as the mean of a Gaussian distribution of integers with a specific standard deviation. A value chosen at random from this distribution is then taken as the mutant. Should the mutant fall outside the range of acceptable values (0-1023), it is wrapped to the other end of the range. In *bit-flip* mutation, one bit from the chosen behavior's bit string, selected at random, is changed. In *random individual* mutation, the integer value of the chosen behavior is replaced with a value selected at random from the range, 0-1023.

3 Experimental Studies of the Artificial Organism's Behavior

Extensive parametric studies of the artificial organism's behavior have been conducted [6], and will be summarized here. The purpose of these studies was to determine whether the behavior of the artificial organism conformed to Equation 1, and if so, whether this conformance depended on specific implementations of the rules of the evolutionary algorithm. In all experiments, reinforcement was set up, or made available, at random times following the delivery of the previous reinforcement. Once reinforcement was made available, it was delivered as soon as the organism emitted the target operant. Environments that work in this way are said to arrange random interval (RI) schedules of reinforcement. An RI schedule is characterized by the mean of its intervals. Evidently, an RI schedule with a small mean arranges frequent reinforcement for the target operant, whereas an RI schedule with a large mean arranges infrequent reinforcement.

In the three series of experiments to be described in this section, the mean of the RI schedules ranged from 1 to 200 time ticks. A single experiment consisted of arranging a series of approximately 10 RI schedules, each with a different mean, one at a time. Each schedule remained in effect for 5,000 to 45,000 generations, or time steps, after which the next schedule was arranged, and so on. Each schedule yielded an average rate of emission of the target operant, R, and an average rate of reinforcement, r.

At the beginning of the experiment, an initial interval from the RI schedule was started, and the organism emitted its first behavior according to the method described earlier. If the emitted behavior came from the target class, and if its latency since the last reinforcement (or since the start of the session in this case) equaled or exceeded the scheduled random interval, then a reinforcer was delivered. A new generation of behaviors was then produced using a fitness density function, and a new interval from the RI schedule was started. The organism then emitted its second behavior, and so on. If at any time a target operant was emitted but not reinforced, or if the emitted behavior did not come from the target class, then a new generation of behavior was produced from random parents, after which the organism emitted its next behavior, and so on.

3.1 Parametric Study of the Form and Mean of the Fitness Density Function

Five experiments were conducted using a uniform fitness density function, five were conducted using a linear fitness density function, and five were conducted using an exponential fitness density function. The five experiments for each function form arranged mean fitnesses (μ in Equations 2-4) ranging from 10 to 200. In all experiments the midpoint fitness definition and bitwise reproduction method were used. Gaussian mutation with a standard deviation of 25 was used to produce mutants of 3% of each generation's behaviors.

The behavior of the artificial organism in these experiments reached a dynamic equilibrium with the RI schedule such that the momentary rate of the organism's behavior varied around a stationary mean value. Reinforcements tended to pull the organism's population of bit strings into the target class, while nonreinforcement tended to pull the population of bit strings out of the target class and return the organism to its baseline state.

An example of the steady-state behavior of the artificial organism over the range of RI schedules used in these experiments is shown in Figure 1. The data in the left panel were generated using a linear fitness density function with $\mu = 40$. Data are shown from only the last 500-generation block, and for only a few of the RI schedules used in the experiment. The smooth curve is the best fitting hyperbola (Equation 1), which accounts for 98% of the variance in the target behavior. The outcome shown in this panel is typical of experiments with live organisms. It also may be worth noting that the method of arranging RI schedules and of averaging response and reinforcement rates used in these experiments are identical to the methods used in experiments with live organisms. Data in the right panel of Figure 1 were generated using an exponential fitness density function with $\mu = 40$. The data were averaged over approximately 40 500-generation blocks and are shown for all the RI schedules used in the experiment. The smooth curve is the best fitting hyperbola, which accounts for more than 99% of the variance in the target behavior.

The outcome shown in the right panel of Figure 1 is typical of the outcomes of all experiments in this series. When the data were averaged over 5,000 to 45,000 generations, which reduced the standard errors of these means to very small values, Equation 1 accounted for essentially all the variance (> 99% in most cases) in the artificial organism's target behavior, and this was the case regardless of the form or mean of the fitness density function used to generate the data.

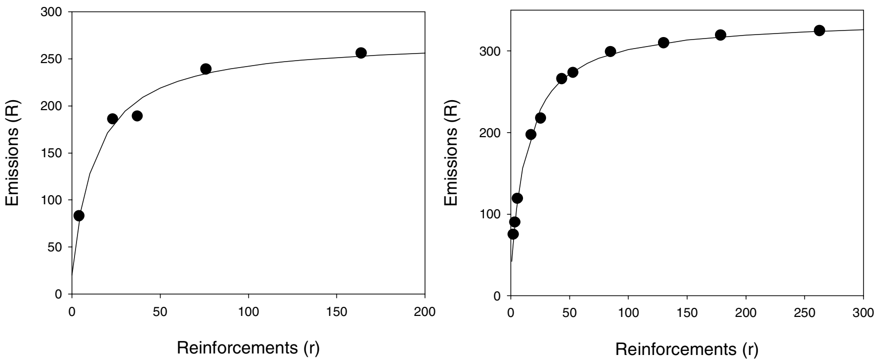

Fig. 1. Target behavior emissions per 500-generation block plotted as a function of contingent reinforcements per 500-generation block. Smooth curves are best fitting hyperbolas (Equation 1). The left panel, which shows data from only the last 500-generation block, is similar to the outcome of an experiment with a live organism. The right panel shows data for more RI schedules (using a different fitness density function), and averaged over about 40 500-generation blocks.

The hyperbolic form of the function relating the rate of the target behavior and the rate of reinforcement obtained for the target behavior was further tested by comparing it to fits of a two-parameter asymptotic exponential, a two-parameter asymptotic power function, and a two-parameter ramp function. The latter is a piecewise continuous function consisting of a line that increases from the origin, followed by a constant value that begins at some positive reinforcement rate. This is arguably the simplest function form that can describe data that ascend from the origin and then level off. The asymptotic exponential and asymptotic power functions have differential properties similar to those of a hyperbola.

The four function forms (including the hyperbola) were compared on the basis of the percentage of variance they accounted for, and in terms of the randomness of the residuals left by their least squares fits. Based on these criteria, the hyperbola provided a better fit to the data from the fifteen experiments in this series than did the other function forms and it accounted for essentially all the variance in the data. The other forms accounted for significantly less variance, and left residuals that showed significantly more deviations from randomness. These results indicate that the artificial organism's steady-state behavior was consistent with the quantitative law of effect, and that the hyperbolic form of the relationship between target behavior frequency and reinforcement frequency was both unique and robust, that is, it provided a better description of the data than other, similar, function forms, and did not depend on the form or mean of the fitness density function.

3.2 Study of Variations in the Components of the Evolutionary Algorithm

In a series of twelve experiments, different combinations of fitness definition, reproduction method, and mutation method, along with various fitness density function forms, were studied. Specific component variations tested included the specific individual fitness definition, crossover reproduction, bit-flip mutation, and random individual mutation. Twelve combinations of these component variations, along with the variations used in the first series of experiments were tested.

Least squares fits of a hyperbola to the data from these experiments accounted for essentially all the variance of the target behavior. The three alternative function forms were also fitted to the data and were found to account for significantly less variance than the hyperbola, and to leave residuals showing significantly more deviations from randomness. These results indicate that the hyperbolic relationship between target behavior frequency and reinforcement frequency did not depend on any specific definition of fitness, or on any specific implementation of reproduction or mutation, or on any specific combination of these component variations, although only a subset of the possible combinations of component variations was tested.

3.3 Parametric Study of Mutation Rate

Using the same component variations as in the first series of experiments, together with a linear fitness density function, all possible combinations of five fitness function means (10, 20, 40, 100, and 200) and six mutation rates (1%, 3%, 5%, 10%, and 20%) were studied in thirty experiments.

Again, least squares fits of a hyperbola accounted for essentially all the variance in the target behavior for these thirty data sets, and the three alternative function forms accounted for significantly less variance and left residuals that were significantly less random than the hyperbola. These results indicate that the hyperbolic form of the behavior-reinforcement relationship does not depend on the mutation rate.

Data from these experiments also permit a parametric examination of the effects of mean parental fitness and mutation rate on the parameters, k and r_e, of the hyperbola. Both were affected by the two variables, but k was much more strongly affected by mean parental fitness, whereas r_e was much more strongly affected by mutation rate. Recall that lower mean parental fitnesses cause fitter parents to be selected for mating. The results of these experiments showed that the fitter the parents selected for mating, the higher the asymptote of the hyperbola. Put another way, a given reinforcement rate, r, generated a higher response rate, R, the fitter the parents selected for mating. This effect is analogous to the effect of reinforcer magnitude on the behavior of live organisms. Hence larger reinforcer magnitudes can be represented by lower mean parental fitnesses in the evolutionary algorithm.

The principal effect of higher mutation rates was to increase the value of r_e and hence decrease the curvature of the hyperbola. Put another way, to achieve a given response rate, R, a greater reinforcement rate, r, was required the greater the mutation rate. Not surprisingly, then, mutation diluted the effect of reinforcement.

4 Conclusion and Future Directions

An evolutionary algorithm, the operation of which was not associated with an optimality condition, was used as a behavioral mechanism for an artificial organism, and was shown to generate steady-state behavior consistent with the well-established quantitative law of effect (Equation 1). Three series of experiments demonstrated that this result was robust and unique. The result was robust inasmuch as it was independent of the specific methods of implementing the rules of the evolutionary algorithm. Evidently, robust outcomes of evolutionary algorithms for reinforcement learning are not unusual [8]. The result was unique inasmuch as a hyperbola described the organism's steady-state behavior better than other, similar, function forms.

While steady-state behavior was the focus of this research, the evolutionary algorithm used in these experiments also gives an artificial organism the ability to adapt continuously to a dynamic environment by tracking changes in reinforcement contingencies. In the absence of reinforcement, the organism's repertoire reverts to its baseline state over a number of generations. The extent to which the dynamics of the evolutionary algorithm, such as its time course, correspond to the dynamics of the behavior of live organisms remains a topic for future research.

The state space of the artificial organism used in these experiments was very simple, which reflects its origin as an analog of the basic unit of behavioral experimentation, namely, a single organism in a restricted environment that interacts with only one class of behavior. Indeed, the artificial organism operated in just one state, from which it could emit one of only two classes of behavior. This restricted repertoire is much simpler than the policies that are often studied in research on utility-based and action-based reinforcement learning [5, 8]. But just as the basic laboratory preparation is a building block for more complicated experimental environments, the evolutionary algorithm described here might prove useful as a building block for dealing with more complicated state spaces.

In the experimental analysis of behavior, a state space is characterized by what is called a discriminative stimulus, and behavior associated with that (often complex) stimulus is said to be under its control or, more generally, under stimulus control. Mapping behavior to discriminative stimuli in behavior analysis is analogous to mapping actions to states in artificial intelligence research, although the former mapping is always probabilistic. A sequence of mappings between discriminative stimuli and behavior constitutes what would be referred to in artificial intelligence research as a policy. The work described in this article dealt with a single mapping of one state to a set (with only two members) of probabilistic actions. There are many approaches to building a more complicated policy. One is to switch from 10-character bit strings to 100-character integer strings, each of which represents the artificial organism's behavioral repertoire in the presence of a different discriminative stimulus. The repertoire represented by each integer string would evolve (presumably in conformance with Equation 1) in the presence of its discriminative stimulus, and the collection of integer strings at any moment would constitute the organism's policy at that moment. This approach would engage the problem of credit assignment inasmuch as reinforcement could be delivered after a sequence of actions. Methods of dealing with this problem include using chaining mechanisms and conditioned reinforcement, an approach taken by Touretzky and Saksida [13], or using a delay-of-reinforcement gradient that is informed by findings from live organisms.

This research lies at the interface of the experimental analysis of behavior and artificial life. The evolutionary algorithm described in this article is a good candidate for a dynamics of live behavior, and it might be a useful building block for more complex artificial organisms that have the ability to adapt continuously to complex environments.

References

1. Davison, M., McCarthy, D. *The matching law*. Erlbaum, Hillsdale, N.J. (1988)
2. Daw, N.D., Touretzky, D.S. Operant behavior suggests attentional gating of dopamine system inputs. *Neurocomputing* 38-40 (2001) 1161-1167.
3. Herrnstein, R.J. Relative and absolute strength of response as a function of frequency of reinforcement. *Journal of the Experimental Analysis of Behavior*, 4 (1961) 267-272.
4. Herrnstein, R.J. On the law of effect. *Journal of the Experimental Analysis of Behavior*, 13 (1970) 243-266.
5. Kaelbling, L.P., Littman, M.L., Moore, A.W. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4 (1996) 237-285.
6. McDowell, J. J A computational model of selection by consequences. *Journal of the Experimental Analysis of Behavior* 81 (2004) 297-317.
7. McDowell, J. J, Bass, R., Kessel, R. A new understanding of the foundation of linear system theory and an extension to nonlinear cases. *Psychological Review* 100 (1993) 407-419.
8. Moriarty, D.E., Schultz, A.C., Grefenstette, J. J Evolutionary algorithms for reinforcement learning. *Journal of Artifical Intelligence Research* 11 (1999) 241-276.
9. Rachlin, H., Battalio, R., Kagel, J., Green, L. Maximization theory in behavioral psychology. *Behavioral and Brain Sciences* 4 (1981) 371-417.
10. Seth, A.K. Evolving behavioural choice: An investigation into Herrnstein's matching law. In Floreano, D., Nicoud, J.D., Mondana, F. (eds.) *Proceedings of the Fifth European Conference on Artifical Life*. Springer-Verlag, Berlin Heidelberg New York (1999) 225-236.
11. Seth, A. K. Modeling group foraging: Individual suboptimality, interference, and a kind of matching. *Adaptive Behavior*, 9 (2002) 67-90.
12. Sutton, R.S., Barto, A.G. *Reinforcement learning: An introdu*ction. MIT Press, Cambridge, MA (1998).
13. Touretzky, D.S., Saksida, L.M. Operant conditioning in Skinnerbots. *Adaptive Behavior*, 5 (1997) 219-247.