Using a net to catch a mate: Evolving CTRNNs for the Dowry Problem

Elio Tuci* **Inman Harvey*** **Peter M. Todd****

*Centre for Computational Neurosciences and Robotics,
School of Cognitive and Computing Sciences
University of Sussex, Brighton BN1 9QH, United Kingdom
eliot, inmanh@sussex.ac.uk

**Center for Adaptive Behavior and Cognition
Max Planck Institute for Human Development, Lentzeallee 94, 14195 Berlin, Germany
ptodd@mpib-berlin.mpg.de

Abstract

Choosing one option from a sequence of possibilities seen one at a time is a common problem facing agents whenever resources, such as mates or habitats, are distributed in time or space. Optimal algorithms have been developed for solving a form of this sequential search task known as the Dowry Problem (finding the highest dowry in a sequence of 100 values); here we explore whether continuous time recurrent neural networks (CTRNNs) can be evolved to perform adaptively in Dowry Problem scenarios, as an example of minimally cognitive behavior [Beer, 1996]. We show that even 4-neuron CTRNNs can successfully solve this sequential search problem, and we offer some initial analysis of how they can achieve this feat.

1 Introduction

Life does not always present us with choices in the most convenient form. Rather than give us a supermarket of options to compare and choose between at our leisure — "pick a card, any card" — the world often conspires to confront us with a single possibility at a time — "take it or leave it". Another better option might come along later, but then again, it might not. This type of sequential choice problem can be encountered in searching for a job, or a place to live, or even a mate, when possibilities are seen one at a time and cannot be kept around for comparison to later options — the job, or house, or date you turn down today won't be waiting for you tomorrow. More generally, agents may have to deal with such sequential choice whenever resources they need are distributed in time or space: Should this prey item be pursued, or will a better one be found over the hill? Should this mate be courted, or will my genes stand a better chance with another individual later?

How can such a decision be made? This problem has been studied widely in statistics, economics, psychology, and other fields, and both optimal and non-optimal but reasonable approaches have been proposed, as we will discuss. But these proposals have all been made at the level of concrete symbolic information-processing algorithms: check this item, update that aspiration level. We wanted instead to find out whether lower-level subsymbolic dynamic systems of the sort proposed for biologically plausible agent architectures could also solve this problem, and explore what kinds of solutions evolution could find within this framework. Inspired by previous research on minimally cognitive behavior [Beer, 1996; Slocum et al., 2000], we set out to evolve continuous time recurrent neural networks (CTRNNs) that can deal with this important cognitive task of finding good options from sequences of possibilities.

1.1 The Dowry Problem

To study behavioral approaches to the problem of sequential choice, it is useful to take a well-specified form of the problem as a starting point. Probably the best-studied example of sequential search is captured in the Dowry Problem, in which the task is to select the woman with the highest dowry (money brought to a marriage) out of a sequence of 100, or its alternate form, the Secretary Problem, in which the goal is to select the best secretary (e.g. on typing speed) out of 100 applicants seen one at a time. In more detail (and with sex roles reversed and species changed for variety), the Dowry Problem goes like this:

A young female hangingfly wishes to find a mate, but she has many suitors to choose from. Male hangingflies attempt to woo her by offering her a nuptial gift (the equivalent of female dowries): an insect or other tasty morsel for her to dine on while the male goes about his inseminating business unnoticed. (From this point on we begin to take liberties with the true situation in nature for the purposes of formulating the problem properly — see Preston-Mafham and Preston-Mafham [1993] for the more peculiar real story.) She must choose a single male to mate with on the basis of the size of the nuptial gift he brings her. Her life is short, so she only has time to evaluate 100 of the eager males visiting her on her chosen leaf before she runs out of lifespan in which to mate successfully. Now her task in this problem is to select the one male, out of the 100 that she can possibly see, who has the biggest nuptial gift to offer her [1]. She can only see one male and his gift at a time, and then she must decide immediately if she thinks he is the one with the biggest nuptial gift out of all 100 males and accept his advances, or else rebuff him and go on to the next candidate. She cannot return to any male she has seen before — once she rejects them, they are gone forever. Moreover, the poor female has no idea of the range of gift sizes she might encounter, before she starts seeing the males — all of the males might be bringing fleas this season, or one might have bagged a whole rhino. What strategy can she possibly use to have the highest chance of picking the male with the biggest gift for her?

The Dowry Problem can be considered a formal model of many real-world decision problems, in which agents encounter options in temporal sequence, appearing in random order, drawn from a population with parameters that are partially or completely unknown ahead of time. What strategy can the agents possibly use to maximize the expected payoff or to minimize the expected cost associated with their choice? This general question has motivated research in statistics and probability theory [Ferguson, 1989], economics [Seale and Rapoport, 1997], and biology and psychology [Todd and Miller, 1999; Dudey and Todd, In Press]. Statisticians and economists tend to develop optimality theorems relevant to job search and consumer search. In biology and psychology, more psychologically plausible rules or heuristics for sequential choice have been proposed and investigated, decision algorithms that are not guaranteed to yield the highest likelihood of a best choice but which generally do a good job. Many of these studies focus on the problem of sequential mate search and mate choice, because of the evolutionary importance of this domain.

1.2 Structure of the paper

The Dowry Problem is a useful setting in which to explore the minimal cognitive mechanisms necessary to tackle the problem of sequential choice. To find such minimal cognitive mechanisms, we adopt the framework of evolutionary simulation, modelling the process of natural selection [Holland, 1975; DiPaolo et al., 2000] applied to dynamic neural networks as abstract instantiations of an agent's decision making structure. We take an evolutionary approach because it allows the exploration of possible cognitive architectures relatively unencumbered by *a priori* assumptions [Cliff et al., 1993; Cliff and Miller, 1996; Seth, 1998; Nolfi and Floreano, 2000]. As already pointed out by Slocum et al. [2000], these simple idealized models can serve as "frictionless planes" in which basic theoretical principles of the dynamics of agent-environment systems can be worked out.

In the next section, we present the analytic solution for the Dowry Problem and discuss the minimal cognitive requirements for this solution. In section 3 we describe our simulation for evolving Continuous Time Recurrent Neural Networks (CTRNNs) [Beer and Gallagher, 1992] with a genetic algorithm [Goldberg, 1989] to implement sequential choice strategies for the Dowry Problem. In section 4 the performance of the best evolved neural networks is analyzed and compared with the performance of some standard sequential choice algorithms. We demonstrate that the evolved CTRNNs can successfully deal with the slightly modified and more biologically plausible versions of the Dowry Problem in which perfection is not strictly required. Within these contexts, the neural network performance can actually surpass that of the standard strategies. Further analysis in section 5 gives some hints as to how the networks perform their search. Finally, we consider the implications of this work in section 6.

2 Solving the Dowry Problem

Roughly speaking, the Dowry Problem refers to a class of problems in which an agent has to maximize (or minimize) the expected payoff (or cost) given by choosing a single item from among a population of sequentially encountered items. The Dowry Problem is easy to state and has a striking solution. In its simplest form it has the following features[2]:

1. The agent (e.g., the female hangingfly) can make only one choice.

2. The number n of items in the population (e.g., number of males) is known.

3. The items are presented sequentially in random order, with each order being equally likely.

4. The items can be ranked from the best to the worst, without ties, according to a specific criterion (e.g.,

[1] Note that realistically she could do well to pick any male with a big-enough gift, and that evolution's goal would probably be to endow her with a way of picking large gifts on average. This is the payoff function we will explore in more detail later in the paper, but the original Dowry Problem is stated in terms of an all-or-nothing payoff, rewarding only selection of the single highest value.

[2] This formal description has been taken from [Ferguson, 1989] in which the problem is presented via its alter ago, the Secretary Problem.

size of nuptial gift). The decision to accept or reject an item must be based only on the relative ranks of those items presented so far.

5. Once an item is rejected, it cannot later be recalled (returned to).

6. The agent is very particular and will be satisfied with nothing but the very best item. (That is, the hangingfly's payoff is 1 — e.g., numerous offspring — if she chooses the best of the n males, and 0 — e.g., single death — otherwise).

7. If the agent does not make any choice before the end of the sequence, he or she must take the final option.

Agents facing the Dowry Problem seek perfection, with a payoff of one only for picking the very best item (highest dowry or biggest nuptial gift) and zero for picking anything else. They also ignore search costs such as time, ignore the problem of mutual choice (i.e., the possibility that the male hangingfly that the female selects will not agree to mate with her in return), and assume that they know the exact number of items that will be presented.

The solution to the Dowry Problem — that is, the strategy that gives the highest chance of selecting the single best option — requires sampling a certain initial subset of the population of items $r-1$ with $r \in [1, n-1]$, remembering the best of them, and then picking the next item that is even better [Ferguson, 1989]. It can be shown that for large n, it is optimal to wait until about 37% (i.e., $1/e$) of the population have been seen and then to select the next option encountered that is better than any seen previously. The probability of success in this case is also about 37%. For small values of n, the optimal r is the one that maximizes the probability $\phi_n(r)$, where

$$\phi_n(r) = (\tfrac{r-1}{n}) \textstyle\sum_{j=r}^{n} \tfrac{1}{j-1}$$

One simple way to accomplish this optimal procedure is to set an aspiration level equal to the quality of the best option seen so far, updating this aspiration level as necessary as each new option is seen; then after 37% of the population has been encountered, fix this aspiration level and use it to stop the search with the next option seen that exceeds that aspiration. This approach is akin to Herbert Simon's notion of *satisficing* [Simon, 1990], and is what Seale and Rapoport refer to as a *cut-off rule* [Seale and Rapoport, 1997]. Thus, given that the searching agent knows the optimal 37% sample size — which assumes that the searcher first knows the total number of options n that could be encountered — following this *37% rule* is simple. The searcher must only keep in memory the optimal (37%) sample size and the quality of the best option seen so far. With each new option seen, a simple pair of comparisons (whether the option's quality exceeds the aspiration level, and whether the sequence-position number of that option exceeds the 37% value) is sufficient to decide whether to stop or continue search.

In a repeated version of the Dowry Problem where an agent sees multiple sequences and must choose the best value in each one independently, the agent must also be able to respond differently to the same item seen in different sequences, as a consequence of the fact that the item's relative ranking will vary across the sequences. This is an instance of the problem generally known as *sensory aliasing* [Nolfi and Marocco, 2001]. Sensory aliasing refers to the situation wherein two or more agent/environment states correspond to the same sensor pattern but require different responses. An agent that repeatedly plays the Dowry Problem game might encounter an item which is the best choice in one population and also the worst in a different population. An optimal strategy in this case should aim to select the item in the former case, and reject the same item in the latter. The analytic solution given above explicitly includes the memory and "behavioral plasticity" that is required for an agent to perform optimally when faced with multiple cases of the Dowry Problem. Our abstract model of the agent's decision making structure (i.e., CTRNNs) does not explicitly include these requirements. The challenge is for evolution to shape such a dynamic system so that its performance will compare favorably with that of the optimal strategy.

3 Evolving CTRNNs for the Dowry Problem

3.1 Overview of the evolutionary scenario

We evolve a population of CTRNNs to maximize the expected payoff given by choosing one item among a sequence of 20 items, where payoff is proportional to the relative rank (on some criterion) of the item chosen. (Here we use the term "rank" in the opposite of its usual sense — higher rank-numbers are better, and give higher payoffs.) The items are presented (see section 3.4) sequentially in random order, each order being equally likely. The network has to "decide" immediately after an item has been presented whether or not to select that item. A binary thresholded output neuron signals the decision of the network (see section 3.3). If, after the item is presented, the output neuron holds an activation value of 1, this item is chosen. If the output neuron holds 0, a new item is presented. Once rejected, an item cannot be recalled. The presentation of items is stopped either when the network selects an item or after the last of the 20 items is introduced; if the network does not select any item before the end of the sequence, the last item is taken as the network's choice. Then, the network is scored according to the relative ranking of the chosen item (see section 3.5). After that, the network is reset (see section 3.3), and a new sequence can be presented.

Our evolutionary scenario differs from the Dowry Problem mainly in the rank-proportional payoff function used (as opposed to the original all-or-nothing payoff). While picking the single best item yields the highest payoff, choosing the second best gives a somewhat reduced (but non-zero) payoff, picking the third gives slightly less than the second, and so on. This "friendlier" payoff function was used both because it is more realistic (see section 4.1), and because it proved to allow for much easier evolution of the networks. Aside from this, our scenario follows the Dowry Problem, with items simply being integer numbers randomly drawn from a pre-defined range (see section 3.2). Each network is evaluated on 60 different populations of items; we refer to each presentation of a sequence from a population of items as a trial. The 60 trials differ in the range from which their 20 numbers are drawn. Between any two trials there may be some sensory aliasing, in that the same items can occur in each trial but with a different ranking (a specific example is shown in the next section).

3.2 How item values are drawn

The networks are run through 60 independent trials (T_c, $c = 1, ..., 60$), each of which is made up of 20 integers. The integers are randomly drawn without replacement from a uniform distribution on the interval $[l_c, h_c]$, with the lowest number l_c and the highest number h_c of the range defined by the following rule:

$$c = 1, ..., 60 \left\{ \begin{array}{c} l_c = c; \\ h_c = l_c + 29 \end{array} \right.$$

While the networks see the trials in the simulation in order from the first to the sixtieth, this order is irrelevant, because the networks are reset (see section 3.3) at the beginning of every trial. (This of course cannot be done so easily for human subjects in Dowry Problem experiments, where great care must be taken to randomize the order of presentation of multiple trials, or to limit the experiment to a very small number of trials.)

Among all the trials, the lowest number that the networks can possibly experience is $l_1 = 1$ and the highest is $h_{60} = 89$. Values between 20 and 70 can be both the lowest within a trial and the highest within a different trial, while values between 2 and 88 can have a different ranking in different trials. The greatest amount of sensory aliasing between two trials occurs when they are consecutive (e.g., trials T_c and T_{c+1}); if two trials are more then 30 steps away from each other (e.g., T_c and T_{c+30}), they are certain to be made up of completely different items, so that no sensory aliasing exists between them.

Table 1 shows an example of two possible sequences of item-values that could occur on the first trial (T_1) and the 28th trial (T_{28}). The value 28 is an item in both populations, but in T_1 it is the best item, whereas in T_{28} it is the worst, yielding a very different rank-based score if it is chosen in the two trials. This is an example of sensory aliasing, because the network should select this item in T_1 and reject it in T_{28}.

Rank	**values**		**Rank**	**values**	
	T_1	T_{28}		T_1	T_{28}
1	4	28	11	16	44
2	5	33	12	17	46
3	6	35	13	18	47
4	7	36	14	21	48
5	9	37	15	22	49
6	10	39	16	23	50
7	11	40	17	24	51
8	12	41	18	25	52
9	13	42	19	27	53
10	14	43	20	28	57

Table 1: Two example trials of 20 values each, showing sensory aliasing owing to the value 28 being the highest-ranked item in the first trial and the lowest-ranked item in the second.

3.3 The structure of the networks

We model the agent decision-making mechanisms as fully connected, 4 neuron CTRNNs. All neurons are governed by the following state equation:

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^{k} \omega_{ji} z_j + g I_i$$

$$\text{with } z_j = \frac{1}{1+exp[-(y_j+\beta_j)]}, \; i = 1, ..., 4$$

where, using terms derived from analogy with real neurons, y_i represents the cell potential, τ_i is the decay constant, ω_{ji} is the strength of synaptic connection from neuron j to neuron i, k corresponds to the number of input connections to neuron i both from other neurons and from itself, z_j is the firing rate, I_i is the intensity of the sensory perturbation on sensory neuron i, g is the sensory gain factor, and β_j is a bias term. Only one neuron actually receives sensory perturbation in the networks we started with. There is also one output neuron that registers the network response. This neuron has a thresholded binary activation function: it will output 0 if its y-value ("cell-potential") is less than 0.5, otherwise it will output 1. The strength of synaptic connections ω_{ji}, the decay constant τ_i, the bias term β_j, and the gain factor g are genetically encoded parameters. Activation levels are integrated using the forward Euler method with an integration step-size of 0.2, which means that there are 5 updates of the network per second. To reset the network, all unit states are set to 0.

3.4 How items are presented to the network

Among the several possible ways in which an input can be presented to a network, we chose to use time-based input. While this is to some extent an arbitrary choice for our initial explorations described here, time (e.g. duration of sensory input) may correspond to magnitude in many natural situations. For instance, a bee can judge the size of a potential new hive site by the time it takes to fly from one side to the other, and a robot judging lengths of walls during navigations can use the time taken to pass along them. With such time-based input for the CTRNNs, the value of each item seen specifies the lapse of time in seconds (multiplied by 5 to give the number of network updates) during which the network external input I_i is set to 1 (see the network state equation in the previous section).

Thus, each trial proceeds as follows. At the beginning of each trial the network is reset. Then, before each item is introduced, the network is cleared for 2 seconds by setting the network external input I_i to 0 (where 2 seconds corresponds to 10 updates of the network's state). After that, the network is presented with the current item by setting the network external input to 1 for s seconds, with s equal to the value of the item being presented. After presentation, the network's output is checked to see if the item is accepted or not. If the item is rejected, the network is again cleared for 2 seconds, and then the next item is introduced.

3.5 How network fitness is scored

Each network is evaluated based on the items it has chosen in the 60 trials. Choices are made either by the network expressing its preference for some item, or by the network not expressing any preference and therefore being assigned the last presented item of the current trial as its choice. For each single trial, the payoff corresponds to the normalized relative ranking of the selected item. For example, assume that the network selects the item that holds value 13 among the population of items that made up trial T_1 in Table 1. The network's payoff for this trial corresponds to $P_1 = 9/20$ because 9 is the relative ranking of the selected number, 13, in that trial.

The total payoff $\bar{P}$ for each network is calculated by averaging the payoff obtained across the 60 trials:

$$\bar{P} = \frac{1}{60}\sum_{c=1}^{60} P_c, \quad c = 1, ..., 60.$$

3.6 The genetic algorithm

A real-valued steady-state genetic algorithm [Goldberg, 1989] was used to evolve the CTRNN parameters. A population of 50 individuals was maintained, with each individual's 4-neuron network encoded as a vector of 25 real numbers (16 connections, 4 decay constants, 4 bias terms, 1 gain factor). Initially, a random population of vectors was generated by setting each component of every individual to a random value uniformly distributed over the range [0, 1]. Individuals were selected for reproduction using a linear rank-based method with implicit elitism. Each vector component in a newly-created individual was mutated with a probability of 0.2 using a "creep" mutation operator which added a random number chosen uniformly from the range [-0.2, +0.2]. During evolution, vector component values could not move outside the range [0, 1]. Recombination when creating offspring was applied with a probability of 0.3.

Genetically encoded values were linearly mapped into CTRNN parameters with the following ranges: biases $\beta_j \in [-2, 2]$; connection weights $\omega_{ji} \in [-5, 5]$; and gain factor $g \in [1, 7]$. Decay constants were first linearly coded in the range $\tau_i \in [0, 2.8]$ and then exponentially mapped into $\tau_i \in [10^0, 10^{2.8}]$. These parameter ranges were chosen on the basis of having proven useful in other CTRNN experiments. The large range of the exponentially mapped decay constants is meant to allow for evolution to select both neurons that tend to change their state (cell potential) radically every time step (i.e., neurons with small decay constants), and neurons that tend to change their state only minimally every time step (i.e., those with large decay constants).

i	1	2	3	4
τ_i	1.0	330.161170	1.071498	42.389673
β_i	-1.032616	0.187748	0.482778	-1.282664

Table 2: The table above gives the decay constants τ_i and the bias terms β_i, for each neuron of the best evolved network. The gain factor g is equal to 1.

4 Results

Ten evolutionary simulations were run for 5000 generations each with 50 networks, and each network was assessed on all 60 sequences. We tested the best network of the final generation from each of these runs to establish how well these networks perform in the Dowry Problem and in another series of slightly different test problems. These problems differ from the standard Dowry Problem in the way in which payoffs are determined for chosen items, as explained in the next section. The very best network from all 10 simulation runs is shown in figure 1 with its connection weights ω_{ij}, while table 2) shows its decay constants τ_i, bias terms β_i, and gain factor g. This network's performance is compared with the average performance of the best networks from the 10 runs and with the performance of a set of more standard cutoff rules in section 4.2.

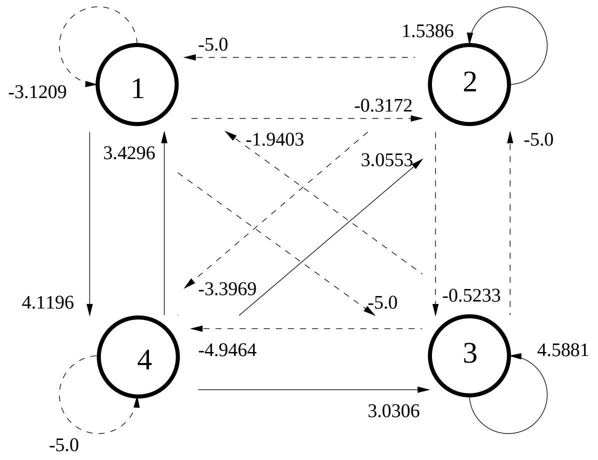

Figure 1: Morphology and connection weights of the best evolved network. The continuous lines indicate excitatory connections (positive weights), whereas dashed lines indicate inhibitory connections (negative weights).

4.1 Evaluation method

To evaluate the performance of the evolved networks, we developed an extended set of test problems because the Dowry Problem itself is probably not the most realistic or common form of sequential choice task that real agents could encounter. In the Dowry Problem, the single best option is the only good one — the only option that receives a non-zero payoff. But this very strict payoff function would be found in few natural situations.

In many species, many (if not most) animals find some mate, some food source, and some place to live, and thus receive some payoff in these search domains, even if they are not getting the highest possible payoff. Thus in these cases, a payoff proportional to the quality of the alternative chosen (e.g., proportional to the size of the nuptial gift selected, and as used in our fitness function for evolving the CTRNNs) is more appropriate than the all-or-nothing payoff of the standard Dowry Problem. In other species, the available alternatives (e.g., mates or habitats) may be limited so that not all individuals succeed in making a viable choice — for instance, only a quarter of the possible habitats in which one could settle may provide enough resources to raise offspring. For such cases, the search payoff function might fully reward only choices made in the top 25% of all available alternatives and give zero payoff to all other choices. Based on such considerations, we used the following set of test problems and associated payoff criteria [Todd and Miller, 1999]:

1. Dowry Problem — For each trial, the payoff is 1 for selecting the single best item, and 0 otherwise.
2. Top 10% Problem — For each trial, the payoff is 1 for selecting an item in the top 10% of the population of items for that trial, and 0 otherwise.
3. Top 20%, 30%, 40%, 50% Problems — As above, with payoff 1 for selecting an item in the top n% of the population, and 0 otherwise.
4. Bottom 20% Problem — For each trial, the payoff is 1 for selecting an item that is *not* in the bottom 20% of the population of items, and 0 otherwise.
5. Bottom 10% Problem — For each trial, the payoff is 1 for selecting an item that is *not* in the bottom 10% of the population of items, and 0 otherwise.
6. Maximize Mean Ranking Problem — For each trial, the payoff is proportional to the ranking of the value picked, calculated as explained in section 3.5.

Note that the last payoff function is the same as the fitness function that was used during network evolution; all of the other performance measures are testing the ability of the CTRNNs to generalize to new (though related) tasks. (Each network and strategy was evaluated on all performance measures simultaneously — that is, each choice made was evaluated for whether it was the single best, and in the top 10%, and bottom 20%, etc.)

To provide a benchmark for evaluating the performance of our best evolved networks, we also tested a range of cutoff rule strategies as introduced in section 2. These cutoff rules (as described in section 2) require checking a certain number $r-1$ of items from the population with $r \in [1, n-1]$ (where n is the number of items in the whole population), remembering the best of those, and then choosing the next item seen that is better than the best seen so far [Seale and Rapoport, 1997]. Varying the parameter r across its whole range, we get $n-1$ possible different cutoff rule strategies, or 19 different cutoff rules for our population of 20 items. For example, the cutoff rule defined by setting $r = 1$ samples 0 items (so that the best seen so far is also 0, or undefined), and hence always chooses the first presented item. This yields random performance in the Dowry Problem, which is a useful benchmark. The cutoff rule defined by setting $r = 11$ samples 10 items, remembers the best of them, and then picks the next seen (starting with item 11) that is even better. If no subsequent item is seen that is better than the best encountered during the initial sample (that is, no further item exceeds the aspiration level set), then the network "picks" the last item presented as its choice.

It is important to point out that these cutoff rules are not necessarily the optimal solution for each of our particular test problems, but they are a simple strategy that agents can use (and that people seem to use–see Seale and Rapoport [1997]). Thus, by finding the best cutoff rule for a particular test (i.e., by finding which value of r leads to the highest performance), we can establish a useful, and psychologically plausible, benchmark

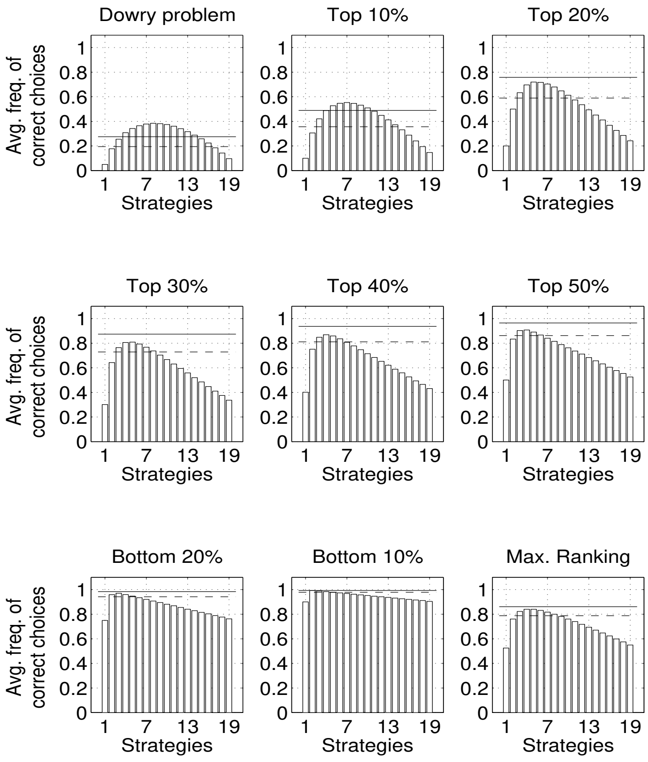

Figure 2: Performance on the different test problems for all cutoff rule strategies (white bars) compared with the single best evolved network (continuous lines) and the average performance of the best evolved networks from the 10 runs (dashed lines). For all graphs, the x-axis indicates the r value for the cutoff rules, while the y-axis shows the average payoff, calculated according to the particular test problem definition.

for comparison with the evolved networks. The optimal solutions are actually difficult to find; for the task of minimizing the mean rank selected (here with rank 1 being best), it is known that the optimal solution will find a rank of 4 or better on average [Chow et al., 1964], yielding another rough benchmark of 85% performance on the Maximize Mean Ranking Problem.

4.2 Comparing performance

Figure 2 shows the performance on the different test problems of the 19 cutoff rule strategies compared with that of the very best evolved network and the average performance of all the best evolved networks from the 10 simulation runs. In the Dowry Problem, the performance of the best evolved network does not match that of the best cutoff rule (for $r = 8$), achieving 27% success rate instead of the optimal 37%.

However, when perfection is not the goal (as it is for the Dowry Problem), the performance of our best evolved network gets more impressive. In the Top 10% Problem the performance of the best evolved network comes very close to the performance of the best cutoff rule strategy. Even more surprisingly, for all the other test problems the best network overcomes the performance of the best cutoff rule. How can this be?

5 Analyzing the best evolved network

It should be recognized that CTRNNs, with multiple recurrent connections and no a priori division into modules, are inherently difficult to analyze. There may be no direct translation of the network behavior into a strategy described in terms of simple rules. However, we can look into how the networks process their input information both behaviorally (e.g., by testing their choices in different experimental settings) and "neurologically" (e.g., by monitoring changes in internal activation levels over time) and see if this suggests any rules that can summarize their behavior.

To try to understand how the best evolved network can perform so well on search criteria related to the Dowry Problem, we should first consider how the problem we created differs from the true Dowry Problem. First, we always present values from a uniform distribution; second, we always present values in one sequence (trial) from within a 30-number span; third, the evolving networks only ever see numbers from 1-89. Given these restrictions, the evolutionary process can build a more specific solution to the search problem it faces. For instance, one approach would be to use the first few (even just two or three) values seen in a trial to compute an estimated mean of the range for this trial, and then add 12 (a bit less than half of the full range of 30) to this estimate to create the aspiration level for use in further search. We have not yet determined the strategy used by the best evolved network to this level of detail, but we have begun to take steps in that direction, as we now describe.

The first aspect of the network's behavior that we looked at is the percent of choices that are made at each relative rank within each of the 60 trials (see Figure 3), both for the best evolved network and for the best cutoff rule for the original Dowry Problem ($r = 8$). Choices here are both those that the network or strategy actively makes, and those that it is passively assigned when it must take the final presented option by default. Figure 3–Network shows that the best evolved network chooses the best item (rank 20) with the highest frequency throughout all the trials, regardless of the actual value of the item itself. The network also performs slightly better within the range of values that can be found in trials 10 to 40. Figure 3–Strategy shows that the cutoff rule distributes its choices across the relative ranks in a similar, but more consistent, manner, and with more of an emphasis on the single best value.

We focus in on explicitly expressed preferences (i.e., not including default final-value choices) in Figure 4,

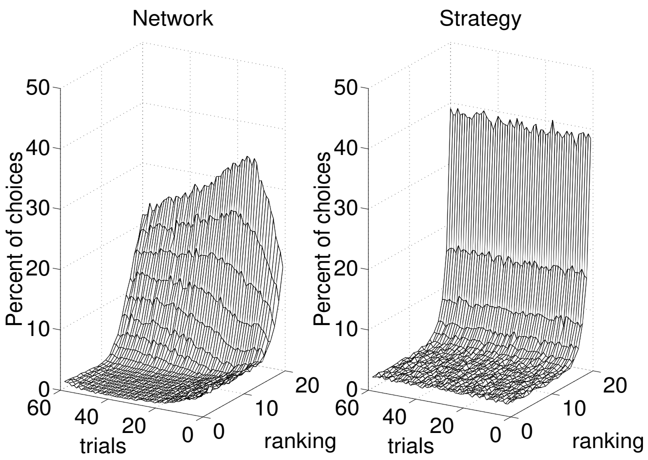

Figure 3: Percent of choices made per rank and per trial by the best evolved network (on the left) and by the cutoff rule with $r = 8$ (on the right). Ranking 1 refers to the item with the lowest (worst) value, while ranking 20 refers to the item with the highest (best) value.

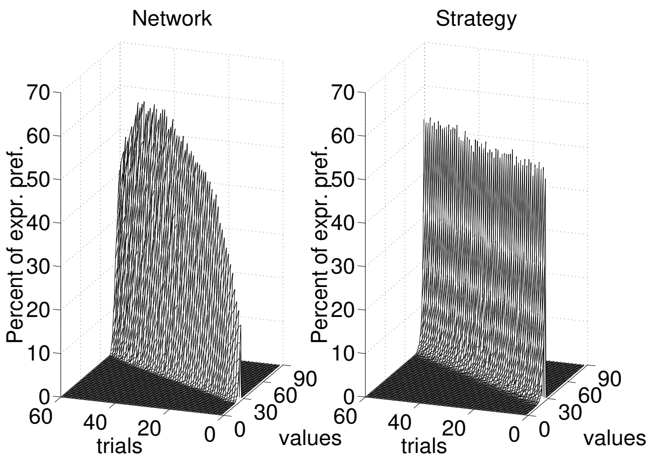

Figure 4: Percent of expressed preferences (actively-made choices) per item value and per trial by the best evolved network (on the left) and by the cutoff rule with $r = 8$ (on the right).

where we show the percentage of trials on which particular values (rather than ranks) are selected by the best network and cutoff rule. This graph makes it clear that the network tends to choose particular item values only when they are among the highest within the current population, thus overcoming the sensory aliasing problem. For example we can see from figure 4–Network that values around 30 were selected more frequently when they appeared in the first 10 trials than when they appeared in trials 20 and above.

Finally, figure 5 helps us to speculate on the possible approach that the network might have evolved. To understand the behavior of the network we refer to four different features of its decision making recorded during the evaluation test. Figure 5a shows the percent of expressed preference per trial. Figure 5b shows the average position at which a preference is expressed per trial. Figure 5c shows the average number of seconds of presentation of the ultimately-selected item in a given trial (that is, the amount of time that the input neuron receives input=1) until the activation level of the network's output neuron goes to 1. This is essentially the aspiration level or threshold that the network has set, based on the items it has seen so far in this trial, at the time that it makes its active choice. Graph d shows the average total number of seconds of input=1 from the beginning of each whole trial to the time when the activation level of the network's output neuron goes to 1 (making its choice) in each single trial. This is essentially how much "history" (in the sense of total summed input values) is required within a trial before the network makes an active choice.

Bearing in mind that the average value of the trial-population of items increases from trial 1 to trial 60, we can then describe and summarize the results shown in figure 5 in a few concepts: First, the network is not always expressing its preference with the same frequency (see Figure 5a). Looking at figure 5a, we can see that, from trial 1 to trial 15, the conditions triggering the network choice occur more often with later trials. After trial 15 these conditions seem always to be satisfied by the distribution of item values, until about trial 50, when the network starts getting more silent and expresses fewer explicit preferences. For the moment, we leave out any explanation of this phenomenon, but it will be treated more extensively in the next section.

Secondly, the response of the network is triggered by somewhat different conditions in each trial (see Figure 5b,c,d). These conditions are defined by the item values seen before a choice is made. Recall that the network experiences the item values in terms of 10 updates before each item with the external input set to 0, followed by a number of updates (5 times the item value) with the external input set to 1. Figure 5c then shows that the aspiration level of the network (in terms of amount of time or number of updates over which the selected item is presented before it is selected) goes up as the trials, and hence population values seen, go up. This is to be expected, in order for the network to perform well; but the increase in the aspiration level is not strictly proportional to the increase in the values in the population, with the first trial (values between 1 and 29) leading to an average aspiration level of about 25, while the last trial (with values between 60 and 89) leads to an aspiration level of about 80 (rather than 85). This can also explain why choices are made later in each trial for the early trials, and sooner in each trial for the later trials (Figure 5b). Finally, Figure 5d shows that, if we

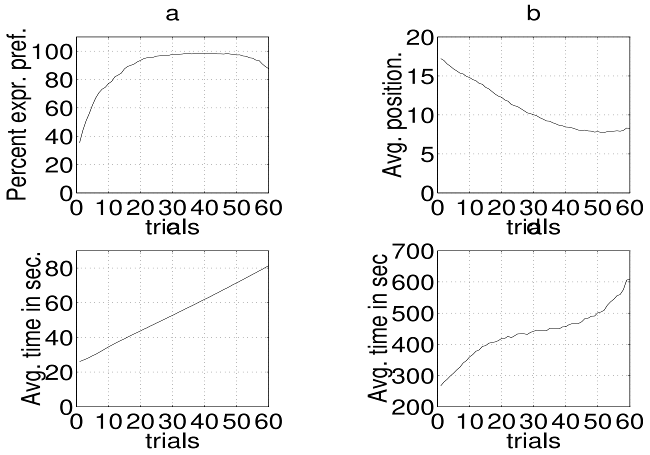

Figure 5: a. Percent of trials in which the best network actively expressed a preference, versus trial number. b. Average position at which an actively expressed choice is made per trial. c. Average number of seconds of input=1 from the beginning of the presentation of the selected item to the time when the activation level of the network's output neuron goes to 1. d. Average number of seconds of input=1 from the beginning of each whole trial to the time when the network output neuron's activation level goes to 1.

consider the sum of values seen before an active choice is made, this sum goes up with later trials; however, this is counteracted by the fact that fewer items are needed to create these higher sums in later trials (so again as shown in Figure 5b, with later trials fewer individual items must be seen before a choice is made).

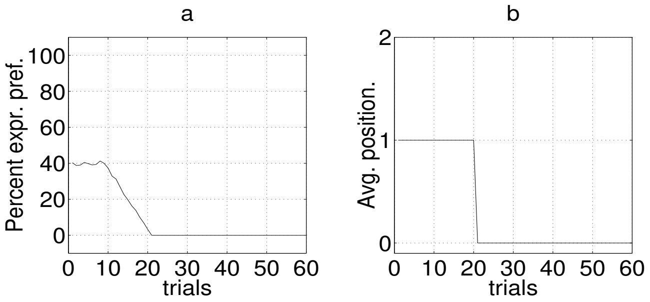

Figure 6: Robustness of the best evolved network. a. Percent of expressed preference per trial. b. Average position at which a preference is expressed in each trial.

5.1 *Robustness of the best evolved network*

We have carried out further tests to explore the behavioral robustness of the best evolved network. We ran the network 5000 times in an "unknown" Dowry Problem scenario, where the item values were taken from a distribution that had not been seen by the network nor by any of its ancestors. Values were drawn from a uniform distribution on the interval $T_c \in [l_c, h_c]$, with the lowest number l_c and the highest number h_c of the range defined as follows:

$$c = 1, ..., 60 \begin{cases} l_c = 89 + c; \\ h_c = (l_c) + 29; \end{cases}$$

The message from the graphs in figure 6 is quite clear. The network starts expressing its preference with a rather low frequency (figure 6a), always selecting the first presented item (figure 6b). The frequency of expressed preference continues to decrease until trial 20 when it reaches 0 (figure 6a). At this point, the network stops making active choices (and then gets the last value as a default choice, yielding random performance on the task).

This message is clear, but it is also counterintuitive. We expected a different outcome, because we thought that any value that can trigger the network's response in trial 1 in this problem will appear more frequently on the following trials as the values get higher, so the rate of active choice should go up. Also, we thought that higher values should produce more active choice. Why was this not the case?

Part of the explanation may lie in the way that we interpret the output of the network. Bearing in mind that the network's output neuron — with which the network expresses its active choices — is checked only at the *end* of any item presentation, the behavior of the network may contradict our intuition because the output neuron's activation value need not rise to 1 and stay there during an item's presentation. That is, after the output neuron activation has been set to 1 by the conditions that trigger an active network choice, it may return to 0 if the network continues to be updated longer with the external input set to 1. Then, when the output level is checked at the end of the item's presentation, it may be back to 0, indicating that this item should not be chosen, even though it was at 1 earlier (which could have indicated active choice). Of course, we can get around this by having the unit act as a toggle, so that once output is 1, the choice is made immediately; this interpretation of the output unit must be tried during evolution to see what effect it has on the ability of these networks to generalize to other value ranges.

This same phenomenon may be responsible for the decrease in frequency of expressed preference shown by the best evolved network during the last 10 trials of the evaluation test (see figure 5a). In these trials, the item values are certainly the largest ever experienced by the network (and its ancestors). For particular orders of presentation, these items might cause the network to output 0 (do not choose) at the end of a particular item's presentation, even though the network might have output 1 (choose this item) during the presentation.

6 Conclusions

We have shown here that simple (or at least small) dynamic neural networks, shaped by evolution, can successfully solve the important adaptive problem of sequential choice, as embodied in variants of the Dowry Problem. This work demonstrates that CTRNNs are capable of exhibiting another form of minimal cognitive behavior [Beer, 1996; Slocum et al., 2000] — a behavior that agents in the real world face whenever they must find resources distributed in time or space. The surprising finding here is that the evolved networks can even outperform simple cutoff rules that usually do very well on such search problems; this performance advantage is pronounced when more biologically realistic forms of payoff (such as rank-proportional payoff) are employed. In addition, our results showed that the evolved CTRNNs can overcome a form of the sensory aliasing problem: the networks employ a "behavioral strategy" that is plastic enough to respond differently to the same value seen in different contexts determined by the current item population.

The reasons for this surprising performance must still be uncovered; so far, we have just begun to analyze how the networks achieve this feat. While our analyses to date have helped us to begin to see how the best evolved network is operating, several further tests are needed. In particular, we want to see how the network's aspiration level changes as a result of particular sets of input values it sees. For instance, does seeing the values {12, 18} result in the same aspiration level as {18, 12} or {12, 15, 18}? (It does not in human experiments.) How do the activation levels of the individual neurons change over time with presentation of a given value? Does the network seem to keep track of minimum or maximum values seen, or the average of all values, or the range of all values, or some combination of these? (We can try to determine this by a combination of manipulating the network's input and looking at what the neurons are doing.) By gathering these additional observations, we can home in on the kinds of strategies that this best network and other evolved networks are using. As a further step, we can then compare the mechanisms used by the networks with those used by people in similar experimental settings, as a way to try to understand more about how real biological agents deal with sequential search and choice problems.

Future work will explore the applicability of sequential choice CTRNNs for search problems influenced by other factors such as changing environments (e.g., where the population of values goes up or down in overall quality over time as the search is progressing) or multiple cues (e.g., where the agent learns about different features of each item rather than about each item's criterion value directly). These settings will allow our approach to be generalized to more biologically realistic settings including mate choice and habitat choice, enabling us to further extend the range of cognitive behaviors that CTRNNs can model beyond the merely minimal.

Acknowledgements

The authors wish to thank all the members of the Centre for Computational Neuroscience and Robotics (http://www.cogs.sussex.ac.uk/ccnr/), the ABC Research Group (http://www.mpib-berlin.mpg.de/abc) for constructive discussion and the Sussex High Performance Computing Initiative (http://www.hpc.sussex.ac.uk/) for computing support. We also thank two anonymous reviewers for their useful comments.

References

Beer, R. D. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In Maes, P., Mataric, M., Meyer, J., Pollack, J., and Wilson, S., (Eds.), *From Animals to Animats IV: Proc. of the 4^{th} Intl. Conf. on Simulation of Adaptive Behavior*, pages 421–429. Cambridge, MA: MIT Press.

Beer, R. D. and Gallagher, J. C. (1992). Evolving dynamic neural networks for adaptive behavior. *Adaptive Behavior*, 1(1):91–122.

Chow, Y. S., Moriguti, S., Robbins, H., and Samuels, S. M. (1964). Optimal selections based on relative ranks. *Israel Journal of Mathematics*, 2:81–90.

Cliff, D., Harvey, I., and Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):73–110.

Cliff, D. and Miller, G. (1996). Co-evolution of pursuit and evasion II: Simulation methods and results. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson., S., (Eds.), *From Animals to Animats IV: Proc. of the 4^{th} Intl. Conf. on Simulation of Adaptive Behavior*, pages 506–515. Cambridge, MA: MIT Press.

DiPaolo, E. A., Noble, J., and Bullock, S. (2000). Simulation Models as Opaque Thought Experiments. In *Artificial Life VII: The 7^{th} Intl. Conf. on the Simulation and Synthesis of Living Systems.*

Dudey, T. and Todd, P. M. (In Press). Making good decisions with minimal information: Simultaneous and sequential choice. *Journal of Bioeconomics.*

Ferguson, T. S. (1989). Who Solved The Secretary Problem? *Statistical Science*, 4(3):282–296.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA:Addison-Wesley.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University Press.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* Cambridge, MA: MIT Press/Bradford Books.

Nolfi, S. and Marocco, D. (2001). Evolving robots able to integrate sensory-motor information over time. *Theory in Biosciences*, 120(3-4):287–310.

Preston-Mafham, R. and Preston-Mafham, K. (1993). *The encyclopedia of land invertebrate behavior*. Cambridge, MA: MIT Press.

Seale, D. A. and Rapoport, A. (1997). Sequential decision making with relative ranks: An experimental investigation of the "secretary problem". *Organizational Behavior and Human Decision Processes*, 69(3):221–236.

Seth, A. (1998). Evolving Action Selection and Selective Attention Without Actions, Attention, or Selection. In Meyer, J.-A. and Wilson, S. W., (Eds.), *From Animals to Animats V: Proc. of the* 5^{th} *Intl. Conf. on Simulation of Adaptive Behaviour*, pages 139–147. Cambridge, MA:MIT Press.

Simon, H. A. (1990). Invariants of human behavior. *Annual Review of Psychology*, 41:1–19.

Slocum, A. C., Downey, D. C., and Beer, R. D. (2000). Futher Experiments in the Evolution of Minimal Cognitive Behavior: From Perceiving Affordances to Selective Attention. In Meyer, J.-A., Berthoz, A., Floreano, D., Roitblat, H. L., and Wilson, S. W., (Eds.), *From Animals to Animats VI: Proc. of the* 6^{th} *Intl. Conf. on the Simulation of Adaptive Behavior*, pages 430–439. Cambridge, MA: MIT Press.

Todd, P. M. and Miller, G. F. (1999). From Pride and Prejudice to Persuasion: Realistic Heuristics for Mate Search. In Gigerenzer, G., Todd, P. M., and the ABC Research Group, *Simple Heuristics That Make Us Smart*, chapter 13. Oxford University Press, New York.