Optimisation in 'Self-modelling' Complex Adaptive Systems

Richard A. Watson¹, C.L. Buckley², Rob Mills¹

¹Natural Systems group, Electronics and Computer Science, University of Southampton, U.K. raw@ecs.soton.ac.uk.

²Informatics, Sussex University, U.K.

When a dynamical system with multiple point attractors is released from an arbitrary initial condition it will relax into a configuration that locally resolves the constraints or opposing forces between interdependent state variables. However, when there are many conflicting interdependencies between variables, finding a configuration that globally optimises these constraints by this method is unlikely, or may take many attempts. Here we show that a simple distributed mechanism can incrementally alter a dynamical system such that it finds lower energy configurations, more reliably and more quickly. Specifically, when Hebbian learning is applied to the connections of a simple dynamical system undergoing repeated relaxation, the system will develop an associative memory that amplifies a subset of its own attractor states. This modifies the dynamics of the system such that its ability to find configurations that minimise total system energy, and globally resolve conflicts between interdependent variables, is enhanced. Moreover, we show that the system is not merely 'recalling' low energy states that have been previously visited but 'predicting' their location by generalising over local attractor states that have already been visited. This 'self-modelling' framework, i.e. a system that augments its behaviour with an associative memory of its own attractors, helps us better-understand the conditions under which a simple locally-mediated mechanism of selforganisation can promote significantly enhanced global resolution of conflicts between the components of a complex adaptive system. We illustrate this process in random and modular network constraint problems equivalent to graph colouring and distributed task allocation problems.

Local constraint satisfaction and associative memory

Many natural dynamical systems have behaviours that can be understood as the local minimisation of an energy or potential function [1]. The Hopfield network [2] is a well-understood exemplar of such dynamical systems, exhibiting only point attractors, that has provided a vehicle for studying dynamical systems across many disciplines (Eq. 1). In this paper we investigate the interaction of two well-known properties of complex systems that have each been independently well-studied in the Hopfield network: i) The energy minimisation behaviour of dynamical systems [2] which can be interpreted as a local optimisation of constraints [3,4], and ii) Hebbian learning [5] with its capacity to implement associative memory [2,6]. The former is analogous in some circumstances to the behaviour of multiple autonomous agents in a complex system, such as servers in a grid computing system or people in a social network, attempting to maximise productivity or consensus using local rules, given intrinsic pair-wise constraints. The latter is generally assumed to be relevant only to neural networks and cognitive systems – but in this paper we introduce the idea of implementing associative memory in a distributed complex adaptive system and discuss its effects on system behaviour.

The dynamics of a Hopfield network, consisting of N discrete states $s_i = \pm 1$, can be described by updates to individual states:

$$S_{i}(t+1) = \theta \left[\sum_{j}^{N} \omega_{ij} S_{j}(t) \right]$$
(1)

where ω_{ij} are elements of the connection matrix Ω , and θ is the Heaviside threshold function (taking values -1 and +1 for negative and positive arguments respectively). The Hopfield network is run by repeatedly choosing a unit, *i*, at random and setting its state according to the above formula. Hopfield showed that if the connection matrix is symmetric, $\omega_{ij} = \omega_{ji}$, and under suitable constraint on the self-weights (here $\omega_{ii}=0$), all trajectories described by Eq. 1 converge on point attractors which are minima of the energy function given by:

$$E_{s} = H(S(t), \Omega) \equiv -\sum_{ij}^{N} \omega_{ij} s_{i}(t) s_{j}(t)$$
⁽²⁾

Consequently one can describe the asymptotic behaviour of such a network in terms of a process that locally minimises this function. The energy function (Eq.2) intuitively corresponds to the sum of 'tensions' in all state variables, or the degree to which influences from other state variables act to oppose the current state. A state change under Eq.1 necessarily resolves more constraints than it violates (under the Hopfield conditions), i.e. creates a net reduction in tensions, and reduces the total energy of the system. Minima in this function thus correspond to attractors in the network dynamics that are locally optimal resolutions of the opposing influences between variables or of the systems' constraints [3,4] (see also the stochastic counterpart, the Boltzmann machine [7,8]). However, networks with interactions that are difficult to resolve simultaneously create a dynamical system that has a large number of local attractors. In this case, when the state configuration of the system is set to an arbitrary initial condition and allowed to relax to an attractor it will generally not result in a configuration that is globally minimal in energy, or a globally optimal resolution of constraints [9].

In a quite unrelated scenario, training a dynamical system to have a particular energy function may be interpreted as a model induction process which takes as input a set of points in configuration space, 'training patterns', and returns a network which 'models' those points by exhibiting point attractors that correspond to those configuration patterns. The system may then act as an associative or content addressable memory [2] which takes as input a partially specified or corrupted input pattern and 'recalls' the training pattern that is most representative of that input pattern. A Hopfield network may be trained to implement such a dynamical system via Hebbian learning [5], i.e. the distributed application of Hebb's rule to all connections in the system (i.e. the change in weight, $\Delta \omega_{ij} = \delta s_i s_j$, $\delta > 0$). That is, for all ω_{ij} , $i \neq j$:

$$\omega_{ii}(t+1) = \omega_{ii}(t) + \delta s_i(t) s_i(t)$$
(3)

where $\delta > 0$ is a constant controlling the learning rate. During training, s_i and s_j represent the states of a given training pattern and each pattern in the training set is presented in turn repeatedly.

Although the energy minimisation behaviour of the Hopfield network and its interpretation as a local optimisation process is well known [3,4,1], and similarly, the ability of Hebbian learning to implement an associative memory of a set of training patterns and 'recall' them or 'recognise' them from noisy or partial examples is also well known [2], the idea of combining these behaviours in the same network may seem unnatural. In the former, the weights of the network take fixed values that represent the constraints between variables in a combinatorial optimisation problem, the objective is to discover configurations that optimisation. In contrast, in training an associative memory, the weights of the system are initially neutral ($\omega_{ij}=0$) but change over time such that local optima are created that represent the patterns to be stored. These seem like incompatible objectives [12]. For example, if a network is being used to find solutions to a combinatorial optimisation problem then it does not obviously make sense to change the weights that represent the problem. Can it be useful for a network to store patterns at the same time as recalling patterns?

In this paper we show that under certain conditions the application of associative memory and repeated energy minimisation in the same network on different timescales creates a positive feedback process that significantly improves the constraint satisfaction ability of the system – i.e. enhances its ability to find configurations that minimise constraints between system variables and globally minimise energy. We refer to this as a 'self-modelling' complex adaptive system.

The concurrent evolution of state dynamics and changes in weights has been studied in the Hopfield network previously [11], and more generally, the notion that network topology affects behaviours on a network and, vice versa, that behaviours on a network may affect network topology, is gaining increasing attention [10]. The current paper illustrates conditions where self-organisation of topology in an adaptive network alters its ability to minimise energy and hence its optimisation capabilities.

A 'self-modelling' dynamical system

We consider systems with the following conditions: 1) The initial dynamics of the system (given the initial connections between variables) exhibits multiple point attractors; 2) The system configurations are repeatedly relaxed from different random initial conditions such that the system samples many different attractors on a timescale where connections change slowly. 3) The system spends most of its time at attractors.

The first condition is consistent with scenarios where the initial network represents a difficult optimisation problem [3,4]. To satisfy condition (2), the system takes a random state configuration, $R=\{-1|1\}^N$, every τ time steps (state updates). We refer to the behaviour of the network between these perturbations as a *relaxation* of the network. This effects multiple attempts at solving the optimisation problem. The second condition also asserts that δ is sufficiently small that the distribution of attractor states visited changes slowly – we show examples of this below. The third condition above asserts that $\tau >> t^*$, where t^* is the number of time steps required to reach a local attractor state.

Under these conditions the state configurations that are experienced most often will be the attractor states of the system's own dynamics, and as changes to connections are slowly accumulated, these modifications to the network constitute an associative memory of its own attractor states. What does it mean for a system to 'learn' its own attractor states? From a neural network learning point of view, a network that forms a memory of its own attractors is a peculiar idea (indeed, the converse is more familiar [13]). Forming an associative memory means that a system forms attractors that represent particular patterns or state configurations. For a network to form an associative memory of its own attractors therefore seems redundant; it will be forming attractors that correspond with attractors that it already has. However, in accumulating weight changes that constitute an associative memory of the original attractors, the system will nonetheless alter its attractors; it does not alter their positions in state configuration space, but it does alter the size of their basins of attraction (i.e. the set of initial conditions that lead to a given attractor state via local energy minimisation).¹

Specifically, the more often a particular state configuration is visited the more its basin of attraction will be enlarged and the more it will be visited in future, and so on. Because every initial condition is in exactly one basin of attraction it must be the case that some attractor basins are enlarged at the expense of others. Accordingly, attractors that have initially small basins of attraction will be visited infrequently and as the basins of other attractors increase, these attractors will decrease. Eventually, with continued positive feedback, one attractor will out-compete all others and there will only be one attractor remaining in the system.

But what has this got to do with resolving the constraints that were defined in the original weights of the system? To understand the relationship between the original constraints of the system and the new dynamics of the system given its learned connections, it is informative to define the 'true' or original energy, E_S^0 , defining the degree to which a configuration of the system successfully resolves the original constraints between problem variables, using $\alpha_{ij} = \omega_{ij}(t=0)$, as follows:

$$E_S^0 \equiv H(S, \Omega(t=0)) = -\sum_{ij}^N \alpha_{ij} s_i s_j$$
⁽⁴⁾

It is this original energy function (solid curve Fig.1) that we are interested in minimising in order to resolve the true constraints between problem variables. In contrast, ω_{ij} and Eq. 2 define a modified or augmented energy function (dashed curves Fig. 1) which determines the behaviour of the system at any given time, as per Eq.1. The behaviour of the system given the modified connections will, in general, be different from that given the original connections. Thus, the true energy of configurations that the system reaches may change over time, not because the true energy of any given configuration is different, but because the distributions of configurations that are visited is different. In particular, if only one attractor state remains, we want to know what the energy of this state configuration is under Eq. 4. One might expect, given naïve

¹ The application of Hebbian learning does not 'form' a memory *de novo* since the 'learned' attractors were already present in the system's initial dynamics. Nonetheless, the *modifications* that accumulate to these weights constitute a memory in the proper sense that they represent state configurations that the system has visited in the past (and generalisations thereof).

positive feedback principles, that it would have the mean or perhaps modal energy of the attractor states in the original system; but this is not the case.

In order to understand whether the competition between attractors in a self-modelling system enlarges attractors with especially low true energy or not, we need to understand the relationship between attractor basin size and the energy of their attractor states. At first glance it might appear that there is no special reason why the largest attractor should be the 'best' attractor – after all, it is not generally true in optimisation problems that the basin of attraction for a locally optimal solution is proportional to its quality. But in fact, for systems that are additively composed of many low-order interactions, existing theory tells us that this is highly probable.

Specifically, in systems that are built from the superposition of many symmetric pair-wise interactions, the depth (with respect to energy) of an attractor basin is positively related to its width (the size of the basin of attraction). A robust relationship between minima depth and basin size [14] is complicated by the possibility of correlations between minima [15], but minima depth and basin size are in general strongly correlated on average as evidenced by recent numerical work by [16,17,18]. Accordingly, the global minimum is likely to have the biggest basin of attraction. One must not conflate, however, the idea that the global optimum has the largest basin with the idea that it is easy to find the global optimum: In particular, the global optimum may be unique whereas there will generally be many more attractors that lead to inferior solutions. The basins of these sub-optimal attractors will collectively occupy much more of the configuration space than the basin of the global optimum.

Given that low energy attractors have larger basins than high energy attractors, they are therefore visited more frequently and therefore out-compete high energy attractors in a self-modelling system. Thus, (in the limit of low learning rates such that the system can visit a sufficient sample of attractors) we expect that when a dynamical system augments its dynamics with an associative memory of its own energy minimisation behaviour it will produce a dynamics with ultimately only one attractor, and this attractor will correspond to a minimisation of constraints between variables in the original system that is likely to be near globally optimal. This is depicted schematically in (Fig. 1), but basins of attraction in real systems may be much more complex. Although this behaviour will be reliable on average, it should be clear that the system is sensitive to initial random events. Accordingly, when the learning rate is too high, this will cause an arbitrary local optimum to be reinforced.



Fig. 1: Schematic overview of the relationship between the original energy function and the modified energy function. a) The original connections in the system determine the distribution of points (dots) in state space (here represented one-dimensionally) that the system is likely to visit. b) As the connections of the network are slowly modified, forming an associative memory of the distribution in (a), the energy function and hence the distribution of points visited is altered. The new energy function (dotted) is a simplified and generalized model created by the associative memory which may remove small basins and

merge or enlarge others. c) As attractors in the modified energy function compete with one another, the learned model of the system's own dynamics becomes an increasingly simplified caricature of its original behaviour and eventually all local optima are removed. The final distribution of points found at the single attractor of the *new* energy function corresponds to a state configuration that has especially low energy in the *original* energy function (solid curve).

An illustration of a self-modelling dynamical system

Here we illustrate the effects of this self-modelling process in a simple dynamical system. First we show that in general cases self-modelling can cause a system to find low-energy configurations *more reliably* over time, and then we show that it can find low-energy solutions *faster* in some scenarios. The state dynamics and connection dynamics are defined above (Eqs. 1, 2 and 3, and previous section). We examine a system where each initial connection of the system α_{ij} ={-1,1} takes the value -1 or 1 with equal probability. Fig. 2 illustrates how the dynamics of the system are changed by the application of a self-modelling associative memory. The original distribution of attractor energies (Fig. 2.c., relaxations 1-1000) shows that this problem, built with random constraints, has many local optimal solutions when each node acts to minimise constraint violations independently. But Fig. 2 also shows that good solutions to this problem can be found more reliably with this self-modelling approach.



Fig. 2. Self-modelling in a random problem structure. a) Ten example trajectories of system behaviour before learning (N=100, relaxation length 10N), b) attractor states visited (end points of curves in (a)) without learning (relaxations 1-1000) and with learning (relaxations 1001-2000, δ =0.001/10N), c) example trajectories after learning (note that energy minimisation in the modified energy function can result in transient increases in true energy), d) histogram of attractor energies before and after learning. Showing that after learning the system finds one of the lowest-energy configurations reliably, i.e. from any initial condition.

The initial weights of the system represent a weighted-Max-2-SAT problem. Each $\alpha_{ij}s_is_j$ term represents a clause that is satisfied when $\alpha_{ij}s_is_j > 0$ and unsatisfied when $\alpha_{ij}s_is_j < 0$. Thus, each weight

dictates whether the two problem variables it connects should be the same sign ($\alpha_{ij}>1$) or different ($\alpha_{ij}<0$), and the magnitude of α_{ij} denotes the importance of satisfying this constraint. The objective is to find an assignment to the state variables that maximises the number of satisfied clauses, weighted by their importance. Weighted-Max-SAT includes Max-SAT, and the maximisation problem Max-2-SAT is NP-hard [21], and equivalent to other well-known problems. For example, the system can be interpreted as a distributed constraint optimisation problem such as graph-colouring [21]. That is, each node in the network represents an area to be coloured with one of two colours (-1/+1), and the edges in the network represent constraints with other areas determining whether a connected area should be coloured the same ($\alpha_{ij}>1$) or differently ($\alpha_{i}<0$). The objective is to minimise the number of constraint violations.

Finally, to take a slightly more applied example, the system is also equivalent to a simple distributed resource allocation problem such as grid or cloud computing [22]. Suppose each processing node in a network receives jobs of two types at an equal rate from a user. At any one time, however, a processor can only service jobs of one type and jobs of the other type are sent to other connected processors of that node in proportion to a fixed weighting (α_{ij}) for that pair of nodes (e.g. inversely related to distance in order to minimise communication costs). To keep things simple, jobs are only redirected in this manner once (i.e. one 'hop'). Each processor seeks to service the most jobs possible and adopts the servicing mode that services the majority of jobs they receive. The mode they adopt is therefore sensitive to the weighting with each other processor and the modes currently adopted by those processors. (In our implementation, all jobs are the same size, requiring 100% of a processor for one time step, and all processors have the same capabilities – but in principle, such symmetries may be relaxed). In general, this type of system may exhibit many locally optimal configurations where no processor wants to change mode, but the number of jobs arriving at processors that are in the incorrect mode is not globally minimal. This maps to a distributed graph-colouring problem where connected nodes will ideally adopt complementary services/colours ($\alpha_{ii} < 0$) and in each cases where this is not achieved the cost is proportional to the magnitude of α_{ij} number of jobs that are forwarded (but remain unserviced).

Recall versus optimisation

In other instantiations of the system a more surprising result can be observed. We examine the number of trials where a system with learning finds a lower energy configuration than the same system run under the same conditions without learning. When this happens it indicates that the learning process is enabling the network to find better solutions *faster*, even for the first time, and not merely to recall good solutions that have *already* been visited.

Two properties are necessary to observe this. First the problem must admit good solutions that are difficult for local search (i.e. a single relaxation of the system) to find. In this case, we can create a problem that is difficult for local search by introducing modularity. Specifically, we define a nearly-decomposable [20,23,12] modular structure (or subdivided network [24]) (30 modules of 5 variables each, N=150) where

intra-module connections are much stronger than inter-module connections (i.e. $|\alpha_{ij}| = 1$, if $\lfloor \frac{i}{5} \rfloor = \lfloor \frac{j}{5} \rfloor$ ($i \neq j$);

 $|\alpha_{ij}| = 0.01$, otherwise). Strong local connections and weak inter-module connections have the effect of producing many local optima that are distant in Hamming space [23,32], and the balance of these weights can be used to control the relative size of the basin of attraction for the global optima and local optima [12]. Second, local optima that are found by local search must reveal some regularity that associative learning can exploit. This is straightforwardly introduced by biasing the initial connections/constraints. Specifically, here we use $\alpha_{ij} > 0$ with probability 0.8 (rather than 0.5 as in the previous case) creating some consistency in the problem, and increasing the likelihood that local optima contain partial solutions that coincide with one another.

We conducted 100 trials on different random instances of this class of system. Each trial compares one run with learning and one run without; 300 relaxations of each (length of each relaxation is 10*N* state updates). The same 300 random initial conditions and the same random order of state updates are used for the learning and non-learning runs of each trial – thus differences between learning and non-learning trials can only be observed if learning changes the basins of attraction (δ =0.00025/10*N*).

In 90 of the trials the learning system finds a lower energy configuration than the non-learning case (see e.g. Fig. 3), and there are no trials where the non-learning system finds a superior solution to the learning system. Thus the learning system not only finds low-energy configurations with greater reliability

over time but, in a given number of relaxations, finds lower-energy configurations than the non-learning case. Finding these low energy configurations is therefore not simply a matter of recalling good configurations that have already been visited by chance.



Fig. 3. Self-modelling in a modular problem structure. Example trial of non-learning (relaxations 1-300) and learning system (relaxations 300-600) in a nearly-decomposable system. Solid line is the best configuration found without learning. After approximately 250 relaxations with learning, the system converges on a single attractor that is superior to all attractors found without learning over 300 relaxations. (It is also superior to the energy of the all-1s state configuration (broken line) – the energy attainable as if by ignoring the negative connections.)

The significance of this modular problem structure warrants some discussion: Consider a weighted-Max-2-SAT problem where problem variables are clustered into subsets exhibiting strongly-weighted clauses, with weakly-weighted clauses between clusters. This corresponds, for example, to a system where subgroups of processor nodes have high-bandwidth connections between them, such that the majority of job redirection occurs within subgroups, and hence coordination of processors within a subgroup is more important than coordination between subgroups.

The strongest constraints have a large impact on individual node energy and state changes respond reliably to these constraints. But this then makes it difficult/unlikely for the untrained system (i.e. before learning) to respond to/satisfy the remaining dependencies between clusters even though, collectively, their effect is still significant for the system as a whole. However, when locally-optimal configurations have some consistency with one another that associative learning can exploit, initially weak and unreliable intergroup coordination can be gradually strengthened and reinforced.

This means that locally optimal configurations tend to be 'variants on a theme', exhibiting partially reliable sub-patterns. For example, this might be created if there is a biased pattern of incoming job-types across the network of processors, creating a consistency in the pattern of satisfied and unsatisfied constraints that occur in a sample of locally optimal solutions. In a similar manner, the bias we use in this example creates a weak consistency in the resolution of constraints between modules. Although this consistency is not revealed in any one relaxation, it can be recognised and exploited by generalising over many relaxations. (This bias toward states that agree does not make the solution trivial, however; the energy of the state configuration where all states agree is not optimal (Fig.3, broken line)).

Discussion and related work

Given that in a distributed complex adaptive system there is no central mechanism to store and reapply the best result of previous experience, a mechanism that causes a dynamical system to increase the probability or reliability of visiting good configurations that have been visited in the past is significant for many types of engineered complex adaptive systems [25,26,27]. Importantly, the mechanism demonstrated above is extremely simple and completely distributed. Updates to connections depend only on the states of the two

variables they connect, and the method is therefore implementable in distributed complex adaptive systems where centralised optimisation methods (calculating adjustments based on global information) are inapplicable. Its suitability also relies on only weak assumptions about the domain; i.e. systems built from the super-position of many pair-wise constraints. Conceptually, the idea of using neural network mechanisms to enhance the performance of complex adaptive systems suggests a literally connectionist way of thinking about adaptation in complex systems such as, for example, ad hoc communications networks or grid computing, that draws attention away from the intelligence of individual nodes [37].

Optimisation ability

The initial results above illustrate an ability to recall good solutions that have already been visited but the ability to find better solutions *faster* in the latter results is more intriguing. The illustrations with nearlydecomposable systems show that the learning system is finding low-energy configurations on a timescale in which the non-learning system has not yet visited them. It must therefore be enlarging the basin of attraction for these configurations before they are visited for the first time. We conclude that the system is in effect *predicting* the location of superior solutions by generalising the patterns observed in a stochastic sample of inferior local optima [37]. In particular, 'spurious attractors', though generally considered a nuisance in associative memory research [28,13], in fact represent a simple form of generalisation producing new attractor states that are new combinations of features (sub-patterns) observed in the training patterns [29]. This enables the globally optimal attractor to be enlarged even though it has not yet been visited/used as a training sample. However, this will not work well unless common sub-patterns of local optima indicate the position of the global optima or other superior optima. This is not universally true in optimisation problems, and this limitation is related to the observation that Hebbian learning can enhance classification [30] only when the intrinsic patterns of self-similarity in the data support the classification that needs to be learned. But, we suggest that for a problem that can be described in terms of weighted pairwise constraints as per the premise of these investigations (and certainly systems with modular structure, as illustrated) this relationship between local and global optima is a reliable heuristic. The scalability of an associative optimisation process based on related principles is shown to be algorithmically superior to local search in a formal sense [31,32] (see also [33]). However, it is not the aim of the current investigations to suggest this provides a robust or universal optimisation method in general [34] - only to use an optimisation framework to demonstrate how this subtle, and completely distributed, self-modelling process influences the self-organisation and hence future behaviour of a complex adaptive system.

Spontaneous self-organisation in systems of autonomous agents

Hebb's rule is an extremely simple rule: It merely applies positive reinforcement to the current state correlations. In the above experiments we have mandated that Hebb's rule is applied to each connection, but if agents in a complex adaptive system were free to modify connections with other agents in the manner that suited their own self-interest, in which direction would they change them? In fact, it is straightforward to show that Hebb's rule is equivalent to a rule that changes connections in the direction that reduces the strength of constraints that oppose the current state and strengthens connections that support the current state – that is, the direction of changes to connections that reduce the energy of the current state configuration are necessarily Hebbian: i.e.

$$-\sum_{ij}^{N}(\omega_{ij} + \Delta\omega_{ij})s_{i}s_{j} < -\sum_{ij}^{N}\omega_{ij}s_{i}s_{j} \Leftrightarrow sign(\Delta\omega_{ij}) = sign(s_{i}s_{j})$$

This means that if an agent is motivated to choose its *behaviour* to maximise a utility function that is a weighted sum of pair-wise interactions with other agents (analogous to minimisation of the energy function above) then if that agent also has the ability to slowly change *connections* with other agents [35,25] in a manner that maximises that same utility function it will necessarily do so in a manner that is Hebbian. This is an observation that we are developing in related work on complex adaptive systems [37], social networks [38], and also in the context of co-evolving species in an ecosystem where species may evolve the coefficients of a Lotka-Volterra system [39] (see also [40]) or evolve symbiotic relationships [41]. This connects the current work with concepts we refer to as 'social niche construction' [42,43,44,45] – the idea that organisms manipulate their social context in a manner that modifies selection on their own social traits.

In a different domain, recent results [46] have investigated how the evolvability of a population changes over time when it is subjected to a fluctuating but structured environment (we achieve the same conditions using repeated relaxation from random initial conditions in a static but structured environment [47]). Parter et al find that organisms develop a "memory" of their evolutionary history and observe that evolved networks "generalise to future environments, exhibiting high adaptability to novel goals". Wagner et al [48] explain part of the mechanism that might be involved by referring to genetic loci that affect the correlation of phenotypic traits [49] as follows: "natural selection can act on [such loci] to either increase the correlation among traits or decrease it depending on whether the traits are simultaneously under directional selection or not. ...[resulting in] a reinforcement of pleiotropic effects among co-selected traits and suppression of pleiotropic effects that are not selected together"[48]. This clearly describes a Hebbian modification of gene interactions and suggests intriguing parallels we have developed elsewhere [47].

Further work is required to develop these observations and also to examine the effects on systems with asymmetric connections/non-fixed-point attractors and to examine the sensitivity to learning rate (see [12]). Also, here we have applied a very clear separation of timescales between changes to network connections and behaviours on the network (our conditions 2 and 3) – a key which enables us to interpret the interaction of these two dynamics in a simple manner. Without *some* separation of these timescales there is no useful 'signal' in the configuration states that an associative memory can learn, but relaxation of these conditions deserves attention.

In conclusion, the slow application of simple associative learning, given repeated relaxation, provides a fully-distributed mechanism that enhances the ability of a dynamical system to resolve tensions between interdependent components of the system and find globally optimal resolutions of constraints both more reliably and more quickly. We do not claim that this provides a strong optimisation method, and existing theory already tells us a lot about when Hebbian learning can and cannot provide useful generalisation. But the consequences of turning a complex system to the task of modelling its own dynamical attractors, via a mechanism as simple as Hebbian learning, have been previously overlooked and provide a novel frame of reference for thinking about simple modes of self-organisation in complex adaptive systems.

Acknowledgements

Thanks to Jason Noble, Seth Bullock, David Iclanzan, Adam Davies and Ton Coolen.

References

- 1. Strogatz, S.H. (1994) Nonlinear Dynamics and Chaos. (Addison-Wesley, Reading, MA.)
- 2. Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *PNAS USA*, 79 (8) 2554-2558.
- 3. Hopfield, J.J., Tank, D.W. (1985) 'Neural' computation of decisions in optimization problems. *Biol Cybern* 52:141-152.
- 4. Hopfield, J.J. Tank, D.W. (1986) Computing with neural circuits: A model. Science 233: 625-633
- 5. Hebb, D.O. (1949) The organization of behaviour. New York: Wiley.
- 6. Hinton, G.E., Sejnowski, T.J. (1983) Analyzing Cooperative Computation. In *Proceedings of the 5th Annual Congress of the Cognitive Science Society*, Rochester, NY, May 1983.
- 7. Hinton, G.E., Sejnowski, T.J. (1985) Learning in Boltzmann Machines, Cognitiva, 85, Paris, France.
- 8. Ackley, D.H., Hinton, G.E., Sejnowski, T.J. (1985) A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, 9: 147-169.
- 9. Tsirukis, A.G., Reklaitis, G.V., Tenorio, M.F. (1989) Nonlinear Optimization Using Generalized Hopfield Networks. *Neural Computation 1*, 511-521.
- 10. Adaptive networks Springer
- 11. Ito, J., Kaneko, K. (2002) Spontaneous structure formation in a network of chaotic units with variable connection strengths. *Phys. Rev. Lett.*, 88. 028701-1.
- 12. Watson R.A., Buckley, C.L. & Mills, R. (2009) *The Effect of Hebbian Learning on Optimisation in Hopfield Networks*. Technical Report, ECS, University of Southampton.

- Hopfield, J.J., Feinstein, D. & Palmer, R. (1983) 'Unlearning' has a Stabilizing Effect in Collective Memories. *Nature*, 304. 158-159.
- 14. Gardner, E. (1988)The space of interactions in neural network models. J. Phys. A: Math. Gen. 21. 257.
- 15. Coolen, A.C.C. (1991) On the Relation Between Stability Parameters and Sizes of Domains of Attraction in Attractor Neural Networks. *Europhys. Lett.* 16. 73.
- 16. Kryzhanovsky, B., Kryzhanovsky, V. (2008) Binary Optimization: On the Probability of a Local Minimum Detection in Random Search. *ICAISC 2008*: 89-100.
- 17. Kryzhanovsky, B., Magomedov, B.M., Fonarev, A.B. (2006) On the Probability of Finding Local Minima in Optimization Problems. *IJCNN 2006*: 3243-3248.
- 18. Kryzhanovsky, B., Kryzhanovsky, V., Mikaelian, A. L. (2007) Binary optimization: A relation between the depth of a local minimum and the probability of its detection. *ICINCO-ICSO 2007*: 5-10.
- 19. Fontanari, J.F. (1990) Generalization in a Hopfield network. Journal de Physique, 51, 2421-2430.
- 20. Simon, H.A. (1969) The Sciences of the Artificial. (MIT Press, Cambridge, Mass).
- 21. Garey, M. R., Johnson, D. S., & Stockmeyer, L. (1976) Some Simplified NP-Complete Graph Problems, *Theoretical Comp. Sci.*, 1, 237-267.
- 22. Lim, HC. et al. (2009), Automated Control in Cloud Computing: Challenges and Opportunities. 1st Workshop on Automated Control for Datacenters and Clouds, 13-18.
- 23. Watson, R.A. & Pollack, J.B. (2005) Modular Interdependency in Complex Dynamical Systems. *Artificial Life*, 11 (4). pp. 445-457.
- 24. Bar-Yam, Y., (1997) Dynamics of Complex Systems, Perseus Books.
- 25. Jorge M. Pacheco, Arne Traulsen, and Martin A. Nowak. Coevolution of Strategy and Structure in Complex Networks with Dynamical Linking. *Phys. Rev. Lett.* 97, 258103 (2006).
- 26. Heylighen, F.; Gershenson, C.; Staab, S.; Flake, G.W.; Pennock, D.M.; Fain, D.C.; De Roure, D.; Aberer, K.; Wei-Min Shen; Dousse, O.; Thiran, P.; Neurons, viscose fluids, freshwater polyp hydraand self-organizing information systems. *Intelligent Systems*, 18:4. 72-86. IEEE (2005).
- 27. Nettleton, R.W.; Schloemer, G.R.; Self-organizing channel assignment for wireless systems, *Communications Magazine, IEEE* 35:8. 46-51. IEEE (1997).
- 28. Gascuel, J.-D., Moobed, B. Weinfeld, M. (1994) An Internal Mechanism for Detecting Parasite Attractors in a Hopfield Network, *Neural Computation* 6, 902-915.
- 29. Jang, J.-S., Kim, M.W., & Lee, Y. (1992) A Conceptual Interpretation of Spurious Memories in the Hopfield-type Neural Network. *Neural Networks, 1992. IJCNN., International Joint Conference on.* 1:21-26.
- 30. O'Reilly, R.C., Munakata, Y. (2000) Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain, (Cambridge, MA: MIT Press).
- 31. Mills, R., & Watson, R.A. (2009) Symbiosis Enables the Evolution of Rare Complexes in Structured Environments. *Proceedings of European Conference on Artificial Life 2009* (In press).
- 32. Mills, R. (2010) *How Micro-Evolution Can Guide Macro-Evolution: Multi-Scale Search via Evolved Modular Variation.* PhD thesis, ECS, University of Southampton.
- 33. Iclanzan, D., Dumitrescu, D. (2007) Overcoming hierarchical difficulty by hillclimbing the building block structure. *GECCO*, pp. 1256–1263.
- 34. Wolpert, D.H., Macready, W.G. (1997), No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation* 1, 67.
- 35. Newman, M. E. J., Barabasi, A. L. & Watts, D. J. (Eds.) (2006). *The Structure and Dynamics of Networks*. Princeton University Press.
- 36. Watson, R. A., Buckley, C. L. and Mills, R. (2009) *Global Adaptation in Networks of Selfish Components: Emergent Associative Memory at the System Scale.* Technical Report, ECS, University of Southampton.
- 37. Watson, R. A., Mills, R. & Buckley, C. L. (submitted) Global Adaptation in Networks of Selfish Components: Emergent Associative Memory at the System Scale. *Artificial Life*.

- 38. Davies, A.P., Watson, R.A., Mills, R., Buckley, C. L., Noble, J. (2010) If you can't be with the one you love, love the one you're with: How individual habituation of agent interactions improves global utility. *ALIFE XII* (to appear).
- 39. Lewis, M. (2009) An Investigation Into The Evolution Of Relationships Between Species In An Ecosystem. MSc Dissertation, ECS, University of Southampton.
- 40. Poderoso, F.C., & Fontanari, J.F. (2007) Model ecosystem with variable interspecies interactions, J. Phys. A: Math. Theor. 40 8723-8738
- 41. Watson, R.A., Palmius, N., Mills, R., Powers, S.T., & Penn, A.S. (2009a) Can Selfish Symbioses Effect Higher-level Selection? *European Conference on Artificial Life 2009*. (in press).
- 42. Odling-Smee, F.J., Laland, K.N., Feldman, M.W. (2003) *Niche Construction. The Neglected Process in Evolution.* Monographs in Population Biology. 37. (Princeton University Press).
- 43. Penn, A. S. (2006). *Ecosystem Selection: Simulation, Experiment and Theory*. PhD thesis, University of Sussex.
- 44. Powers, S.T., Mills, R. Penn, A.S., Watson, R.A. (2009) Social niche construction provides an adaptive explanation for new levels of individuality (ABSTRACT), *Proceedings of Workshop on Levels of Selection and Individuality in Evolution, European Conference on Artificial Life 2009.*
- 45. Powers, S.T., Penn, A.S. & Watson, R.A. (2007) Individual Selection for Cooperative Group Formation. *Proceedings of European Conference on Artificial Life 2007*. pp. 585-594.
- 46. Parter, M., Kashtan, N., Alon, U. (2008) Facilitated Variation: How Evolution Learns from Past Environments to Generalize to New Environments. *PLoS Comput Biol* 4(11): e1000206.
- 47. Watson, R. A., Buckley, C. L., Mills, R., Davies, A.P. (2010) Associative memory in gene regulation networks, *ALIFE XII* (to appear).
- 48. Wagner, G.P., Pavlicev, M., & Cheverud, J.M. (2007) The road to modularity. *Nature Reviews Genetics* 8: 921-931.
- 49. Pavlicev, M, Kenney-Hunt, J.P., Norgard, E.A., Roseman, C.C., Wolf, J.B. & Cheverud, J.M. (2008) Genetic variation in pleiotropy: differential epistasis as a source of variation in the allometric relationship between long bone lengths and body weight. *Evolution*. 2008 Jan;62(1):199-213.