Transformations in the Scale of Behaviour and the Global Optimisation of Constraints in Adaptive Networks

Short Title: Transformations and Optimisation in Adaptive Networks

Richard A. Watson[1], Rob Mills[1] and C. L. Buckley[2]

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

Fax: +44 (0)2380 599179, Tel: +44 (0)23 8059 2690

[2]Informatics, Sussex University, U.K.

Keywords: *Hopfield networks, associative learning, dynamical systems, adaptive networks, constraint optimisation, modularity, nearly-decomposable systems, coarse-graining, self-organisation, canalisation.*

Transformations in the Scale of Behaviour and the Global Optimisation of Constraints in Adaptive Networks

The natural energy minimisation behaviour of a dynamical system can be interpreted as a simple optimisation process, finding a locally optimal resolution of problem constraints. In human problem solving, high-dimensional problems are often made much easier by inferring a low-dimensional model of the system in which search is more effective. But this is an approach that seems to require top-down domain knowledge; not one amenable to the spontaneous energy minimisation behaviour of a natural dynamical system. However, in this paper we investigate the ability of distributed dynamical systems to improve their constraint resolution ability over time by self-organisation. We use a 'self-modelling' Hopfield network with a novel type of associative connection to illustrate how slowly changing relationships between system components can result in a transformation into a new system which is a low-dimensional caricature of the original system. The energy minimisation behaviour of this new system is significantly more effective at globally resolving the original system constraints. This model uses only very simple, and fully-distributed positive feedback mechanisms that are relevant to other 'active linking' and adaptive networks. We discuss how this neural network model helps us to understand transformations and emergent collective behaviour in various non-neural adaptive networks such as social, genetic and ecological networks.

1. The computational abilities of dynamical systems

Physical dynamical systems with a large number of simple equivalent components have been shown to exhibit “emergent collective computational abilities” (Hopfield, 1982) such as implementing content-addressable memory or solving constraint satisfaction problems (Hopfield & Tank, 1986; Hopfield & Tank, 1985). In the latter, Hopfield and Tank equate the energy minimisation behaviour (Strogatz, 1994) of a dynamical system with an optimisation process – i.e., the system moves to configurations that better-resolve the conflicting constraints between system variables. But actually, energy minimisation in a simple dynamical system is equivalent to the simplest possible optimisation algorithm, namely gradient descent (or incremental improvement), which in anything but the simplest of problems tends to find only locally optimal solutions. In human design-engineering and optimisation, high-dimensional problems are often made much easier by inferring a low-dimensional model of the system (e.g., a high-level representation that exploits modularity/problem decomposition), such that ‘local’ search in this new or rescaled space is better able to find a globally optimal resolution of constraints. This is an approach that seems to require top-down domain knowledge and design intelligence, and does not appear to be amenable to the spontaneous energy minimisation behaviour of a natural dynamical system. But, can other types of dynamical systems, specifically self-organising systems, perform more sophisticated forms of optimisation? And conversely, can an optimisation framework help us to better understand the behaviour of natural self-organising systems?

Our questions are motivated by consideration of self-organising multi-agent systems, such as species in an ecosystem or agents in a socio-economic network, and their potential to exhibit emergent collective behaviours. In particular, we are interested in the possibility that such systems can spontaneously transform into a new system, operating at a higher level of organisation (Simon, 1969; Maynard Smith & Szathmary, 1995), and that such a dynamical transformation may facilitate (or may even be equivalent to) a transition in the ability to resolve constraints between the system components. Our earlier work in this area, using abstract optimisation algorithms loosely inspired by symbiogenesis (Mills, 2010; Mills & Watson, 2011) suggested that simple associative rules could, in the right context, be sufficient to identify and exploit higher-level problem structure; and individual-based evolutionary models of the formation of new units of selection (Watson, et al., 2009b; Watson, Palmius, Jackson, Mills, Powers, Buckley & Penn, in prep.) showed that such associations could be favoured by natural selection and thereby occur spontaneously under suitable conditions.

To formalise the formation of such relationships we can characterise multi-agent ecosystems or socio-economic network systems as ‘adaptive networks’ (Newman, 2003; Newman, Barabasi & Watts, 2006; Gross & Sayama, 2009); exhibiting the property that the structure of connections between agents affects changes to the agent behaviours and, vice versa, that the agent behaviours affect changes to the structure of connections between agents. The Hopfield network (Hopfield, 1982) easily accommodates such state/topology coadaptation and, at a very abstract level, provides a suitable system with which to explore how self-organisation in adaptive networks alters their ability to resolve conflicting constraints between system components. Working with this analogy implied that the associative connections exploited by evolution and by our abstract algorithms might be directly analogous to associative learning (Hopfield, 1982; Hinton & Sejnowski, 1985) in the Hopfield network. Investigation of this potential homology has been fruitful (Watson et al., 2009a; Watson, et al., 2010a; Watson, et al., 2010c; Davies, Watson, Mills, Buckley, Noble, 2010), but we found that the usual interpretation of learned associations in the Hopfield network did not provide the same ability to resolve constraints as the associations that were exploited in the formation of new evolutionary units (or in our abstract algorithm). Specifically, the usual interpretation of learned associations in the Hopfield network provided an interesting improvement to what is essentially a gradient descent or incremental improvement method – but it did not really provide a scaling-up in the dynamical behaviour of the system. However, in the current paper we find that a different interpretation of learned associations in the Hopfield network does provide a true re-scaling of the dynamical behaviour and constraint optimisation ability. The current paper now provides a formal description of this difference (contrasting the usual type of associations, which we refer to as *selective associations,* with the new interpretation, which we call *generative associations*) and relates the findings more closely with self-organised dynamical systems and formal optimisation methods.

For example, unlike simple gradient descent methods, state-of-the-art computational optimisation techniques (e.g., Pelikan, 2002; Pelikan, Goldberg & Lobo, 2002) have the option to use advanced model-building mechanisms that can identify and exploit high-level structure in a problem domain automatically (Mills, 2010; Mills & Watson, 2011). In general terms, such methods seek to identify a low-dimensional search space within a high-dimensional problem domain by 'memoising' (Michie, 1968) partial solutions (Angeline & Pollack, 1992; Koza, 1994), or 'chunking' (Rosenbloom, & Newell, 1986) sequences of actions, to re-use them as macro-operations that rescale the search process. The resultant lower-dimensional space can facilitate global constraint optimisation far superior to local search methods (Mills & Watson, 2011). But it has not previously been shown that it is possible for simple distributed mechanisms of self-organisation, gradually changing the connections of an adaptive network, to cause it to effectively rescale its dynamics and hence move from local to global energy minimisation in a similar manner.

Our aim is therefore to provide a concrete model of how an adaptive network can be transformed from one scale of dynamics to another via simple fully-distributed self-organising mechanisms. More narrowly, our objective is to devise a simple distributed mechanism that modifies the connections of a *self-modelling* (Watson, et al., 2010a) Hopfield network such that its dynamics change from local to global energy minimisation. In particular, we aim to exploit implicit modular, or nearly-decomposable (Simon, 1969; Watson & Pollack, 2005), structure in the constraints of the system such that, whereas the original dynamics of the system search combinations of the original state variables, the transformed dynamics search combinations of higher-level emergent variables.

Working within an adaptive networks framework, we restrict attention to a self-organising mechanism that changes the connections between components – i.e. causes changes to weighted connections in the network. Specifically, we assume connections grow in strength in a manner that reflects observed correlations between state variables as per our prior work. This is a simple and fully-distributed, positive feedback mechanism familiar in neural network research as associative or Hebbian learning (Hinton & Sejnowski, 1983; Hinton & Sejnowski, 1985; Ackley, Hinton & Sejnowski, 1985; Hebb, 1949). We refer to this as a self-organising mechanism rather than a learning mechanism since the patterns that are 'learned' are created by the system's own dynamics (rather than an external training regime). Although in this paper we simply mandate that such changes to connections occur in our network, our recent work shows that Hebbian changes to connections are natural in many types of complex adaptive systems, and not just neural systems (Watson, et al., 2010c). Accordingly, we aim to demonstrate a mechanism that rescales the optimisation ability of the Hopfield network and thereby to provide a formal model that helps us understand transformations in the scale of 'problem solving' behaviour in other types of adaptive networks.

A central focus of the current paper is the contrast between two types of associative connection. Specifically, characterising the energy minimisation behaviour of the system as a 'generate and test' (Winston, 1992) optimisation process, we show that learned associations can straightforwardly modify either the generate function or the test (or 'selection') function of this process. We find that associations that alter the generate function are much more effective at enabling global minimisation of energy. Such *generative associations* (in contrast to the conventional *selective associations*) have the effect of producing probabilistic, correlated, simultaneous state changes in multiple variables and correspond to emergent 'macro variables' that encapsulate sub-solutions. We find that it is possible for a self-organised dynamical system with generative associations, using only simple distributed mechanisms, to transform into a new system operating at a higher level of organisation, and we show that this transformed system minimises total system energy more reliably and more completely than the original system. In effect this provides a novel distributed optimisation algorithm. But more important for our current aims, this transformation is enacted through very simple, and fully-distributed, positive feedback mechanisms that are relevant to other 'active linking' systems and adaptive networks (Pacheco, Traulsen, & Nowak 2006; Santos, Pacheco & Lenaerts, 2006; Van Segbroeck, Santos, Pacheco, Lenaerts, 2010; Gross & Sayama, 2009). We therefore suggest that this neural network model helps us to understand transformations and emergent collective behaviour in various non-neural adaptive networks such as social, genetic and ecological networks – for example, the emergence of coalitions in social systems and of higher-level adaptive units in the major transitions in evolution.

2. Methods

The Hopfield network (Hopfield, 1982) is an abstract model of neural architecture and a well-understood example of a simple dynamical system that has provided a vehicle for studying adaptive behaviour (e.g. Goetz & Walters, 1997; Paine & Tani, 2005) and the computational properties of physical systems across many disciplines. Hopfield and Tank's work (Hopfield & Tank, 1985; Hopfield & Tank, 1986) makes an explicit bridge between the language of dynamical systems energy-minimisation and the language of optimisation or search which is very useful for formalising the type of dynamical transformation we wish to study. Studying multi-level structure and multi-level optimisation (Mills, 2010; Mills & Watson, 2011) in the context of a neural network model has the advantage that the ability of distributed neural mechanisms to identify high-level features in a set of training samples (McClelland, Rumelhart & the PDP Research Group, 1986; O'Reilly & Munakata 2000), and the potential of recurrent neural networks to find solutions to constraint optimisation problems (Hopfield & Tank, 1985; Tsirukis, Reklaitis & Tenorio, 1989), are already well-understood. Our recent work brings these two well-known uses together (Watson et al., 2009a) in what we call a 'self-modelling' system (Watson, et al., 2010a) – a system that (implicitly) models its own behaviour.

In overview, the model operates as follows. An energy function encodes an optimisation problem which governs the initial energy minimisation dynamics of the system (Hopfield & Tank, 1985). We implement an energy function that represents a constraint optimisation problem by using a set of weights corresponding to constraints imposed by an external environment. The system is then repeatedly relaxed from many different random initial conditions, causing it to visit many different dynamical attractors in this energy function. If the transients of each relaxation are short compared to the duration of each relaxation this causes the system to spend most of its time at local minima in the energy function. Over the course of many relaxations, Hebbian learning is used to modify a set of internal weights. This has the effect of enlarging the dynamical basin of attraction for the system's current state configuration, and therefore tends to make the system more likely to visit state configurations that have already been visited – i.e. simple positive feedback. But this will also enlarge the basin of attraction for other state configurations, even if they have not yet been visited, in proportion to their similarity to state configurations that have been visited. This associative generalisation has the effect of increasing the likelihood that the system visits low energy attractors, including the globally minimal energy attractor, and increases the speed with which it is first found.

2.1 Discrete Hopfield network with restarts, *rHN*

The state of a network consisting of N discrete states $s_i = \pm 1$ where i=1,2,…,N can be written as S=(s_1,…,s_N). The dynamics of the recurrent network used by Hopfield can be described by updates to individual states:

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

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 non-negative arguments respectively). The Hopfield network is run by repeatedly choosing a unit, i, uniformly at random and setting its state according to the above expression.

Hopfield showed (Hopfield, 1982) that if the connection matrix is symmetric, $\omega_{ij}=\omega_{ji}$, and under suitable constraint on the self-weights (here $\omega_{ii}=1$), all trajectories described by Eq. 1 converge on point attractors which are minima of an energy or potential function given by:

$$E_S = H(S,\Omega) \equiv -\sum_{ij}^{N}\omega_{ij}s_i s_j \,. \qquad (2)$$

Consequently one can describe the asymptotic behaviour of such a network in terms of a process that minimises this function. Accordingly, if the weights of a network are appropriately defined they can be used to define an energy function that represents an optimisation problem, and running the network finds locally optimal solutions to the problem (Hopfield & Tank, 1985). The Boltzmann machine (Hinton & Sejnowski, 1985) is a discrete stochastic counterpart of the Hopfield network where a single state change is accepted probabilistically according to the change in energy it produces. We can describe such a dynamical process more generally via a probability of accepting a stochastic change to the system state:

$$\mathrm{P}[S_{(t+1)} \mapsfrom f(S_{(t)})]=\sigma(T,\Delta E)\,, \qquad (3)$$

where the variation operator f is, in the Boltzmann or discrete Hopfield cases, a 'bit flip' operator defined as $f(S) = (s_1,\ldots,\text{-}s_X,\ldots,s_N)$ where X is a uniform random variable on $[1,...,N]$, $\Delta E=E_{S'}\text{-}E_S$ is the change in energy implied by the new state $S'=f(S)$, and $\sigma(T, x) = \frac{1}{1+\exp(-\frac{1}{T}x)}$ is a sigmoid function of x where T is the *temperature* of the system; a parameter that indirectly controls the probability of an increase in energy. When the temperature is reduced gradually this describes a simulated annealing process (Kirkpatrick, Gelatt & Vecchi, 1983). But for a deterministic system, as T=0, we can simply write:

$$\mathrm{P}[S_{(t+1)} \mapsfrom f(S_{(t)})]= \lambda(\Delta E)\,, \qquad (4)$$

where λ is a threshold function taking values 0 and 1 for negative and non-negative arguments, respectively. Thus, the discrete Hopfield network and the Boltzmann machine with T=0 are both equivalent to a bit-flip gradient descent algorithm. Although the continuous state version of the Hopfield network and the non-zero temperature Boltzman machine are, in many circumstances, better optimisers than the bit-flip gradient descent algorithm, the discrete deterministic process is sufficient to illustrate the effect we wish to demonstrate in this paper. This type of 'generate and test' (or generate and select) energy descent process, i.e. updating the network by accepting a stochastic change in state if and only if it decreases system energy, is conventional in stochastic local search or mutation-based evolutionary models (see Section 4.5), whereas the Hopfield equations of motion, which modify system states deterministically in the direction that minimises energy, are conventional in neural networks and physical dynamical system models. However, these frameworks are interchangeable for the basic Hopfield network, and the generate-and-test framework naturally affords the incorporation of generative associations described below.

Since these deterministic dynamics will quickly find a local optimum in the energy function, it is useful to consider a 'random restart' version of the Hopfield network, *rHN*, where the state of the system takes a random state configuration, $R=\{-1|1\}^N$, every τ time steps. More generally, we may suppose that the system is occasionally radically perturbed so as to cause it to move into the basin of a different attractor (this corresponds to many attempts at solving the problem, finding different locally-optimal solutions). We refer to each inter-reset duration of τ time steps as a *relaxation*, and assess the minimum energy and average energy of states visited by the system over many relaxations. Note that the energy of configurations visited within a relaxation will be minimal at the end of that period, so it is often useful to monitor the energy of the system at that time. If each relaxation is long enough for the system to reach a local optimum, this will thus monitor the local optima visited by the system. We refer to this default type of restart Hopfield network (with no learning) as *rHN-0*.

Hopfield and Tank (Hopfield & Tank, 1985) showed that the weights of a Hopfield network can be used to encode the constraints of a combinatorial optimisation problem such that minima in the energy function correspond to local minima in the optimisation problem. This provides a fully-distributed problem solving or optimisation mechanism but the solutions that this basic mechanism produces are only locally optimal and, in general, may be far from globally optimal.

2.2 Hebbian learning and self-modelling in the Hopfield Network

In the following experiments we modify the dynamics of the restart Hopfield network described above using self-modelling (Watson, et al., 2010a). This simply means that Hebbian learning (see below) is applied to the connections of the restart Hopfield network whilst it is running. Given that the duration of relaxations is long enough for the system to spend most of its time at local optima then in effect these locally-optimal configurations become 'training samples' for an associative memory. In this manner the system implicitly uses observations of its own optimisation dynamics to identify problem structure and modify its dynamics, hence *self-modelling*. Clearly, if the system spent most of its time in arbitrary state configurations then this self-modelling could not yield any useful information. But given that the system spends most of its time at configurations that are low in energy, the model that is induced is not arbitrary even though it is modelling its own state dynamics. Importantly, a distribution of locally optimal configurations sampled by *rHN-0* has the potential to exhibit correlations that can reveal problem structure. Thus the self-modelling protocol causes a network to implicitly self-generate a set of training samples from an energy function with the possibility of revealing the system's constraint structure.

The original connection matrix, Ω, representing the problem, remains unaltered by this process but Hebbian learning is applied to all m_{ij}, $i \neq j$, in an internal connection matrix, M, as below. Alternatively, two sets of weights are not required if $H(S,\Omega)$ is replaced by a potential function (or 'fitness function') of arbitrary construction representing environmentally imposed problem constraints that produce correlations in neural activations and M represents the (single) set of internal weights representing the learned response to those activations. The initial value of all m_{ij} is 0.

$$m_{ij}(t+1) = \gamma[m_{ij}(t) + \frac{\delta}{\tau} s_i(t) s_j(t)],$$

where (δ/τ) is a learning rate and γ is a linear threshold function capping all learned weights at a magnitude of 1 (i.e. $\gamma(x)=max(-1,min(1,x))$). If we assume that $\tau >> t^*$, where t^* is the time for the system to reach a local optimum then the cumulative effect of learning over a relaxation will be approximately equal to a single application of learning at the end of each relaxation:

$$m_{ij}(\tau+1) = \gamma[m_{ij}(\tau) + \delta s_i(\tau) s_j(\tau)]. \qquad (5)$$

This end-of-relaxation learning is computationally less expensive to implement so we use this in our experiments, but a continuous learning process may be more natural for models of some systems. We can see that *M* will become an associative memory model of (the local minima of) the original energy function (implicitly defined by *Ω*).

2.3 Modifying dynamics with selective associations, *rHN-S*

This associative memory can then modify the dynamics of the system in two different ways depending on how these learned associations are interpreted. The first of these, which here we term *selective associations*, is to modify the energy function (Watson, et al., 2010a; Tsutsumi, 2003) such that stochastic changes to individual states are accepted if they decrease the value of a new energy function *E′* (Watson, et al., 2010a), i.e.

$$P[S_{(t+1)} \leftarrow f(S_{(t)})] = \lambda(\Delta E'). \qquad (6)$$

where *E′=H(S,Ω)+H(S,M)=H(S, Ω+M).* That is, the new energy function is simply the original energy function (Eq. 2) plus the additional energy derived from the learned weights. This has the effect of changing the relative importance of different constraints in a manner that 'warps' the energy surface (Tsutsumi, 2003). Since this is equivalent to modifying the original weights (that are supposed to represent the problem we are trying to solve) this is a somewhat strange thing to do to a Hopfield network that is being used for optimisation. However, the effect of this is interesting for three reasons:

a) Because the associative memory that is learned is trained at locally optimal solutions to the problem, and the effect of Hebbian learning is to increase the basin of attraction for the training samples, the effect of this learning is to increase the basin of attraction for the locally optimal configurations that are visited.

b) Because low-energy configurations tend to have larger basins of attraction (Gardner, 1988; Coolen, 1991; Kryzhanovsky & Kryzhanovsky, 2008), the lower energy attractors tend to be visited more often, and are therefore enlarged more often, than higher-energy locally optimal solutions.

c) Because sub-patterns that are common to many local optima tend to also be common to the global optimum, the generalisation of patterns provided by associative memory (Fontanari, 1990; Jang, Kim & Lee, 1992) has the potential to enlarge the basin of attraction for the globally optimal configuration even before it is visited for the first time (Watson, et al., 2010a; Watson, et al., 2010c). Since there is no optimisation benefit in recalling a solution only after it has been discovered by chance, this generalisation ability is crucial for a *memory* process to become an *optimisation* process.

Note that although the trajectory of the system through state space is modified by the new energy function, *E′*, as external observers of how the system has changed we will assess the quality of the solutions with respect to the original system constraints, i.e. *E*.

This model, *rHN-S*, is a distributed means to increase the optimisation ability of the Hopfield network (Watson, et al., 2010a). The learned associations in *rHN-S* modify the energy function, or the test (i.e. selection) part of a generate-and-test process – hence *selective associations*. We have investigated self-organisation using selective associations in recent work (Watson et al., 2009a; Watson, et al., 2010c; Watson, et al., 2010b; Mills, 2010; Watson, et al., 2010a; Davies, Watson, Mills, Buckley, Noble, 2010). But we find that this does not provide a true scaling-up in the dynamical behaviour of the system. In this paper we introduce a different type of association which we call *generative associations*.

2.4 Modifying dynamics with generative associations, *rHN-G*

Generative associations modify the *generate* part of the generate-and-test process, rather than the test part. Specifically, learned connections (probabilistically) produce correlated state changes in multiple state variables simultaneously (and the test or selection part of the process is unchanged) (Fig.1).

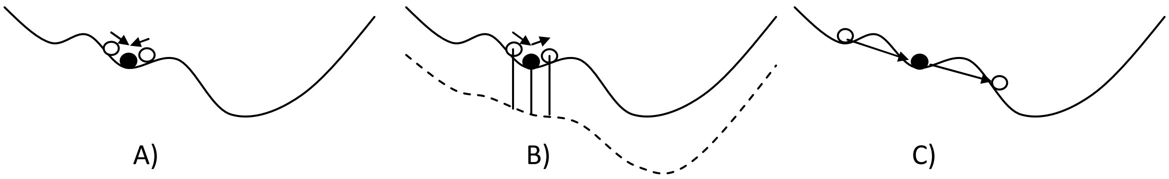

Fig. 1 Different complementary ways to modify the trajectory of a dynamical system with respect to an energy surface. A) Original dynamics: The current system configuration (black dot) is depicted in a local minimum on an energy surface (solid curve), single-state variants on either side (white dots) are higher in energy and hence the local trajectories of the system (arrows) return it to the original location. B) Modified selection: the energy values of the neighbouring positions when measured in the *modified* energy surface (dotted curve) enable the system to move to the right and thereby escape this local optimum in the original energy surface. C) Modified generation: Here the system escapes the local optimum because the variants that are generated are not single-state changes but specific multi-state changes, one of which is lower in energy.

In Fig.1.B the selection criteria of variants is altered but the generation of variants is unchanged; whereas, in C the generation of variants is altered but the selection criteria of variants is unchanged. Another analogy that might be useful is that whereas *rHN-S* modifies the fitness values of a landscape, *rHN-G* can be understood as a process that modifies the neighbourhood structure of a fitness landscape (Mills & Watson, 2011).

Generative associations are implemented with a modified variation operator, *g*, (replacing *f*). This operator interprets the learned weights, *M* (containing explicit knowledge of the problem structure), as defining probabilities of correlated state changes. To facilitate a unified definition of *f* and *g*, we can redefine *f* in terms of *g* and a correlation matrix *C*. *C*=*I*, were *I* is the identity matrix, then represents the state changes produced by *f* in the sense that single-variable state changes are allowed but no other variables are correlated with them – i.e. under *f*, each variable is (with respect to *generating* state changes) 'correlated' only with itself (i.e. it has only self-connections). In contrast, under more general usage of *g*, single state changes are augmented by correlations defined in *M* by using *C*=*I*+*M*. Thus initially, when all m_{ij} = 0, *g*(*S*)=*f*(*S*), i.e. both implement single-variable state changes at random. But later in the learning process, if some node *i* has strong learned connections to another node *j*, then when a new state is assigned to node *i*, an appropriate state is also assigned to node *j*, with probability proportional to correlations observed in the past. Formally,

$$g(S) = (s_1', \ldots, s_X', \ldots, s_N'), \qquad (7)$$

where *X* is a uniform random variable on [1,...,*N*], and s' takes the new value -$\theta(c_{Xj}s_X)$ if $|c_{Xj}|>r$, where *r* is a random threshold (drawn once for each application of *g*) uniformly in the range (*a*,1], and *a* is the mean magnitude of values in *M*. Thus, because *C*=*M*+*I*, the state of s_X is necessarily flipped (as was the case in *f*), but additionally all other states, $s_{j\neq X}$, will either be changed to agree with this new state when $m_{Xj} > 0$, or disagree with the new state when $m_{Xj} < 0$, with a probability that increases with the magnitude of m_{Xj}. The adjustment, a, ensures that only learned correlations with above-average strength will cause additional state changes.

Thus, quite straightforwardly, this new operator recreates sub-patterns in the state configuration according to the frequency with which they have been observed in the past and does so using only the distributed network of pairwise associations generated by normal Hebbian learning – but the mechanistic effect of these associations is interpreted differently. Of course, if the learning rate is too high this will have the effect of moving the system directly to a configuration (e.g. a local optimum) that has been visited before. But if the learning rate is appropriate, this mechanism offers the possibility of recalling commonly occurring *partial* solutions (since a given partial solution may occur in many different complete solutions) and importantly, creating new combinations of these sub-solutions to explore the search space more efficiently. Thus correlated state changes that canalise commonly occurring partial solutions effect a rescaling of the dynamical behaviour and effect a search process operating at a higher level of organisation. But note that if M has arbitrary entries, then the effect of g is merely to produce random 'macro-mutations' (Forrest & Mitchell, 1993) which are not sufficient to solve problems of the form we study here (Watson & Jansen, 2007).

As in *rHN-0*, a (single-variable or multi-variable) state change produced by g is tested against the original energy function, E, to see how it affects energy (a calculation based on the change in energy for each variable that changes state, rather than total energy, may be sufficient). If it reduces energy, the new state configuration is retained, else it is rejected, i.e.:

$$\mathrm{P}[S_{(t+1)} \leftarrow\!\dashv\; g(S_{(t)})] = \lambda(\Delta E). \qquad (8)$$

Whereas we might think of the effect of selective associations as learning to respond to correlations in a stimulus, we can interpret the effect of generative associations as learning to generate correlations in an output or behaviour. Generative associations, like selective associations, are consistent with Hebb's fundamental conjecture (Hebb, 1949) that learning happens by modifying connections between neurons, and also consistent with Hebb's rule that builds connections in proportion to past correlations. They also have the same effect as Hebbian learning in the sense that they make future system states more likely to recreate past correlations. But in conventional neural models (i.e. Eq. 1) attention is placed on single state updates, decided on the basis of activation potentials that are fed into one neuron from many pre-synaptic neurons. This can only change the activation of one neuron at a time. In contrast, in this generative model, the focus is on the effect of an action potential that is fed out from one neuron into many post-synaptic neurons, producing correlated state changes in many neurons simultaneously (e.g. see 'synchronous firing chains' (Abeles, 1982), and 'in star' versus 'out star' configurations (Grossberg, 1978)). But clearly, our emphasis in this paper is not on the physical implementation of the mechanisms involved but in the dynamical and computational properties of such associative connections.

More generally, we are interested in the dynamical and computational properties of a dynamical system undergoing self-organisation via the formation of this type of association. For example, coadaptation between species that modifies the effective evolutionary unit, such as group formation and symbiogenesis (Mills & Watson, 2009; Powers, Penn & Watson, 2007; Watson & Pollack, 2003; Watson, 2006), is analogous to generative associations whereas coadaptation that modifies the 'interaction coefficients' between species, i.e. ordinary coevolution, merely affects the selective environment of the original evolutionary units (Powers, 2010; Lewis, 2009) and is analogous to selective associations. We will briefly discuss this and other non-neural systems later. But our narrow aim in this paper is to show that although the effect of self-organisation with generative associations corresponds to that of a model-building optimisation algorithm, the 'information' required for these mechanisms can be discovered by a simple distributed process and can be 'represented' implicitly in the distributed network of simple associative connections.

2.5 Summary of models

Our underlying model is a Hopfield network with discrete state dynamics and multiple relaxations restarted from random initial conditions after each τ time steps. Modifications to the default model using either selective or generative associations can be understood as a logically complementary pair of processes within the generate-and-test search framework. Specifically, the default generate-and-test procedure, *rHN-0*, generates a candidate state configuration from some probability distribution, e.g., the set of single-bit-flips, then tests for an improvement (decrease in energy). *rHN-S* generates the same distribution of candidates as *rHN-0* but uses a modified test – accepting decreases in the modified energy function. In contrast, *rHN-G* uses a modified generate procedure (macro-variations) but the original test function.

We can define all three algorithms using a general model denoted *rHN*(A,B) where A is a matrix defining the correlations used to generate variants (via the operator g), and B is a matrix defining the energy function used to assess the variant state configurations thus created (via the Hopfield equation, $H(S,B)$, eq. 2). Specifically, *rHN-0* = *rHN*(I, Ω) where I is the identity matrix (producing single state-changes only) and Ω is the original energy function. Then *rHN-S* = *rHN*(I, $\Omega+M$) such that the energy function is augmented by M, and conversely, *rHN-G* = *rHN*($I+M$, Ω) such that the correlations in the variations produced are governed by the learned matrix M. This also suggests a general framework using M in both the generator and the test function, *rHN*($I+\alpha M$, $\Omega+(1-\alpha)M$), creating an algorithm that can be parameterised from one to the other (i.e., α=0 defines *rHN-S* and α=1 defines *rHN-G*), but we have not investigated other values of α in the current paper.

Note that when δ=0, both *rHN-S* and *rHN-G* are identical to the original non-adaptive Hopfield network, *rHN-0*. Similarly, the initial behaviour of *rHN-S* and *rHN-G* before significant learning has taken place will also be qualitatively identical to *rHN-0*.

In some respects, *rHN-S* and *rHN-G* are very different mechanisms: the former uses standard bit-flip exploration but distorts the energy function and the latter enables macro-exploration over the original energy function. However, they both use a matrix, M, generated by the same fully-distributed Hebbian learning mechanism, to exploit correlations in the state variables observed at locally optimal configurations (however, note that since M is learned from a self-referential examination of the system's own dynamics, the values of M learned by *rHN-S* will diverge over time from those learned by *rHN-G* even though the distribution of locally optimal configurations is *initially* identical). Also, both algorithms use this data to make patterns of activation that have been observed in the past more likely to occur again in future dynamics (by either rewarding those sub-patterns or generating those sub-patterns). Both therefore have the potential to exploit problem structure and in particular the ability of associative memory to generalise over a set of patterns (vital for finding *novel* patterns of low energy). However, clearly *rHN-S*, being wedded to single-variable state changes, does not have the possibility of rescaling the search space into a higher-level abstraction. Accordingly, we shall see that the way in which *rHN-G* uses learned correlations has capabilities that are not provided by *rHN-S*. *rHN-S* is nonetheless a more obvious way of adding Hebbian learning to a Hopfield network than *rHN-G* and the comparison between the abilities of *rHN-S* and *rHN-G* is useful to verify that this more simplistic approach to 'recalling past solutions' in a dynamical system with selective associations is not sufficient to solve problems that can be solved by rescaling the search space in a dynamical system with generative associations.

2.6 An Idealised Nearly-Decomposable Constraint Optimisation Problem

In an optimisation framework, our aim is to provide a conceptual illustration of a generic capability to exploit problem structure using distributed mechanisms. This relies, of course, on the existence of structure in the problem domain/original system dynamics that can be exploited. Accordingly, rather than measure the impact of this mechanism in a specific application where the presence or absence of high-level structure is unknown, we investigate it using abstract cases where we can control the structure present in the system in a systematic manner.

Our test problems are represented using a multi-modal energy function defined by different types of weight matrices, Ω, as below and the Hopfield equation (Eq. 2) – this ensures that the problem class is a superposition of pair-wise constraints (Watson, et al., 2010a; Watson, et al., 2010c). We examine two different classes of system: random constraints (*RC*); and an explicitly modular constraint problem (*MC*) that facilitates supporting analysis and illustration. The modular problems are defined as a macro-scale version of the random problems as discussed below.

Both these problems are equivalent to instances of MAX-SAT, graph-colouring and simple resource-allocation optimisation problems (Watson, et al., 2010a) and are used here to illustrate the optimisation capabilities and limitations of our model in a domain-independent manner. These investigations illustrate the conceptual idea behind *rHN-G* and also help us to understand exactly the limitations of the model and, in particular, *MC* assists us in understanding the limitations of selective associations and the different affordances of generative associations.

The aim in defining *MC* is to construct an idealised exemplar – a system that can be solved easily by a process that can discover and exploit structure appropriately, but one that is, at the same time, pathologically difficult for a process that is unable to discover and exploit modularity. At the micro-scale *MC* has neat and easily identifiable modular structure, but at the macro-scale it has a random structure.

RC. **Random Constraints**: A symmetric weight matrix of n variables where weights take values: $\omega_{ij}=1$, if $i=j$ else $\omega_{ij} = \{-p, p\}$ with equal probability.

MC. **Modular Constraints:** A symmetric modular connectivity weight matrix of size $N=kn$ variables: $\omega_{ij}=R\left(\left\lfloor \frac{i}{k} \right\rfloor, \left\lfloor \frac{j}{k} \right\rfloor\right)$, where R is a matrix from the *RC* definition above with size n.

(for example, the main experiments use N=200, n=20, k=10 to define 20 modules of 10 variables each).

A small example *RC* matrix and corresponding *MC* matrix is given in Table. 1. The *MC* matrix is simply a ‘macro-scale’ version of the *RC* matrix, but to exploit this structure successfully an algorithm will need to first identify which variables in *MC* belong together in a module and then search combinations of module sub-solutions as though each module were a single ‘emergent’ variable.

1	0.1	-0.1	0.1
0.1	1	-0.1	**-0.1**
-0.1	-0.1	1	-0.1
0.1	-0.1	-0.1	1

1	1	1	0.1	0.1	0.1	-0.1	-0.1	-0.1	0.1	0.1	0.1
1	1	1	0.1	0.1	0.1	-0.1	-0.1	-0.1	0.1	0.1	0.1
1	1	1	0.1	0.1	0.1	-0.1	-0.1	-0.1	0.1	0.1	0.1
0.1	0.1	0.1	1	1	1	-0.1	-0.1	-0.1	**-0.1**	**-0.1**	**-0.1**
0.1	0.1	0.1	1	1	1	-0.1	-0.1	-0.1	**-0.1**	**-0.1**	**-0.1**
0.1	0.1	0.1	1	1	1	-0.1	-0.1	-0.1	**-0.1**	**-0.1**	**-0.1**
-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	1	1	1	-0.1	-0.1	-0.1
-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	1	1	1	-0.1	-0.1	-0.1
-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	1	1	1	-0.1	-0.1	-0.1
0.1	0.1	0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	1	1	1
0.1	0.1	0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	1	1	1
0.1	0.1	0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	1	1	1

Table. 1. Left) A small random *RC* matrix (n=4, p=0.1). Right) A corresponding modular, *MC*, matrix (N=12, n=4, k=3). For illustration, one of the weights in *R* and the corresponding set of weights in *MC* are shown in bold.

The *RC* matrices create energy functions that tend to have many locally optimal configurations but for problems with n=20 it is not particularly difficult to find low-energy configurations (i.e. significantly better than random configurations) using *rHN-0* (i.e. a restart bit-flip hill-climber). A corresponding *MC* with k=10, creating a problem with N=200, is in contrast, very difficult to optimise effectively with a local search process. Given that p is sufficiently small, the relatively strong and consistent intra-module weights on the block diagonal create local optima that correspond to each of the state configurations in the language, $S=\{-1^{k},1^{k}\}^{n}$ (where n is the number of modules), and k is the size of each module (Fig. 2). For example, in a *MC* system with k=5 and n=6 the state configuration (1,1,1,1,1,-1,-1,-1,-1,-1,1,1,1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,1,1,1,1,1), or $(1^{5}\text{-}1^{5}1^{5}1^{5}\text{-}1^{5}1^{5})$, is locally optimal. Note that such a configuration is equivalent to the state configuration (1,-1,1,1,-1,1) in the underlying *R* problem, and the set of *locally optimal* strings in *MC* correspond to *arbitrary* binary-strings in this lower-dimensional state space. Accordingly, solutions found by running *rHN-0* on *MC* will (with respect to inter-module dependencies) be no better than random configurations in *R*. Moreover, the likelihood of escaping such locally optimal configurations in *MC* using random k-bit state changes would decrease exponentially with k even if one knew which bits needed changing (but not what values to change them to) and decreases exponentially in N if the module partitions are unknown (Watson & Jansen, 2007).

But, in the case where an adaptive system is able to identify the high-level problem structure in *MC*, and manipulate modules as if they were ‘emergent bits’ of a lower-dimensional macro-scale problem, *MC* problems will become as easy as hill-climbing in the underlying problem *R*. (Note that the ordering of rows and columns in *MC* may be randomly shuffled thus precluding any algorithm from exploiting adjacency information as a heuristic for identifying problem structure (Watson, 2006). Likewise, applying logical-XOR with a constant, randomly generated bit-string may be applied to the state configurations to preclude any assumption that all 1s or all -1s are special configurations.) In *MC* problems we will often measure only the energy contribution from inter-module constraints since the intra-module constraints are easily resolved and any resolution to them adds the same constant to total energy.

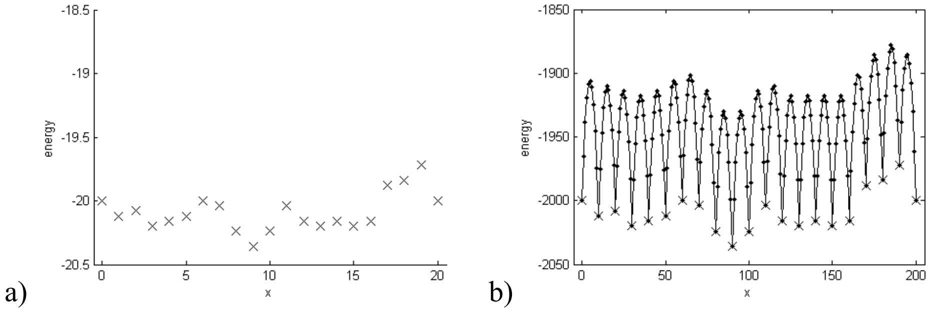

Fig. 2 Cross-sections through example energy surfaces: (a) *RC* energy surface, (b) corresponding *MC* energy surface. Specifically, the cross-section created by the sequence of state configurations in the language $-1^{x}1^{(N-x)}$ for x=[0, N]. 'X' markers in (b) indicate the subset of state configurations that are locally optimal (where all states that are in the same module take the same value, i.e. configurations in the language $-1^{kx}1^{(N-kx)}$ for x=[0, n]). These correspond exactly to the points in (a) when adjusted in magnitude by a factor of p.

Energy-minimising state-space trajectories in *MC* (as described by *rHN-0*) create an idealised nearly-decomposable dynamical system (Simon, 1969; Watson, 2006; Watson & Pollack, 2005). The strong dependencies within modules in *MC* are quickly and easily resolved and the aggregate effect of one module on another is produced via many pairwise connections that act in a coherent direction (governed by the corresponding entry in *R*). This effects an essentially low-dimensional interaction between any two modules (see also '*R-matrix*', Wagner, Booth & Bagheri-Chaichian, 1997). This utilises our 'modular interdependency' formalisation (Watson, 2006; Watson & Pollack, 2005) of the nearly-decomposable concept in an optimisation framework. In practical terms *MC* can be loosely interpreted as a modular optimisation problem such as a resource-allocation problem with modular constraints (Watson, et al., 2010a). It is worth emphasising that *if* we were to approach the *MC* problem using a sensible problem representation that exploited the underlying *n*-dimensional search space defined by *R*, then it would be a fairly easy problem; But, our dynamical systems are not given such a representation, and without it, local search in the *N*-dimensional space cannot resolve the constraints of the underlying problem *R*. If the global optimum of *R* is unique then the basin of attraction for it under local search in *MC* may be as small as 2^{-n} of the configuration space, and to escape from any local optimum will require a specific *k*-bit change to find a configuration of higher-fitness.

3. Simulation Results

Fig. 3 shows some example trajectories of *rHN-0* in a random (*RC*) problem. We see clearly that this problem has several different locally optimal configurations but it is not difficult to make significant reductions in energy (compared to the initial conditions) using *rHN-0*.

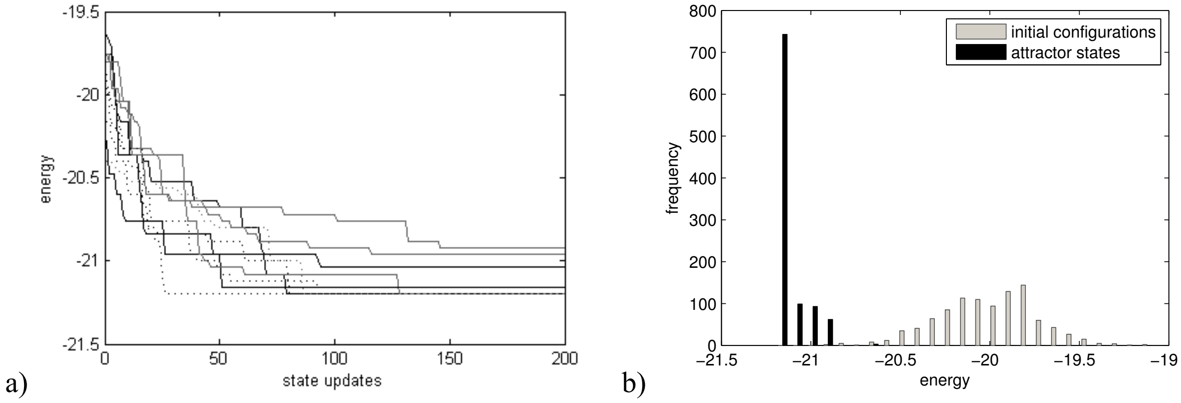

Fig. 3 a) ten example trajectories of *rHN-0* in a random (*RC*) problem (*n*=20). b) the distribution of energies for initial (random) configurations in comparison to final (locally optimised) configurations (1000 samples of each).

Fig. 4 shows some example trajectories of *rHN-0* in the modular problem corresponding to the random problem of Fig. 3. We see that *rHN-0* makes significant reductions in energy as we might expect. But, as predicted, *rHN-0* is unable to resolve any of the *inter-module* dependencies in *MC* and improvements shown in Fig. 4.a come solely from resolving the trivial-to-solve intra-module dependencies. This is revealed in Fig. 4.b where only the energy contribution from inter-module weights are plotted. These show almost no improvement on average because the intra-module weights are over-powering inter-module weights and causing the network to settle on whichever random configuration of *R* was closest to the initial condition.

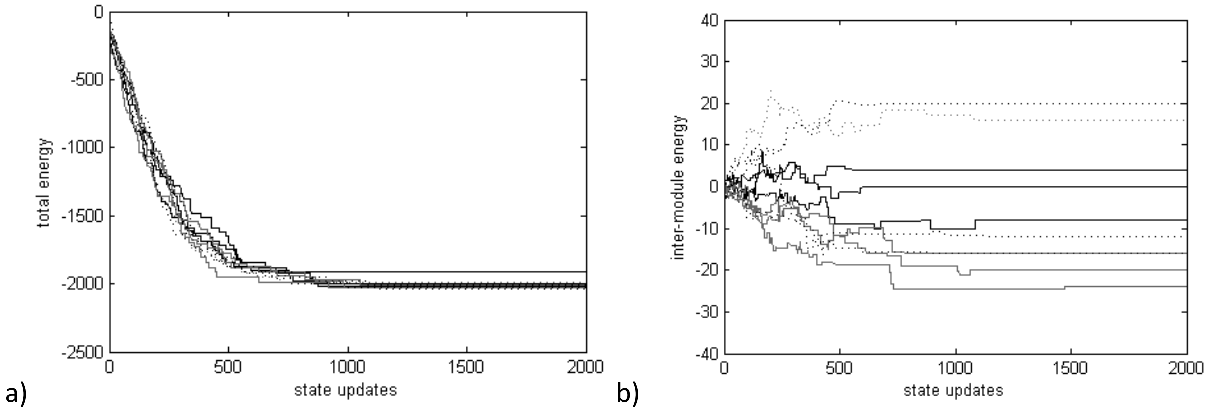

Fig. 4 Ten example relaxations of *rHN-0* on modular problem corresponding to random problem of Fig. 3 (here, and henceforth except where noted, *n*=20, *k*=10, *N*=200, p=0.01, τ =2000). a) reporting total energy, b) reporting only the energy contributions from inter-module constraints. Values in (b) may increase because inter-module constraints are violated in favour of satisfying relatively strong intra-module constraints.

In Fig. 5 we apply *rHN-G* to the modular problem (reporting energies from the inter-module constraints only, as per Fig. 4.b). The initial trajectories are identical to those of *rHN-0* (compare Fig. 5.a and Fig. 4.b). The distribution of the end-points of these trajectories, i.e. locally optimal attractors, is shown in relaxations 1-1000 of Fig. 5.b. When associative learning is activated (at relaxation 1001 onwards) we see that within a very small number of relaxations *rHN-G* has significantly improved its ability to resolve inter-module constraints and the distribution of attractor states is very different from *rHN-0*. After associations have been learned, we see trajectories that fall in energy faster and to lower energy attractors (Fig. 5.c). By the end of the learning process, *rHN-G* is not only climbing out of the locally-optimal configurations created by the intra-module weights, but it is consistently finding the best resolution of inter-module weights as well. This is because *rHN-G* is not (merely) equivalent to running *rHN-0* on the underlying *RC* problem, but in fact it becomes equivalent to running itself, i.e. *rHN-G*, on the underlying *RC* problem – with the consequence of canalising correlations in this reduced search space also.

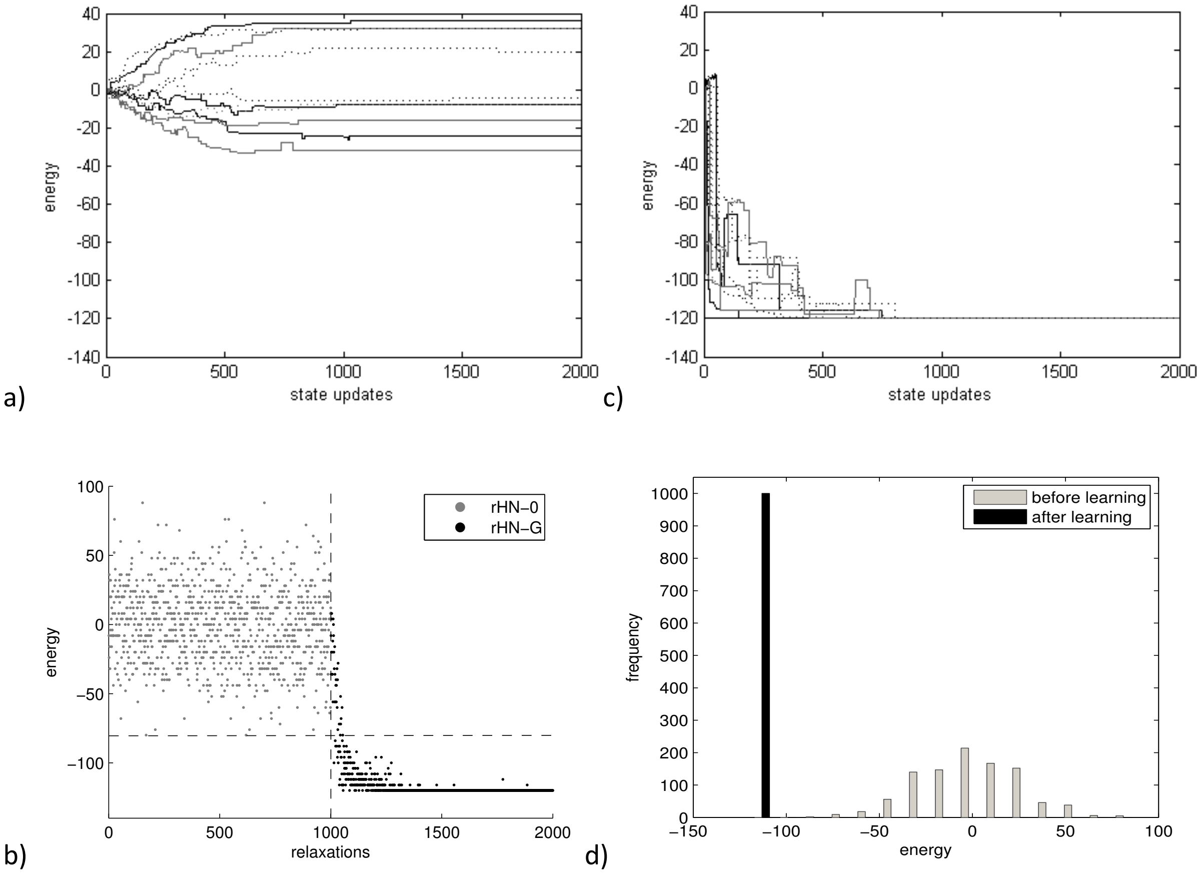

Fig. 5 *rHN-G* applied to the modular problem (Fig. 4), (energies measured using only inter-module constraints). a) ten example trajectories before learning, b) end points of all relaxations without learning (relaxations 1-1000) and during learning (1001-2000), the broken horizontal line indicates the lowest energy attractor found over all non-learning relaxations, c) 10 example trajectories after learning, d) histogram of energies found before learning (=*rHN-0*) and after learning. Example trajectories show that before learning (equivalent to *rHN-0*) inter-module constraints are not resolved on average (mean reduction in energy is negligible), whereas after learning *rHN-G* is resolving many inter-module constraints very quickly. (learning rate for *rHN-G*, δ=0.0003).

Fig. 5 already shows that generative associations significantly improve the ability of the system to globally optimise constraints; but is *rHN-G* succeeding in the manner we hypothesised? Is it transforming the scale of behaviour/re-scaling the search space to search in combinations of modules instead of combinations of the original problem variables? Because we know where the modules are in *MC* we can easily assess whether *rHN-G* is creating state changes that correspond to modules. Fig. 6

examines the behaviour of *rHN-G* on the modular problem in more detail. This shows that *rHN-G* learns the intra-module correlations quickly (Fig. 6.a) revealing the modular structure of the problem correctly (Fig. 6.b). Note that the individual squares in Fig. 6.b are blocks of 10x10 variables, not individual pixels - this neatness results from learning intra-module correlations from locally optimal configurations. *rHN-G* exploits this learned structure to create correlated state changes that correspond to modules (Fig. 6.c - e). Thus *rHN-G* is effectively moving from one local minimum to another, corresponding to the discrete points in Fig. 2.a, and hence behaving very like *rHN-0* would behave in the underlying *R* problem.

Moreover, towards the end of this experiment, *rHN-G* is recalling combinations of multiple modules of very high quality (Fig. 6.f - i) – thus it not only learns the basic modularity of the problem in order to operate like *rHN-0* at the macro-scale but additionally, learns inter-module structure above the level of individual modules. This means that it is no longer using search in the space of module combinations but is simply recalling more or less complete state configurations – so the ability to exploit modularity, although vital to finding good solutions, is transient and becomes redundant (this is reminiscent of (Hinton & Nowlan, 1987; Mills & Watson, 2005) where the ability to find solutions becomes redundant when complete solutions can be remembered/inherited). There is, of course, no special feature of the system that pre-specifies the scale of problem structure that might be discovered, so if higher-level correlations exist, learned multi-module correlations that exploit them are to be expected (this indicates potential for exploiting hierarchical (Lenaerts, Chu, & Watson, 2005; Watson, 2006) or multi-scale modularity not examined in this paper).

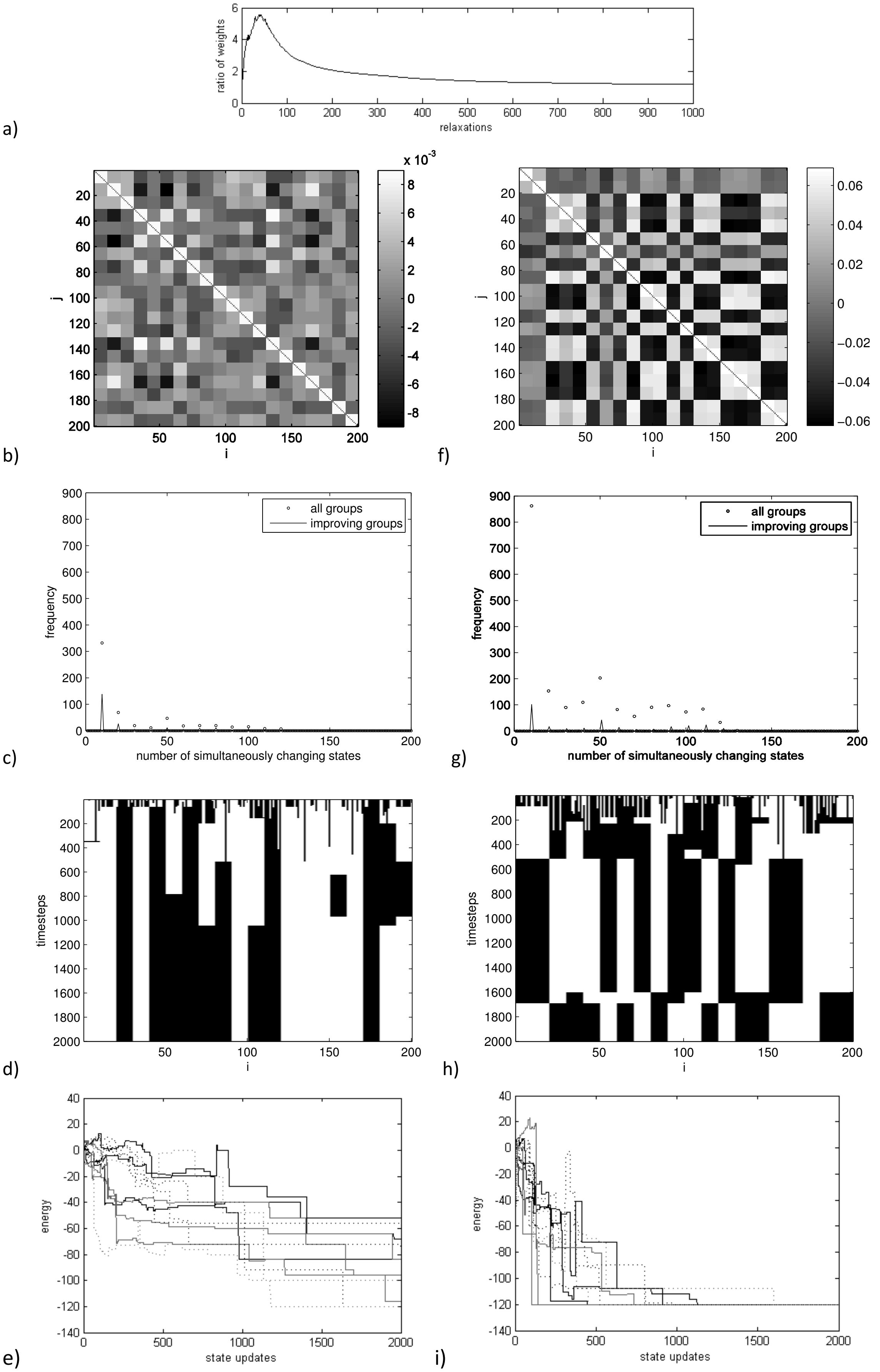

Fig. 6 Details of *rHN-G* on *MC*. a) The ratio of learned intra-module to inter-module average weight magnitudes. This shows that for some period the correlations identified within each module are more than five times

stronger than the correlations between variables in different modules. b) The learned weight matrix, *M*, after 30 relaxations. This shows that the intra-module (block-diagonal) weights are strongly positive whilst the inter-module weights are lower in magnitude and vary in sign (m_{ii} values are visualised as 0 to facilitate appropriate image scaling). Note that the individual squares apparent in this figure are not individual pixels but blocks of 10x10 pixels (compare with Fig. 7.a). c) The frequency of state changes generated by this matrix that change more than one state variable simultaneously, and the frequency with which these improve the solution (i.e. lower energy). These are measured during relaxations 30-60, showing that at this stage 'block-flips' of size *k*=10 are common, but changes smaller than this do not occur (except changes of size 1). d) example state configurations over one relaxation (relaxation 60) visualising example block-flips, e) example trajectories (relaxations 50-60, energies measured using only inter-module constraints), f)-i) as per *b-e*, for relaxations 230 to 260, showing that at this later stage in learning *rHN-G* is making more multi-state changes and a greater proportion of them are likely to make multi-block state changes that update several modules simultaneously – i.e. it is recalling useful combinations of modules, and creating larger instantaneous decreases in energy.

Fig. 6 shows that generative associations enable the system to sample specific high-quality configurations that are distant in Hamming space from the current configuration without visiting the lower-quality intermediate solutions, and this gives the system the opportunity to escape from poor quality, locally-optimal solutions that would otherwise trap the local search process. These operate by effectively modifying the neighbourhood of the space such that particular correlated macro-variations occur with high probability. A search trajectory using these correlated changes may violate local energy minimisation by skipping over intermediate configurations of higher energy.

The ability to exploit structure in this manner is not to be taken for granted despite the neat structure of the modules in the *MC* problem. Fig. 7 is a control experiment using relaxations of length 200 timesteps rather than 2000 (measured over the same number of learning updates, relaxations 30-60, as per Fig. 6 b-e.) This therefore reduces the separation of timescales between the use of associations (occurring every timestep) and the modification of associations (occurring at the end of each relaxation). These relaxations are not long enough to ensure that a local optimum is reached, and therefore the associations learned are much less neat than those shown in Fig. 6.a. Although the learned weight matrix does still show evidence of the modules on the diagonal, the average size of multi-variable state changes is not neatly peaked at multiples of *k* (compare Fig. 7.b with Fig. 6.c), and the multi-state changes that do occur do not necessarily correspond to whole blocks (compare Fig. 7.c with Fig. 6.d). As a result, the ability to reduce energy (Fig. 7.e) is severely impaired.

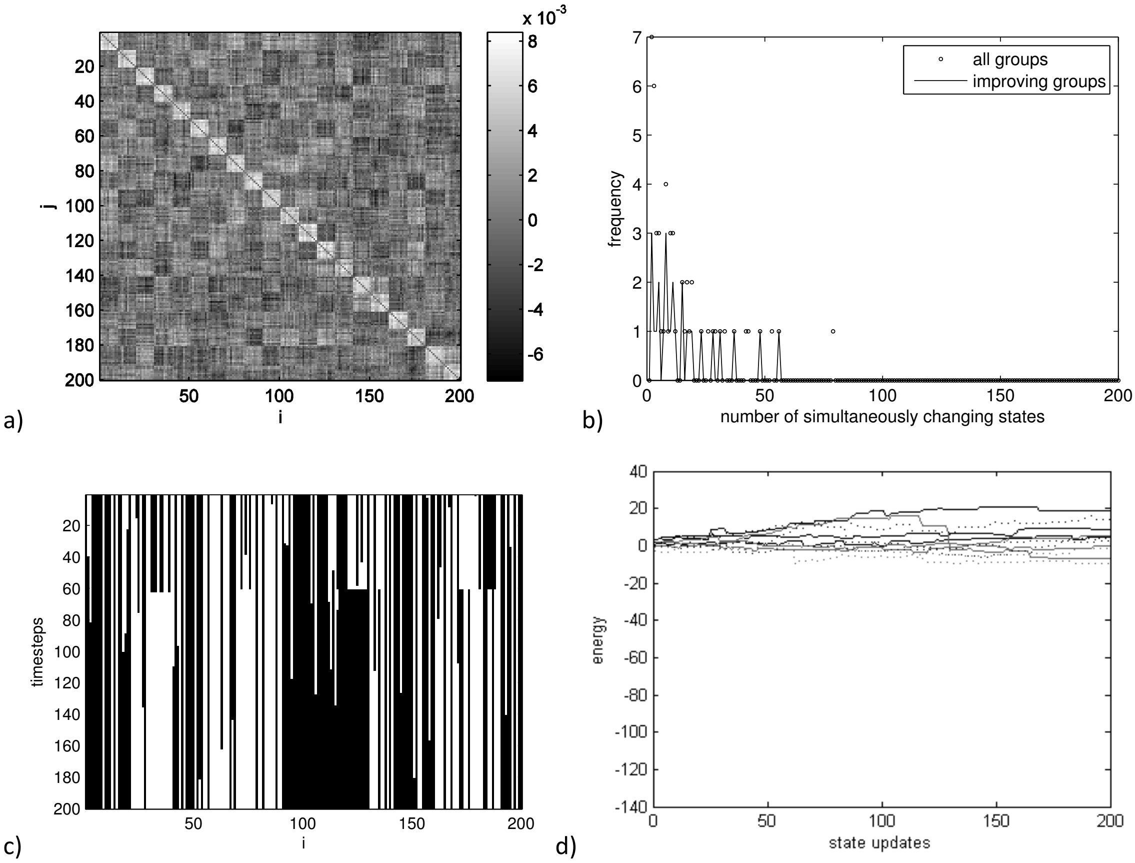

Fig. 7 Details of *rHN-G* on *MC* using short relaxations (t=200). Frames *a-d* are otherwise as per frames *b-e* of Fig. 6.

In Fig. 8 we compare *rHN-G* with the ability of *rHN-S*. We see that when *p*=0.1 *rHN-S* is able to find good quality solutions (albeit more slowly than *rHN-G*). With inter-module dependencies of this strength, all configurations that resolve intra-module constraints are (still) locally optimal, but local optima where more inter-module constraints are resolved have larger basins of attraction than those where less inter-module constraints are resolved (this is true to some extent for all non-zero *p*). Thus some correlations between modules are observable in the initial dynamics of the system that *rHN-S* can learn and exploit. However *rHN-S* can only make progress when inter-module weights are strengthened sufficiently to remove local optima (Watson, et al., 2010a; Watson et al., 2009a), and when inter-module weights are initially weak this must be done slowly to avoid accidentally reinforcing spurious correlations. *rHN-G*, in contrast, is not limited by weak between-module correlations (see Supplementary on-line materials); so long as it can learn the within-module correlations (which are easy to identify from locally optimal configurations) it will be able to flip/exchange one module-solution for another as required to search configurations at the macro scale. It can thereby reveal whether one module configuration resolves the inter-module dependencies better than another. This difficulty of learning inter-module correlations with selective associations is exacerbated when inter-module weights are weaker (Fig. 8.c and d), but generative associations are relatively insensitive to this change since the intra-module correlations that need to be learned are still obvious (see also Fig. S1). Once the intra-module correlations are encapsulated, the energy-changes gained by resolving inter-module constraints are easily revealed even if small.

Note that the difference between *rHN-S* and *rHN-G* in their ability to resolve inter-module dependencies is greatest when inter-module weights are weakest, but in the limit, if inter-module weights are very weak then there is correspondingly little to be gained by resolving them. An examination of this trade-off in a simplified version of the *MC* problem (Mills, 2010; Watson et al., 2009a) and Supplementary on-line material) indicates that there is, nonetheless, a significant middle ground where inter-module dependencies are difficult to learn but make a significant contribution to system energy when resolved (Fig. S1).

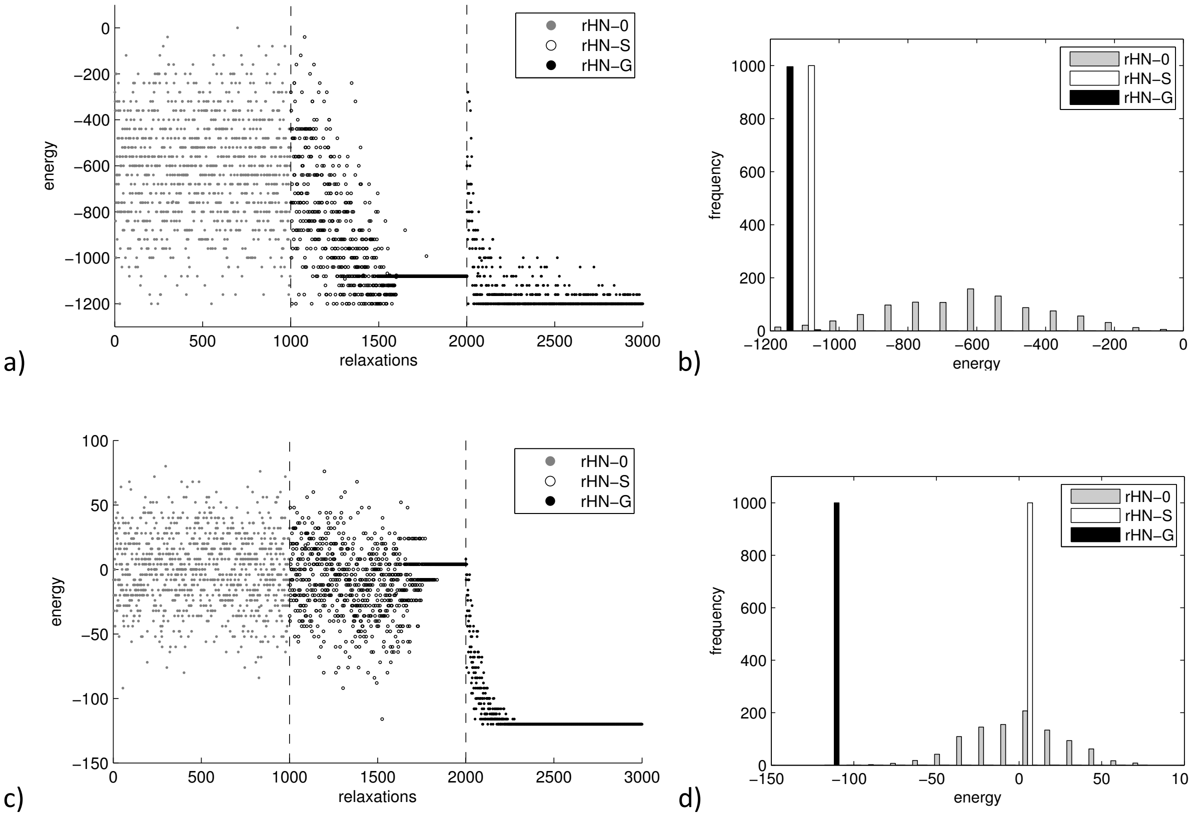

Fig. 8 Comparison of *rHN-G*, *rHN-S* and *rHN-0* on *MC*. a) Energies (inter-module constraints only) of relaxation end-points over 1000 relaxations (relaxations 1-1000 *rHN-0*, 1001-2000 *rHN-S*, 2001-3000 *rHN-G, p=0.1, δ* for *rHN-S* =0.00075). b) energy distributions of end-points after 1000 relaxations of learning. c and d) as per a and b with *p*=0.01, δ for *rHN-S* =0.0015. Showing that *rHN-S* can resolve most inter-module constraints (albeit more slowly) if they are sufficiently strong compared to intra-module constraints, but not otherwise. Also, when inter-module constraints are sufficiently strong for *rHN-S* to operate effectively, although it is successful in increasing the probability of retrieving good solutions, it does not exceed the quality of the best solutions found by rHN-0. Learning rates, δ, for *rHN-S* are hand-tuned to converge as late as possible in the 1000 relaxations to give *rHN-S* the best chance of converging on low-energy configurations (learning rate for *rHN-G* is unchanged, δ=0.0003).

4. Discussion

The above results illustrate how a simple distributed associative mechanism can modify the structure and subsequent behaviour of a dynamical system in a manner that rescales, or coarse-grains, the dynamics of the system to a higher level of organisation. This affords search in a higher-level abstraction of the original configuration space and has the effect of producing more reliable and more complete minimisation of the system energy. *rHN-G* is a self-modelling dynamical system that rescales its own dynamics over time using simple distributed mechanisms that alter the relationships between its components using only local information (i.e. pairwise correlations). Using learned associations in the generative sense (rather than the selective sense) is intrinsic to this result. This shows that, in a manner previously only seen in relatively sophisticated model-building optimisation algorithms, distributed dynamical systems can be transformed by very simple distributed processes into systems that resolve constraints between components more reliably and more completely.

In the following sections we briefly discuss some related optimisation concepts before discussing various analogues of the underlying dynamical processes in other complex adaptive systems.

4.1 Optimisation and computational complexity analysis

The optimisation behaviour of generative associations in *rHN-G* depends on the presence of problem structure that can be recognised and exploited. However, in related work (Mills, 2010), results on minimising the energy of a 2D Ising model demonstrate that an *explicit* modular structure, as per *MC*, is not required to observe the advantages of an algorithm that automatically identifies and exploits problem decomposition in this manner. In this case, correlations in locally optimal configurations reveal the lattice structure of the problem and learned associations enable a re-scaling that effects changes in 'domain-spins' rather than the spins of individual variables. This work also shows that whereas *rHN-S* requires inter-module dependencies to be strong in order to learn and exploit them, an algorithm like *rHN-G* can exploit modules even in the case where inter-module dependencies are obfuscated by relatively strong within-module constraints (see also, Supplementary on-line material).

In the results above we see that the relevant modules are discovered surprisingly quickly - within about 30 relaxations. The lower limit on this number is determined by the probability of observing within-module correlations *without* observing *spurious* between-module correlations. In fact, any single training sample will reinforce some association between any pair of modules, so learned associations must develop slowly enough for many observed configurations to average-out spurious correlations. Analytic work (Mills, 2010) shows that this can be done in surprisingly few observations when local optima are used as the training samples. (In Mills, 2010, each observation is independent, whereas in *rHN-G* even the first observation biases the second observation to some extent; However, the results here indicate that this is not a significant hindrance).

Our model therefore illustrates how a distributed learning mechanism can be used to identify and exploit problem structure efficiently without *a priori* domain knowledge. One might reasonably question, however, exactly how the cost of inducing a model (that subsequently makes solving the problem easy) compares with the cost of 'brute forcing' the solution without a model. In the Supplementary on-line material we show formally that it is possible in some problems to learn the model and solve the problem in time that is a polynomial function of the system size whereas solving the problem without the model takes exponential time. This result is dependent on the presence of appropriate problem structure, of course (Wolpert & Macready, 1997), but there exist systems, namely modular ones, where the mechanism demonstrated can find low-energy configurations in polynomial time that cannot be found by a non-adaptive model in less than exponential time on average. The Supplementary material also shows a numerical study of the sensitivity of *rHN-0*, *rHN-S* and a variant of *rHN-G* (Mills, 2010) to the strength of inter-module connections, *p*. This study shows that there is a significant region of *p* where the problem is difficult for *rHN-0* and that both *rHN-S* and *rHN-G* offer an advantage in this region. But whereas *rHN-S* finds low *p* increasingly difficult (because inter-module dependencies are over-powered by intra-module dependencies), *rHN-G* is almost insensitive to *p*.

4.2 Building-blocks and model-building

The idea of discovering and exploiting modules has been a 'holy grail' for a sub-school of evolutionary computation ever since the inception of the field (Goldberg, 1989; Holland, 2000).

Initially, the idea was that sexual recombination or 'crossover' would automatically identify good combinations of genes ('building blocks') and combine them to find better, larger combinations of genes. However, there has been significant difficulty in illustrating that genetic algorithms, GAs, can achieve this (Forrest & Mitchell, 1993; Watson, 2006). Importantly, it is crucial to realise that, although a GA with crossover can, under some circumstances, identify and recombine building-blocks with provably superior time complexity to a non-crossover GA (Watson, 2006; Watson, Weinreich & Wakeley, 2011; Watson & Jansen, 2007), this advantage relies intrinsically on the correspondence between epistatic dependencies and physical linkage – i.e. the bits of a building-block need to be next to each other on the genome. Thus the canonical GA cannot solve a problem like that which we address here, where adjacency information cannot be used (note that *rHN-G*'s ability to learn linkage information from observed correlations is entirely insensitive to the ordering of problem variables).

Accordingly, there has been considerable research effort directed at methods that discover and exploit building-blocks without presupposing this 'tight linkage', as it is known. So called 'linkage learning' GAs and 'model-building' optimisation algorithms include 'estimation distribution algorithms' like the Bayesian Optimisation Algorithm (Pelikan, Goldberg & Lobo, 2002; Yu, Goldberg, Yassine & Chen, 2003; Pelikan, 2002). *rHN-G* can be understood as a form of model-building optimisation algorithm (see, in particular, bivariate models, e.g. Yu, Goldberg, Yassine & Chen, 2003) but in addition to the differing conceptual motivations and distributed implementation there are also two important algorithmic distinctions (Iclanzan & Dumitrescu, 2007; Mills & Watson, 2011; Mills, 2010): 1) Whereas previous methods have used induced models to bias the initial conditions of further search, we use the model to provide variation in a lower-dimensional space. This is essential to scale-up the search process properly (as per the original building-block intuition, Watson, 2006). 2) Whereas previous methods build a model from correlations observed in above-average-fitness configurations, we build the model from locally optimal configurations. This greatly amplifies the 'signal' from which correlations can be learned and helps to separate meaningful correlations from spurious 'noise' (see Fig. 7). A separation of timescales, using repeated local search from many different initial conditions whilst slowly learning problem structure, is essential to achieve this (Watson, et al., 2010a; Mills, 2010). These features (see also Iclanzan & Dumitrescu, 2007) are central to the behaviour illustrated in this paper and are developed formally in related work on multi-scale search (Mills, 2010; Mills & Watson, 2011). But crucially, in the current paper these behaviours are modelled in a distributed system without using centralised data structures or a centralised control architecture.

4.3 Implications for analogous systems

Cognitive problem solving: Although we are using a neural network model, our main research questions are not motivated by cognitive systems. Nonetheless, distributed constraint optimisation is relevant to cognitive problem solving (Ackley, Hinton & Sejnowski, 1985; Hinton & Sejnowski, 1983). Implications for problem solving of the self-modelling framework with Hebbian learning are discussed by Fernando et al (Fernando, Goldstein & Szathmáry, 2010). These authors use Hebbian learning to modify the trajectory of a random-mutation hill-climbing process as per *rHN-S*[1]. Fernando et al suggest that the ability to (re)structure a local search process in this manner is relevant to solving 'insight problems' that are otherwise problematic for local search processes. The additional ability (as demonstrated by *rHN-G*) to abstract-away or encapsulate low-level details and thereby reason about a

[1] Although these authors attribute the improved optimisation ability to natural selection (Fernando & Szathmáry, 2010), no replication is necessary to implement this algorithm, as shown here and in (Watson et al., 2009a; Watson, et al., 2010a), and the associative network employed in (Fernando, Goldstein & Szathmáry, 2010) is not evolved but learned via Hebb's rule from locally-optimal solutions as per *rHN-S*.

system using higher-level features or concepts, suggests aspects of cognitive problem solving that are also intuitive (Chase & Simon, 1973; Miller, 1956) and have been exploited in classical (symbolic) AI (Rosenbloom, & Newell, 1986), but not previously shown in a distributed neural architecture. Crucially, such processes enable improvements to a proposed solution/neural activation pattern that cannot be explained by incremental improvement in the original space of problem variables/neural activations since they involve (individually inferior) changes to multiple variables that are beneficial only in combination.

Multi-agent systems/games on networks: The default dynamics defined by the Hopfield equation are analogous to those of a multi-agent system where agents play pairwise games on a network (Pacheco, Traulsen, & Nowak 2006; Davies, Watson, Mills, Buckley, Noble, 2010; Van Segbroeck, Santos, Pacheco, Lenaerts, 2010). Each connection in the network represents a game between two agents. The pay-offs for those games, defined by Ω, constitute a mixture of coordination games ($\omega_{ij}>0$) and anti-coordination games ($\omega_{ij}<0$) (Davies, Watson, Mills, Buckley, Noble, 2010). Each agent (node) adopts the strategy (state) that maximises their pay-off given the conflicting constraints of the multiple games in which they participate. The symmetric nature of connections ($\omega_{ij}=\omega_{ji}$) ensures that each interaction is a potential game, and the fact that the utility of an agent is a sum over many potential games ensures that the total system can be described as a potential game also (Chapman, Rogers, Jennings & Leslie, 20; Chapman, 2009), therefore exhibiting only fixed-point attractors/Nash equilibria. Solving the constraint optimisation problem defined by Ω is then equivalent to the problem of maximising total welfare in the corresponding multi-agent network (Chapman, 2009), and local optima found by *rHN-0* correspond to Nash equilibria where no single agent can improve their utility by changing strategy unilaterally.

Our recent work shows that if agents gradually change their 'perception' of the utility function with another player, so as to prefer the situations they currently find themselves in, then these preferences modify subsequent agent interactions according to Hebbian principles (Watson, et al., 2010c), causing the system as a whole to improve its ability to find configurations of high total welfare in the same manner as *rHN-S* (Davies, Watson, Mills, Buckley, Noble, 2010). This is closely related to a simple form of reinforcement learning where agents play the strategy that maximises utility given the expected strategies adopted by other agents based on past behaviour (Leslie & Collins, 2006; Watson, et al., 2010c; Davies, Watson, Mills, Buckley, Noble, 2010). But given rapid relaxations and slowly changing associations (Watson, et al., 2010a), a simple habituation of *observed* correlations at equilibria achieves the same outcome as a preferential reinforcement of *good* correlations (Davies, Watson, Mills, Buckley, Noble, 2010). Generative associations in such a system would be analogous to the formation of coalitions where agents use a correlating mechanism (i.e. the random variable *X*, in Eq. 7) to enact simultaneous strategy changes and share the utility gained (i.e. adopt strategy changes if they increase group utility) (Leslie & Collins, 2006; Chapman, Rogers, Jennings & Leslie, 20; Chapman, 2009; Young, 2001; Hart & Mas-Colell, 2000).

Ecological networks: Our recent work has investigated a Lotka-Volterra model of an ecosystem of multiple species with evolvable inter-species fitness dependencies analogous to selective associations. This work shows that natural selection at the species level to alter these fitness dependencies results in associative learning and the system level (Lewis, 2009). An abstract model of evolved symbiotic relationships that effect changes in the unit of selection (Watson, et al., 2009b; Mills & Watson, 2009; Watson, Palmius, Jackson, Mills, Powers, Buckley & Penn, in prep.) (see also Powers, Mills, Penn, Watson, 2009; Powers, Penn & Watson, 2007; Powers, 2010 for discussion of general concepts) illustrates *rHN-G*-style adaptation in an evolving ecosystem. This latter work provides a model for the major transitions in evolution where "entities that were capable of independent replication before the transition can replicate only as part of a larger whole after the transition" (Maynard Smith &

Szathmary, 1995; Watson & Pollack, 2003; Watson, 2006). In that work we endow agents with the ability to affect the probability of 'co-dispersal' with other species and we observe that the associations they evolve create new units of selection that canalise naturally co-occurring species in the same manner that *rHN-G* creates 'emergent macro-variables' that canalise partial solutions. In the current neural network model we are better able to formalise the adaptive capabilities of this process and understand its relationship to associative learning.

Gene expression networks: At a completely different biological scale, the same principles apply to the evolution of connections in a gene expression network (Watson, et al., 2010b). That is, if a gene expression network, controlling the mapping between genotype and phenotype, is evolved on a given fitness landscape it will tend to increase the robustness of, or 'canalise' (Wagner, Booth & Bagheri-Chaichian, 1997; Siegal & Bergman, 2002; Riegler, 2008), phenotypes that are locally optimal. Evolved changes to the connections in this network therefore build-up an associative memory of past selective environments that enables superior ability to maximise fitness. We are not yet sure whether this result is analogous to *rHN-S* or *rHN-G* – both seem to be possibilities.

Since canalisation reduces the effective degrees of freedom of a system, it might seem counter-intuitive that it has the effect of increasing adaptability or evolvability in this manner. But canalisation and evolvability are really two sides of the same coin (Kirchner & Gerhart, 1998; Watson, et al., 2010b; Draghi, Parsons, Wagner & Plotkin, 2010). That is, removing some degrees of freedom whilst retaining others can enhance the ability of the system to find solutions by focussing variation in useful regions of configuration space. These ideas are related to evolved exploration distributions (Toussaint & von Seelen, 2007; Izquierdo & Fernando, 2008; Jones, Arnold & Bürger, 2007) and 'facilitated variation' where an evolved 'memory' of past selective environments and generalisation to new selective environments has also been suggested (Parter, Kashtan & Alon, 2008).

General Dynamical Systems: The notion that self-organisation can transform a high-dimensional (disordered) system into a low-dimensional (ordered) system is familiar in many domains. Often we are interested in characterising the emergent low-dimensional variables (see 'order parameters', Haken, 1983) to provide a description that represents this order, and in using these to characterise a transformation in the system dynamics to a higher-level of organisation (Fuller, 1975). Clearly, the behaviour of *rHN-G* with learned associations on an *N*-dimensional *MC* problem has a low-dimensional description – the learned associations reduce the effective degrees of freedom of the system. And it arrived at this configuration of associations by self-organisation; by the application of simple local rules that change the relationships between system components in a manner determined by its own dynamics.

However, we are not merely interested in the fact that the system has a low-dimensional description, but more specifically, we are interested in the similarities and differences between this new low-dimensional behaviour and the original behaviour of the system. That is, after some learning (Fig.6.b-e) *rHN-G* behaves like *rHN-0* on the underlying *n*(=*N/k*)-dimensional *RC* problem. To some degree, this is similar to a coarse-grained description of the behaviour of its *former self*, i.e., of *rHN-G* without learned associations (i.e. *rHN-0*) on the full *N*-dimensional *MC* problem. Thus, the emergent variables in *rHN-G* reflect to the modularity/correlations in its initial dynamics. But *rHN-G* with learned associations is not exactly a low-dimensional description of its former self – if it were, it would reach the same attractors as *rHN-0*, but it reaches a different distribution of attractors than *rHN-0*. It does not find the same distribution of attractors that *rHN-0* finds on the full *N*-dimensional *MC* problem; it finds the distribution of attractors that *rHN-0* finds on the underlying *n*-dimensional *RC* problem. Accordingly, *rHN-G* is not just transforming into any low-dimensional system, nor more specifically into a low-dimensional model of itself – but, in fact, it is transformed into a new system, one with *improved* ability to minimise energy. It does so by creating short-cuts in the neighbourhood structure of state-configuration space; enabling movements in a lower-dimensional space that skips over intermediate configurations of higher-energy.

We can make some sense of this idea of a dynamical system that spontaneously improves by taking a 'meta-dynamical system' perspective (Bourgine & Varela, 1992). Because the association variables are treated separately from the state variables we can treat them as *parameters* of the state dynamics even though they are in actuality (slow changing) *variables*. In the joint space of state variables and associations together we simply observe a dynamical system doing what it does naturally – there is no sense in which the system is improving its ability to minimise energy. But if we regard the associations as parameters of the state dynamics, then we can characterise the changes in the associations by how they transform the dynamics of the state variables. In particular, we can assess whether this transformation improves the ability of the state dynamics to minimise energy, as we have shown. This enables us to characterise the system behaviour as 'problem solving' rather than just self-organising, and it enables us to say that the system is transformed into a different system (with respect to the behaviour of its state dynamics), even though at the same time it is merely the same system in a new configuration (when described in terms of states and associations together). Likewise we can characterise how the evolution of robustness effects an increase in evolvability (an improvement in ability to evolve), even though at the same time it is merely the (normal) evolution of a given system (the genotype and the gene regulation network together) that produced this result (Watson, et al., 2010b). This meta-dynamical systems perspective is useful in other, non-network, models too – for example, the evolution of social traits with the concurrent evolution individual traits that alter the contextual parameters of that social behaviour (Powers, Penn & Watson, 2011). But in an adaptive network, the manner in which changes to connections can alter state dynamics is much richer and has potential to exploit principles of associative memory.

5. Conclusions

This paper uses the Hopfield network as a model system to extend the emergent collective computational abilities (Hopfield, 1982) of dynamical systems. This provides a concrete model of how a self-organising dynamical system or adaptive network can transition from one scale of dynamics to another over time as a result of the changing relationships between components. The processes involved use only very simple, and fully-distributed, positive feedback mechanisms, which are relevant to other 'active linking' systems and adaptive networks, beyond neural network models (Watson, et al., 2010c). We have been investigating the effect of selective associations in various adaptive networks such as social (Davies, Watson, Mills, Buckley, Noble, 2010), genetic (Watson, et al., 2010b) and ecological networks (Lewis, 2009). However, the generative associations studied in the current paper offer a transformation in the dynamical behaviour of a system, and the ability of the system to minimise energy, that is not observed with selective associations. Specifically, whereas selective associations reduce degrees of freedom by guiding state trajectories in particular directions, generative associations 'fold-away' redundant degrees of freedom, effectively encapsulating partial configurations/partial solutions, and skipping over energy barriers/fitness valleys rather than trying to smooth them out. Computationally, *rHN-G* thus provides a distributed optimisation algorithm with similar capabilities to our formal optimisation algorithms that explicitly operate over several scales of organisation (Mills, 2010; Mills & Watson, 2011).

This suggests a neurally plausible algorithm with potential importance for cognitive problem solving, but because the model with generative associations is still fully distributed it is also relevant to non-neural adaptive networks. For example, the behaviour exhibited in the evolution of new units of selection (Watson, et al., 2009b) is closely analogous to the behaviour of generative associations in the Hopfield model shown here. In fact, it seems that the difference between selective and generative associations is closely analogous to the difference between ordinary coevolution and symbiosis that creates new units of selection, respectively (Watson, Palmius, Jackson, Mills, Powers, Buckley & Penn, in prep.). Together this suggests the exciting idea that we can view the organisation of symbiotic relationships in an ecosystem as a distributed optimisation process greater than the sum of the parts; i.e., over and above the evolution of the individual species.

In conclusion, we find that it is possible for a dynamical system to be modified by generative associations into a new system, operating at a higher level of organisation, which minimises total system energy more reliably and more completely. The Hopfield model with generative associations offers an algorithmic framework for understanding transformations and emergent collective behaviour in various (non-neural as well as neural) domains – e.g., the emergence of coalitions in social systems and of higher-level adaptive units in the major transitions in evolution.

Acknowledgements: Simon Powers, Alex Penn, Archie Chapman, David Iclanzan, Jason Noble, Seth Bullock, Jordan Pollack, Chrisantha Fernando.

References

1. Abeles M. (1982) *Local Cortical Circuits: An Electrophysiological study*. Springer, Berlin
2. Ackley, D.H., Hinton, G.E., Sejnowski, T.J. (1985) A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, 9: 147-169.
3. Angeline, P.J & Pollack, J.B. (1992) The evolutionary induction of subroutines. *In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 236-241. Lawrence Erlbaum
4. Bourgine, P. & Varela, F.J. (1992). Toward a Practice of Autonomous Systems. In Varela, F.J. & Bourgine, P. (Eds.) *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, (pp. xi-xvii). Cambridge, Mass.: MIT Press
5. Chapman, A. C. (2009) *Control of Large Distributed Systems using Games with Pure Strategy Nash Equilibria.* PhD thesis, University of Southampton
6. Chapman, A.C. Rogers, A. & Jennings, N.R. & Leslie, D.S. (2010) A Unifying Framework for Iterative Approximate Best Response Algorithms for Distributed Constraint Optimisation Problems. *The Knowledge Engineering Review* (in press).
7. Chase, W. G. & Simon, H. A. (1973) "Perception in chess". *Cognitive Psychology* 4, 55-81.
8. 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.
9. 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. *Artificial Life,* (in press).
10. Draghi, J.A., Parsons, T.L., Wagner, G.P. & Plotkin, J.B. (2010) Mutational robustness can facilitate adaptation, *Nature* 463, 353-355.
11. Fernando, C., Goldstein, R. & Szathmáry, E. (2010) The Neuronal Replicator Hypothesis. *Neural Computation*, Vol. 22, No. 11, Pages 2809-2857.
12. Fernando, C. & Szathmáry, E. (2010) Natural Selection in the Brain. *Chapter 20. Towards a Theory of Thinking Vol 2.* Springer/Parmenides book series.
13. Fontanari, J.F. (1990) Generalization in a Hopfield network. *Journal de Physique*, 51, 2421-2430.
14. Forrest, S. & Mitchell, M., (1993) "What makes a problem hard for a Genetic Algorithm? Some anomalous results and their explanation" *Machine Learning* 13, pp.285-319
15. Fuller, R.B. (1975) *SYNERGETICS: Explorations in the Geometry of Thinking*. Macmillan Publishing Co. Inc.
16. Gardner, E. (1988) The space of interactions in neural network models. *J. Phys. A: Math. Gen.* 21. 257.
17. Goetz, P., & Walters, D. (1997) The Dynamics of Recurrent Behavior Networks Adaptive Behavior 6: 247-283
18. Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, MA.
19. Gross, T. & Sayama, H. (2009) *Adaptive Networks. Theory, Models and Applications*. Series: Understanding Complex Systems. Springer-Verlag: Berlin.
20. Grossberg, S. (1978). A theory of human memory: self-organization and performance of sensory-motor codes, maps, and plans. *Progress in Theoretical Biology*, Vol. 5, pp. 233-374. Academic Press.
21. Haken, H. (1983) *Synergetics, an Introduction: Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology*, 3rd rev. enl. ed. New York: Springer-Verlag.

22. Hart & Mas-Colell (2000) A Simple Adaptive Procedure Leading to Correlated Equilibrium, *Econometrica*, 68. 1127-50

23. Hebb, D.O. (1949) *The organization of behaviour*. New York: Wiley.

24. Hinton, G.E. & Nowlan, S.J. (1987) How Learning Can Guide Evolution. *Complex Systems, 1*, 495-502

25. 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.

26. Hinton, G.E., & Sejnowski, T.J. (1985) Learning in Boltzmann Machines, *Cognitiva*, 85, Paris, France.

27. Holland, J.H. (2000) “Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions”, *Evolutionary Computation 8*, 373

28. Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *PNAS USA*, 79 (8) 2554-2558.

29. Hopfield, J.J. & Tank, D.W. (1986) Computing with neural circuits: A model. *Science* 233: 625-633

30. Hopfield, J.J., & Tank, D.W. (1985) 'Neural' computation of decisions in optimization problems. *Biol Cybern* 52:141-152.

31. Iclanzan, D., & Dumitrescu, D. (2007) Overcoming hierarchical difficulty by hillclimbing the building block structure. *GECCO*, pp. 1256–1263.

32. Izquierdo, E.J. & Fernando, C. (2008) The Evolution of Evolvability in Gene Transcription Networks. *Proceedings of Alife XI,* 265-273.

33. 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*.*

34. Jones, A.G., Arnold, S.J. & Bürger, R. (2007) The mutation matrix and the evolution of evolvability. *Evolution* 61: 727-745.

35. Kirchner, M. & Gerhart, J. (1998). Evolvability. *Proc. Natl. Acad. Sci. USA*, 95:8420–8427

36. Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*. 220 (4598): 671–680.

37. Koza, J.R. (1994) *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge MA.

38. Kryzhanovsky, B. & Kryzhanovsky, V. (2008) Binary Optimization: On the Probability of a Local Minimum Detection in Random Search. *ICAISC 2008*: 89-100.

39. Lenaerts, T., Chu, D., & Watson, R. (2005). Dynamical Hierachies. *Artificial Life*, 11, 403–405

40. Leslie, D.S. & Collins, E.J. (2006) Generalised weakened fictitious play, *Games and Economic Behavior, 56: 2*, pp 285-298

41. Lewis, M. (2009) *An Investigation Into The Evolution Of Relationships Between Species In An Ecosystem*. MSc Dissertation, ECS, University of Southampton.

42. Siegal, M.L. & Bergman, A. (2002) Waddington's canalization revisited: Developmental stability and evolution, Proceedings of the National Academy of Sciences of the United States of America 99: 10528–10532.

43. Maynard Smith, J. & Szathmary, E. (1995) *Major Transitions in Evolution*. W. H. Freeman

44. McClelland, J.L., Rumelhart D.E., & the PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations*. Cambridge, MA*:* MIT Press.

45. Michie, D. (1968) "Memo Functions and Machine Learning," *Nature*, No. 218, pp. 19-22.

46. Miller, G. A. (1956) "The magic number seven, plus or minus two: Some limits on our capacity for processing information". *Psychological Review* 63, 81-97
47. Mills, R & Watson, R. A. (2011 sub.) *Multi-scale search, modular variation, and adaptive neighbourhoods*. (submitted)
48. Mills, R. (2010) *How Micro-Evolution Can Guide Macro-Evolution: Multi-Scale Search via Evolved Modular Variation.* PhD thesis, ECS, University of Southampton.
49. Mills, R. & Watson, R. A. (2005) Genetic Assimilation and Canalisation in the Baldwin Effect. *Proceedings of 8th European Conference on Artificial Life (ECAL 2005), LNCS.* pp. 353-362
50. Mills, R., & Watson, R.A. (2009) Symbiosis Enables the Evolution of Rare Complexes in Structured Environments. *Proceedings of European Conference on Artificial Life 2009. pp.110-117*.
51. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review* 45 (2), 167-256.
52. Newman, M. E. J., Barabasi, A. L. & Watts, D. J. (Eds.) (2006). *The Structure and Dynamics of Networks*. Princeton University Press.
53. O'Reilly, R.C., & Munakata, Y. (2000) *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*, (Cambridge, MA: MIT Press).
54. Pacheco, J.M., Traulsen, A. & Nowak, M.A. (2006) Coevolution of Strategy and Structure in Complex Networks with Dynamical Linking. *Phys. Rev. Lett. 97*, 258103.
55. Paine, R.W. & Tani, J. (2005) How Hierarchical Control Self-organizes in Artificial Adaptive Systems *Adaptive Behavior* 13: 211-225
56. 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.
57. Pelikan, M. (2002) *Bayesian Optimization Algorithm: From Single Level to Hierarchy*, Ph.D. Dissertation, Dept. of Computer Science at the University of Illinois at Urbana-Champaign.
58. Pelikan, M., Goldberg, DE. & Lobo, F. (2002) *A Survey of Optimization by Building Probabilistic Models*. *Computational Optimization and Applications*, 21, 5–20.
59. 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*.
60. 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.
61. Powers, S.T., Penn, A.S. & Watson, R.A. (2011) The Concurrent Evolution of Cooperation and the Population Structures that Support it. *Evolution*, in press.
62. Powers, ST. (2010) *Social Niche Construction: Evolutionary Explanations for Cooperative Group Formation*, PhD dissertation, ECS, University of Southampton.
63. Riegler, A. (2008) Natural or Internal Selection? The Case of Canalization in Complex Evolutionary Systems. *Artificial Life*, Vol. 14, No. 3: 345–362
64. Rosenbloom, P. S., & Newell, A. (1986) The chunking of goal hierarchies: A generalized model of practice. In *Machine Learning: An Artificial Intell!gence Approach, Volume II*, R. S. Michalski, J. G. Carbonell, & T. M. Mitchell, Eds., Morgan Kaufmann Publishers, Inc., Los Altos, CA,.
65. Santos, F.C., Pacheco, J.M. & Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. *PLoS Comput Biol, 2(10),* e140: 1284–1291
66. Simon, H.A. (1969) *The Sciences of the Artificial*. (MIT Press, Cambridge, Mass).
67. Strogatz, S.H. (1994) *Nonlinear Dynamics and Chaos*. (Addison-Wesley, Reading, MA.)

68. Tononi, G. & Sporns, O. (2003) Measuring information integration, *BMC Neuroscience*, 4:31

69. Toussaint, M. & von Seelen, W. (2007) Complex adaptation and system structure, *BioSystems 90*: 769–782

70. Tsirukis, A.G., Reklaitis, G.V. & Tenorio, M.F. (1989) Nonlinear Optimization Using Generalized Hopfield Networks. *Neural Computation 1*, 511-521.

71. Tsutsumi, K. (2003) Relaxing in a warped space: an effect due to the cooperation of static and dynamical neurons. *Proceedings of the International Joint Conference on Neural Networks*. pp. 897 - 901 vol.2

72. Van Segbroeck, S., Santos, F.C., Pacheco, J.M. & Lenaerts, T. (2010) Coevolution of Cooperation, Response to Adverse Social Ties and Network Structure. *Games*, *1*(3), 317-337

73. Wagner G.P., Booth, G. & Bagheri-Chaichian, H. (1997) A population genetic theory of canalization. *Evolution* 51: 329-347.

74. Watson R.A., Buckley, C.L. & Mills, R. (2009a) *The Effect of Hebbian Learning on Optimisation in Hopfield Networks.* Technical Report, ECS, University of Southampton.

75. Watson R.A., (2006) *Compositional Evolution: The Impact of Sex, Symbiosis, and Modularity on the Gradualist Framework of Evolution*. MIT Press.

76. Watson R.A., Palmius, N., Jackson, A., Mills, R., Powers, S.T., Buckley, C.L., & Penn, A.S. (in prep) *The Origin of Group Selection via Individual Selection*.

77. Watson, R.A. & Jansen, T. (2007) A Building-Block Royal Road Where Crossover is Provably Essential. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)* . pp. 1452-1459

78. Watson, R.A., Buckley, C. L. and Mills, R. (2010a) Optimisation in 'Self-modelling' Complex Adaptive Systems. *Complexity*. (in press)

79. Watson, R.A., Buckley, C. L., Mills, R., Davies, A.P. (2010b) Associative memory in gene regulation networks, *ALIFE XII*, 194-202.

80. Watson, R.A., Mills, R. & Buckley, C. L. (2010c) Global Adaptation in Networks of Selfish Components: Emergent Associative Memory at the System Scale. *Artificial Life* (accepted).

81. Watson, R.A., Weinreich, D.M. & Wakeley, J. (2011) Genome Structure and the Benefit of Sex. *Evolution* (in press)

82. Watson, R.A. & Pollack, J.B. (2005) Modular Interdependency in Complex Dynamical Systems. *Artificial Life,* 11 (4). pp. 445-457.

83. Watson, R.A., Palmius, N., Mills, R., Powers, S.T., & Penn, A.S. (2009b) Can Selfish Symbioses Effect Higher-level Selection? G. Kampis, I. Karsai, and E. Szathmáry (Eds.): *ECAL 2009, Part II, LNCS 5778*, pp. 27–36, 2011. Springer-Verlag Berlin Heidelberg.

84. Watson, RA. & Pollack, JB. (2003) Symbiotic composition in evolutionary transitions, *BioSystems 69* pp.187-209.

85. Winston, P.H. (1992) *Artificial Intelligence* (3rd edition). Addison Wesley.

86. Wolpert, D.H. & Macready, W.G. (1997), No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation* 1, 67.

87. Young, HP. (2001) *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*, Princeton.

88. Yu, T.-L., Goldberg, D.E., Yassine, A., & Chen, Y.-P., (2003) A genetic algorithm design inspired by organization theory: a pilot study of a dependency structure matrix driven genetic algorithm. *Procs. Genetic and Evolutionary Computation Conference (GECCO 2003)*, LNCS 2724.

About the authors.

Richard Watson was awarded his Ph.D. degree in Computer Science from Brandeis University (USA) in 2002. His background in computational models of evolution was complemented by a research fellowship at Harvard University in the Dept. of Organismic and Evolutionary Biology. Since 2004 he has been a Senior Lecturer in the Natural Systems group, part of the School of Electronics and Computer Science at the University of Southampton (UK). In 2006 he was a recipient of the IEEE international award "Ten to Watch in Artificial Intelligence" as part of the IEEE's celebration of 50 years of AI. He is presently Vice President of the International Society for Artificial Life. His publications span topics including artificial life, robotics, evolutionary modelling, population genetics, evolutionary computation, and computational biology; and he is author of "*Compositional Evolution: The Impact of Sex, Symbiosis, and Modularity on the Gradualist Framework of Evolution*" with MIT Press. His current research focuses on understanding the algorithmic principles of natural processes that create adaptive biocomplexity. Address: School of Electronics and Computer Science, University of Southampton, SO17 1BJ, U.K.. E-mail: raw@ecs.soton.ac.uk

Rob Mills is a postdoctoral research fellow in the Natural Systems group, part of the School of Electronics and Computer Science at the University of Southampton. His research interests, bridging the domains of evolutionary computing and evolutionary biology, are focused around understanding the mechanisms that lead to the modular organisation of interactions in evolving systems, as well as the adaptive consequences arising from such organisation. Other interests include the evolution of evolvability, innovation, and multi-scale processes; and the interaction between learning and evolution. Address: School of Electronics and Computer Science, University of Southampton, SO17 1BJ, U.K.. E-mail: rob.mills@soton.ac.uk

Christopher L. Buckley is a research fellow in the Centre for Computational Neuroscience and Robotics at the University of Sussex. His research involves using the tools and techniques of non-linear dynamical system theory to understand the functions of the nervous system. His current focus is on invertebrate sensory (moth antennal lobe) and motor (snail feeding CPG) circuits. He also has an interest in non-linear dynamical phenomena in biological systems more generally. Address: School of Informatics, University of Sussex, Sussex House, Brighton, BN1 9RH, U.K.. E-mail: C.L.Buckley@sussex.ac.uk

Transformations in the Scale of Behaviour and the Global Optimisation of Constraints in Adaptive Networks - Supplementary on-line material

Richard A. Watson, Rob Mills and C. L. Buckley

The computational cost of model-building

Our model illustrates how a distributed learning mechanism can be used to identify and exploit problem structure without *a priori* domain knowledge. One might reasonably question, however, how the cost of inducing a model (that subsequently makes solving the problem easy) compares with the cost of 'brute forcing' the solution without a model. Here we show formally that it is possible in some problems to learn the model and solve the problem in time that is a polynomial function of the system size whereas solving the problem without the model takes exponential time.

We illustrate this by considering p close to zero (p=ε), in the modular problem with idealised inter-module constraints (i.e. all inter-module constraints are positive) (Mills & Watson 2011 sub., Mills 2010, Watson, Buckley & Mills 2009a). This makes it a consistent problem such that a single-run of a hill-climber *in the underlying low-dimensional problem* finds the global optimum, but the extreme imbalance of internal and external weights means that the basins of attraction for each local optimum in the high-dimensional problem are almost equal, revealing no inter-module correlations (see Fig. S1). In other words, in the full problem, a single-run of a hill-climber is very unlikely to find the global optimum (no more likely to find the global optimum than any other local optimum).

The time to find a *local* optimum in this problem is polynomial in the size of the system and equal for *rHN-0* and *rHN-G*, so we need only consider the number of relaxations (rather than individual time-steps) required to find the global minimum in these calculations. Firstly, the expected number of relaxations for *rHN-0* to solve this problem is on average $T_{rHN\text{-}0}$ = $2^n/2$, where n is the number of modules or blocks (c.f. the proportion of the search space that is in the basin of attraction for the global optimum). Whenever n is a linear function of N, rather than a constant (Watson & Jansen 2007), the expected time for *rHN-0* to solve this problem is therefore an exponential function of the problem size.

In contrast, if it already had the information that all block-diagonal weights m_{ij}^{B} need to be high, and all off-diagonal weights $m_{ij}^{B'}$ need to be low, then *rHN-G* could solve this problem easily. In the limit where m_{ij}^{B}=1 and $m_{ij}^{B'}$=0, *rHN-G* will find the global minimum in every relaxation in polynomial time. More generally, suppose that when *rHN-G* is carrying out a macro-variation it has a probability of 1 of including all variables within the module of the focal variable and a finite probability (not a function of N) of making a mistake and also including a variable from another module. If it makes such a mistake it may not be able to find the global optimum in this relaxation, but so long as the probability of a mistake is sufficiently small, then a polynomial number of relaxations will be sufficient to complete a relaxation with no mistakes and find the global optimum. We therefore calculate the probability of learning weights that are expected to produce less than one mistake per application of g.

State configurations that are locally optimal in *MC* will always show within-block variables having states that agree. Thus every training sample increases the weights within blocks, m_{ij}^{B}. In contrast, pairs of states in different blocks will sometimes agree and sometimes disagree depending on the particular local optima sampled. When p=ε they disagree or agree with approximately equal probability. This is what makes p=ε difficult for *rHN-S*, because *rHN-S* needs to learn $m_{ij}^{B'}$=1 in order to be successful. But *rHN-G* does not need to correctly learn inter-module correlations – that is, so long as the intra-module correlations are learned correctly *rHN-G* will be able to make correct macro-variations that effectively jump from one local optimum to another. Nonetheless, although *rHN-G* need not learn correct inter-module correlations it must avoid learning incorrect inter-module correlations. So, if $m_{ij}^{B'}$ values increase as often as they decrease on average this means that their expected value is 0 as desired. However, if these weights are learned from a small number of samples then they may be appreciably non-zero and cause mistakes in the macro-variations, as described above (see Mills (2010) for related analysis). Thus:

- The weights within blocks, m_{ij}^{B}, increase linearly with T – so we choose δ=1/T such that they will have the value 1 on the T^{th} epoch.
- All $m_{ij}^{B'}$ values will undergo a random walk – with equal probability of increasing or decreasing in each epoch. The variance of the walk is lower if we take many small steps rather than a few big steps.
- We wish to find the minimum number of relaxations, T, such that $m_{ij}^{B'}$ values will produce on average less than one error in a subsequent relaxation. Thus we want all $m_{ij}^{B'}$ values to be sufficiently small such that, conservatively, none of them imposes a state correlation between variables in different blocks.

Formally, let the external weights $m_{ij}^{B'}$ increment or decrement by Δm with probability ½ every time-step. It is easy to show that after T time-steps the expectation of $m_{ij}^{B'}$ will be zero, $E[m_{ij}^{B'}]=0$, and the variance will be $\sigma^2(m_{ij}^{B'})=T\Delta m^2$. Given $\Delta m=\frac{1}{T}$, the variance will simply be:

$$\sigma^2(m_{ij}^{B'})=\frac{1}{T} \quad . \qquad \text{Eq. 9}$$

For *rHN-G* on *MC* the expected value of the probability of making an erroneous association is $\left|E[m_{ij}^{B'}]\right|$ (per variable, per application of the variation operator *g*). Consequently, the expected number of erroneous associations made if there are *Q* external weights is $Q\tau\left|E[m_{ij}^{B'}]\right|$. As $T\rightarrow\infty$ the mean number of wrongly induced states will converge on the expectation and therefore tend to zero. However, for finite *T*, a pessimistic estimate[1] of the number of wrongly induced states, λ, is given by

$$\lambda = Q\tau\left|E\left[m_{ij}^{B'}\right]+2\sigma(m_{ij}^{B'})\right| . \qquad \text{Eq. 10}$$

From Eq. 9 and Eq. 10 we can show that with high likelihood

$$\lambda < Q\tau\left|0+\frac{2}{\sqrt{T}}\right| .$$

Since $\sqrt{T}$ must be non-negative we can drop the absolute function.-Consequently, the number of time steps needed in order be practically confident that less than λ states are wrongly induced is:

$$T > \frac{4Q^2\tau^2}{\lambda^2} .$$

To be practically confident that less than *one* state is wrongly induced then

$$T > 4Q^2\tau^2 .$$

In conclusion, since *Q* and *τ* are each only a polynomial function of *N* ($Q<N^2$ and, in our experiments, τ =10*N*), the number of relaxations, *T*, required by *rHN-G* to learn weights that ensure that almost no mistakes are made is also a polynomial function of *N*. (In practical terms *T* need not be as large as $4Q^2\tau^2$ since many errors can be tolerated without precluding success). Hence *rHN-G* finds

[1] $m_{ij}^{B'}$ is normally distributed and hence 96% of points lie within two standard deviations from the mean.

the global optimum in polynomial time. Whereas for *rHN-0* (and *rHN-S*), the expected time to find the global optimum is exponential in the system size.

Fig. S1 shows a numerical study of the sensitivity of *rHN-0*, *rHN-S* and a variant of *rHN-G* (Mills 2010) to the strength of inter-module connections, *p*. This study shows that there is a significant region of *p* where the problem is difficult for *rHN-0* and that *rHN-S* and *rHN-G* offer an advantage in this region. The remarkable finding of this study is that *rHN-G* is essentially insensitive to *p*, meaning that, unlike *rHN-S*, its ability to resolve inter-module constraints is not dependent on their strength.

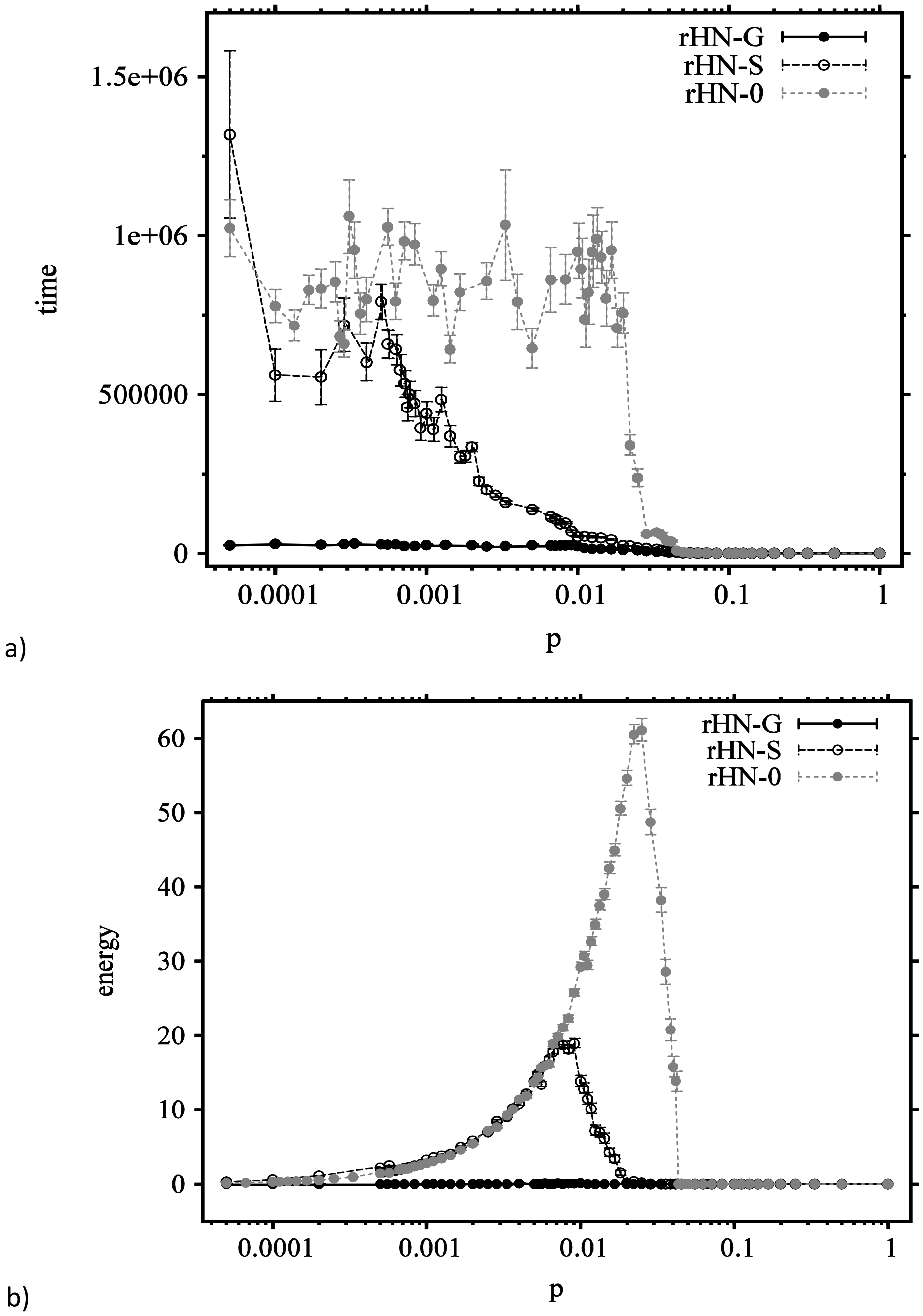

Fig. S1 The sensitivity of *rHN-0*, *rHN-S* and a variant of *rHN-G* to the strength of inter-module connections, *p* (see Mills (2010) pp.128-149 for details) in a simplified *MC* problem (*N*=100, *k*=5); a) number of time-steps to find the global optimum, b) difference in energy between the best-found configuration after 50,000 time-steps and the global optimum. Showing that, although there is a considerable range of *p* where *rHN-S* outperforms *rHN-0*, *rHN-G* outperforms *rHN-S* and *rHN-0* more broadly and is, in fact, almost insensitive to *p* - finding the global optimum very quickly in all cases.

For references see main text.