Generalized Hopfield Networks for Constrained Optimization

Jan van den Berg (Email: jvandenberg@few.eur.nl)

Abstract

A twofold generalization of the classical continuous Hopfield neural network for modelling constrained optimization problems is proposed. On the one hand, non-quadratic cost functions are admitted corresponding to non-linear output summation functions in the neurons. On the other hand it is shown under which conditions various (new) types of constraints can be incorporated directly. The stability properties of several relaxation schemes are shown. If a direct incorporation of the constraints appears to be impossible, the Hopfield-Lagrange model can be applied, the stability properties of which are analyzed as well. Another good way to deal with constraints is by means of dynamic penalty terms, using mean field annealing in order to end up in a feasible solution. A famous example in this context is the elastic net, although it seems impossible - contrary to what is suggested in the literature - to derive the architecture of this network from a constrained Hopfield model. Furthermore, a non-equidistant elastic net is proposed and its stability properties are compared to those of the classical elastic network.

In addition to certain simulation results as known from the literature, most theoretical statements of this paper are validated with simulations of toy problems while in some cases, more sophisticated combinatorial optimization problems have been tried as well. In the final section, we discuss the possibilities of applying the various models in the area of constrained optimization. It is also demonstrated how the new ideas as inspired by the analysis of generalized continuous *Hopfield models, can be transferred to* discrete *stochastic Hopfield models. By doing so, simulating annealing can be exploited in other to improve the quality of solutions. The transfer also opens new avenues for continued theoretical research.*

1 Introduction

We start presenting a general outline of how Hopfield neural networks are used in the area of combinatorial optimization. Since the source of inspiration in this paper is supplied by the classical continuous Hopfield net, we next recall this model to mind. In the consecutive subsections, we present – in a concise overview – which variations and extensions of the original model are proposed after its appearance. All this information serves as a preparation for the rest of the paper, the outline of which is sketched in the final subsection of this introduction.

1.1 Hopfield networks, combinatorial optimization, and statistical physics

Hopfield and allied networks have been used in applications which are modelled as an 'associative memory', and in problems emanating from the field of 'combinatorial optimization' ever since their conception [14, 15]. In both types of applications, an energy or cost function is minimized, while – in case of dealing with combinatorial optimization problems – a given set of constraints should be fulfilled as well. In the latter case, the problem can formally be stated as

$$\begin{aligned} &\text{minimize} && E(x) \\ &\text{subject to :} && C_\alpha(x) = 0, \quad \alpha = 1, \ldots, m, \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)$ is the state vector or system state of the neural net, x_i representing the output neuron i, and where any $C_\alpha(x) = 0$ is a constraint. If the value of the state vector is such that $\forall \alpha : C_\alpha(x) = 0$, we say that x represents a 'valid' or 'feasible' solution.

Roughly spoken, three methods are available in order to deal with the constraints. The first and eldest one is the penalty approach where usually fixed and quadratic penalty functions are added to the original cost function. In practice, it turns out very difficult to find penalty weights that

guarantee both valid and high quality solutions. In a second approach, constraints are directly incorporated in the neural network by choosing appropriate transfer functions in the neurons. Up till now, the applicability of this method was centered in independent, symmetric linear constraints. A third way to grapple with the constraints, is combining the neural network with the Lagrange multiplier method resulting in what we call a Hopfield-Lagrange model.

All these methods can be combined with a type of 'annealing' [1] where during relaxation of the recurrent network, the 'temperature' of the system is gradually lowered in order to try not to land in a local minimum. The technique of annealing originates from an analysis of Hopfield-type networks using the theory of statistical mechanics [26]. In this paper however, we emphasize on a mathematical analysis and only refer to physical interpretations if these yield relevant additional insights. The three above-mentioned methods for resolving constrained optimization problems form part of this analysis.

1.2 The classical continuous Hopfield model

Applying the classical discrete [17] or classical continuous [18] Hopfield model, the cost or energy function to be minimized is quadratic and is expressed in the output of the neurons. In this paper the source of inspiration is the continuous model. Then, the neurons are continuous-valued and the continuous energy function $E_c(V)$ is given by

$$E_{\mathrm{c}}(V) = \underbrace{\Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i}_{E(V)} + \underbrace{\sum_i \int_0^{V_i} g^{-1}(v)dv}_{E_{\mathrm{h}}(V)} \quad (2)$$

$$= E(V) + E_{\mathrm{h}}(V) \quad . \quad (3)$$

$E(V)$ corresponds to the cost function of equation (1), where $V = (V_1, V_2, \ldots, V_n) \in [0,1]^n$ represents the state vector. $E_{\mathrm{h}}(V)$, which we call the 'Hopfield term', has a statistical mechanical interpretation based on a so-called mean field analysis of a *stochastic* Hopfield model [16, 34, 15, 14, 35, 4]. Its general effect is a displacement of the minima of $E(V)$ towards the interior of the state space [18] whose magnitude depends on the current 'temperature' in the system: the higher the temperature is, the larger is the displacement towards the interior[1]. The motion equations corresponding to (2) are

$$\dot{U}_i = \Leftrightarrow\frac{\partial E_{\mathrm{c}}(V)}{\partial V_i} = \sum_j w_{ij}V_j + I_i \Leftrightarrow U_i, \quad (5)$$

where $V_i = g(U_i)$ should hold continuously. U_i represents the (weighted) input of neuron i. After a random initialization, the network is generally not in an equilibrium state. Then, while maintaining $V_i = g(U_i)$, the input values U_i are adapted conform (5). The following theorem [18] gives conditions for which an equilibrium state will be reached:

Theorem 1 (Hopfield). *If (w_{ij}) is a symmetric matrix and if $\forall i : V_i = g(U_i)$ is a monotone increasing, differentiable function, then E_{c} is a Lyapunov function* [3, 15, 14] *for motion equations* (5).

Under the given conditions, the theorem guarantees convergence to an equilibrium state of the neural net where

$$\forall i : V_i = g(U_i) \quad \wedge \quad U_i = \sum_j w_{ij}V_j + I_i. \quad (6)$$

1.3 The penalty model

The oldest approach for solving combinatorial optimization problems using Hopfield models consists of a so-called penalty method, sometimes called the 'soft' approach [35, 29]: extra 'penalty' terms are added to the original energy function, penalizing violation of constraints. The various

[1] In case of choosing the sigmoid $V_i = 1/(1+\exp(-\beta U_i))$ as the transfer function $g(U_i)$, the part of temperature is acted by $1/\beta$ and the Hopfield term in (3) can be written as [16, 35, 4, 6, 7]

$$E_{h_s}(V) = \tfrac{1}{\beta}\sum_i (V_i \ln V_i + (1-V_i)\ln(1-V_i)). \quad (4)$$

From this, the displacement of solutions towards towards the interior is recognized easily since $E_{h_s}(V)$ has one absolute minimum, precisely in the middle of the state space where $\forall i : V_i = 0.5$.

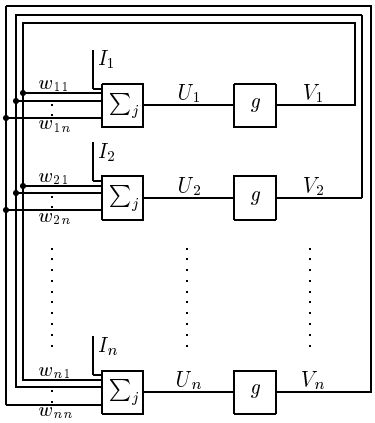

Figure 1: The original continuous Hopfield network.

penalty terms are weighted with – originally – fixed weights c_α (in case we shall speak of a *static* penalty method) and chosen in such a way that

$$\sum_{\alpha=1}^{m} c_\alpha C_\alpha(V) \text{ has a minimum value } \Leftrightarrow \ V \text{ represents a valid solution.} \tag{7}$$

In many cases, the chosen penalty terms are *quadratic* expressions. Applying a continuous Hopfield network, the original problem (1) is converted into

$$\text{minimize } \ E_{\mathrm{p}}(V) = E(V) + \sum_{\alpha=1}^{m} c_\alpha C_\alpha(V) + E_{\mathrm{h}}(V), \tag{8}$$

$E(V)$ and $E_{\mathrm{h}}(V)$ being given by (3). The corresponding updating rule is

$$\dot{U}_i = \Leftrightarrow \frac{\partial E_{\mathrm{p}}}{\partial V_i} = \sum_j w_{ij} V_j + I_i \Leftrightarrow c_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i. \tag{9}$$

Ignoring the Hopfield term for the moment (by applying a low temperature), the energy function E_{p} is a weighted sum of $m+1$ terms and a difficulty arises in determining correct weights c_α. The minimum of E_p is a *compromise* between fulfilling the constraints and minimizing the original cost function $E(V)$. Applying this penalty approach to the travelling salesman problem (TSP) [19, 40], the weights had to be determined by trial and error. For only a small low-dimensional region of the parameter space valid tours were found, especially when larger problem instances were tried[2].

1.4 A first transformation

Instead of calling $V = (V_1, V_2, \ldots, V_n)$ the state vector or system state of the neural net, we here make the generalization of calling the set $\{U, V\} = \{(U_1, U_2, \ldots, U_n), (V_1, V_2, \ldots, V_n)\}$ the system state. At the same time, we drop the condition that the input-output characteristic $V_i = g(U_i)$ should hold continuously. The following theorem appears to hold [23, 4, 6, 7].

Theorem 2. *The energy expression* (2) *can be generalized to the energy expression*

$$F_{\mathrm{g1}}(U, V) = \Leftrightarrow \tfrac{1}{2} \sum_{i,j} w_{ij} V_i V_j \Leftrightarrow \sum_i I_i V_i + \sum_i U_i V_i \Leftrightarrow \sum_i \int_0^{U_i} g(u)\mathrm{d}u. \tag{10}$$

If (w_{ij}) *is a symmetric matrix, then any stationary point of the energy* F_{g1} *coincides with an equilibrium state of the continuous Hopfield neural network. If, moreover,* F_{g1} *is bounded below, and if* $\forall i : V_i = g(U_i)$ *is a differentiable and monotone increasing function, then* F_{g1} *is a Lyapunov function for the motion equations* (5).

[2] Aside we mention that for 'constrained satisfaction problems' (by which we mean combinatorial problems without a cost function to be minimized like the n-queen problem and the 4-coloring problem), the penalty method has proven to be quite useful. See, e.g., [38].

Proof. F_{g1} can simply be derived from Hopfield's original expression (2), using partial integration. Having $V_i = g(U_i)$, we can write

$$\begin{aligned} \sum_i \int_0^{V_i} g^{-1}(v)\mathrm{d}v &= \sum_i \left[g^{-1}(v)v\right]_0^{V_i} \Leftrightarrow \sum_i \int_{g^{-1}(0)}^{U_i} v\mathrm{d}u \\ &= \sum_i U_i V_i \Leftrightarrow \sum_i \int_0^{U_i} g(u)\mathrm{d}u + c, \end{aligned} \tag{11}$$

where $c = \Leftrightarrow \sum_i \int_{g^{-1}(0)}^{0} g(u)\mathrm{d}u$ is an unimportant constant which may be neglected[3]. Substitution of (11) in (2) yields (10). By resolving

$$\forall i : \partial F_{\mathrm{g1}}/\partial U_i = 0 \ \wedge \ \partial F_{\mathrm{g1}}/\partial V_i = 0, \tag{12}$$

the set of equilibrium conditions (6) is immediately found. Taking the time derivative of F_{g1}, using the symmetry of (w_{ij}), and assuming that constantly $V_i = g(U_i)$, we find

$$\begin{aligned} \dot{F}_{\mathrm{g1}} &= \sum_i \frac{\partial F_{\mathrm{g1}}}{\partial V_i}\dot{V}_i + \sum_i \frac{\partial F_{\mathrm{g1}}}{\partial U_i}\dot{U}_i \\ &= \sum_i \big(\Leftrightarrow \sum_j w_{ij}V_j \Leftrightarrow I_i + U_i\big)\dot{V}_i + \sum_i \big(V_i \Leftrightarrow g(U_i)\big)\dot{U}_i \\ &= \Leftrightarrow \sum_i \dot{U}_i\dot{V}_i = \Leftrightarrow \sum_i (\dot{U}_i)^2 \frac{\mathrm{d}V_i}{\mathrm{d}U_i} \leq 0. \end{aligned} \tag{13}$$

Since F_{g1} is supposed to be bounded below (which, generally[4], is the case [4]), applying the motion equations (5) constantly decreases F_{g1} until $\forall i : \dot{U}_i = 0$. □

The given proof induces another updating scheme (also referred to in [15]):

Theorem 3. *If the matrix (w_{ij}) is symmetric and positive definite and if F_{g1} is bounded below, then F_{g1} is a Lyapunov function for the motion equations*

$$\dot{V}_i = g(U_i) \Leftrightarrow V_i, \text{ where } U_i = \sum_j w_{ij}V_j + I_i. \tag{14}$$

Proof. The proof again considers the time derivative of F_{g1}. If (w_{ij}) is symmetric and positive definite and if constantly $U_i = \sum_j w_{ij}V_j$, then

$$\dot{F}_{\mathrm{g1}} = \Leftrightarrow \sum_i \dot{V}_i\dot{U}_i = \Leftrightarrow \sum_i \dot{V}_i \sum_j \frac{\partial U_i}{\partial V_j}\dot{V}_j = \Leftrightarrow \sum_i \dot{V}_i \sum_j w_{ij}\dot{V}_j \leq 0. \tag{15}$$

Thus, updating conform the motion equations (14) decreases the corresponding Lyapunov function until, finally, $\forall i : \dot{V}_i = 0$. □

It is interesting to see that the conditions for which the updating rules (5) and (14) guarantee stability are different. In the first case, stability depends on the symmetry of (w_{ij}) and on the monotonicity of the transfer function chosen. In the second case, stability depends on again the symmetry of (w_{ij}), but furthermore on the positive definiteness of this matrix and therefore on the general structure of the cost function $E(V)$ involved.

Finishing this section we observe that if the sigmoid (see footnote 1) is chosen as the transfer function, equation (10) reduces to

$$F_{\mathrm{g2}}(U,V) = \Leftrightarrow \tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow \sum_i I_iV_i + \sum_i U_iV_i \Leftrightarrow \tfrac{1}{\beta}\sum_i \ln(1+\exp(\beta U_i)). \tag{16}$$

This expression (or similar ones) can also be derived using a statistical mechanical analysis of Hopfield neural networks using a mean field analysis [14, 28, 35, 4, 6, 7].

[3] It is not difficult to see that $g(0) = 0 \Rightarrow c = 0$.

[4] This issue is related to the displacement of solutions as discussed in subsection 1.2.

1.5 Incorporating symmetric linear constraints

In order to deal with the constraints (1), one can try to incorporate them directly in the neural network. A classical example relates to an attempt to solve the TSP [29, 10]. The transfer function chosen was

$$V_i = g_i(U) = \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)}, \tag{17}$$

implying the linear constraint $\sum_i V_i = 1$. By using this transfer function, the original Hopfield network is generalized to a model where $V_i = g_i(U)$ is a function of $U_1, U_2, \ldots, U_n$ and not of U_i alone. As in the classical Hopfield model, an energy expression of type (3) can be derived [20, 4, 6, 7] being

$$E_{\rm cc}(V) = \Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i + \tfrac{1}{\beta}\sum_i V_i \ln V_i, \tag{18}$$

where, again, $1/\beta$ plays the part of temperature. At high temperatures, the constrained local minima of the cost function $E(V) = \Leftrightarrow\frac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i$ are displaced towards the point where $\forall i : V_i = 1/n$, since the Hopfield term $\frac{1}{\beta}\sum_i V_i \ln V_i$ of this model attains its minimal value there. At low temperatures however, the displacement of those local minima is negligible.

Precisely like $E_c(V)$ can be generalized to an expression (16) using a statistical mechanical analysis, so the energy $E_{\rm cc}$ can be transformed to a generalized version [29, 35, 42, 4, 6, 7]:

Theorem 4. *If (w_{ij}) is a symmetric matrix, then the energy of the continuous Hopfield network submitted to the constraint* (17)*, can be stated as*

$$F_{\rm g3}(U,V) = \Leftrightarrow\tfrac{1}{2}\sum_{i,j} w_{ij}V_iV_j \Leftrightarrow\sum_i I_iV_i + \sum_i V_iU_i \Leftrightarrow\tfrac{1}{\beta}\ln(\sum_i \exp(\beta U_i)). \tag{19}$$

The stationary points of $F_{\rm g3}$ are found at points of the state space for which

$$\forall i : V_i = \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)} \ \wedge\ U_i = \sum_j w_{ij}V_j + I_i. \tag{20}$$

If, moreover, $F_{\rm g3}$ is bounded below, if the transfer function as given in equation (17) *is used as the transfer function, and if, during updating, the Jacobian matrix $J_{\rm g} = (\partial g_i/\partial U_j)$ first becomes and then remains positive definite, then $F_{\rm g3}$ is a Lyapunov function for the motion equations* (5)*.*

Proof. For a derivation of expression (19), we refer to the above-cited literature. Resolution of the equations $\partial F_{\rm g3}/\partial U_i = 0, \partial F_{\rm g3}/\partial V_i = 0$ yields equations (20) as solutions. From the conditions as given in the theorem it follows that, in the long run,

$$\begin{aligned}
\dot{F}_{\rm g3} &= \sum_i \frac{\partial F_{\rm g3}}{\partial V_i}\dot{V}_i + \sum_i \frac{\partial F_{\rm g3}}{\partial U_i}\dot{U}_i \\
&= \sum_i (\Leftrightarrow\sum_j w_{ij}V_j \Leftrightarrow I_i + U_i)\dot{V}_i + \sum_i (V_i \Leftrightarrow \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)})\dot{U}_i \\
&= \Leftrightarrow\sum_i \dot{U}_i \sum_j \frac{\partial V_i}{\partial U_j}\dot{U}_j = \Leftrightarrow\dot{U}^T J_{\rm g}\dot{U} \leq 0.
\end{aligned} \tag{21}$$

Since $F_{\rm g3}$ is bounded below, its value decreases constantly until $\forall i : \dot{U}_i = 0$. □

Again, the stationary points of an energy expression coincide with the equilibrium conditions of a continuous Hopfield network. Whether the general condition holds that the matrix $J_{\rm g}$ will become and remain positive definite, is not easy to say. But realizing that if (17) holds and if $l \neq i$, then

$$\frac{\partial V_i}{\partial U_i} = \beta V_i(1 \Leftrightarrow V_i) > 0 \quad \text{and} \quad \frac{\partial V_i}{\partial U_l} = \Leftrightarrow\beta V_iV_l < 0, \tag{22}$$

the symmetric matrix $J_{\rm g}$ can be written as

$$\beta \begin{pmatrix} V_1(1 \Leftrightarrow V_1) & \Leftrightarrow V_1V_2 & \cdots & \Leftrightarrow V_1V_n \\ \Leftrightarrow V_2V_1 & V_2(1 \Leftrightarrow V_2) & \cdots & \Leftrightarrow V_2V_n \\ \vdots & \vdots & & \vdots \\ \Leftrightarrow V_nV_1 & \Leftrightarrow V_nV_2 & \cdots & V_n(1 \Leftrightarrow V_n) \end{pmatrix}. \tag{23}$$

Thus, all diagonal elements of J_g are positive, while all non-diagonal elements are negative. Knowing that $\sum_i V_i = 1$, we argue that, generally, for large n, the inequality

$$\forall i, \forall j, \forall k : V_i V_j << V_k (1 \Leftrightarrow V_k) \tag{24}$$

holds, although this statement is not always true. Nevertheless, it is not unreasonable to expect that in many cases, the matrix J_g is dominated by the diagonal elements, making it positive definite[5]. For these reasons, it is *conjectured* that the motion equations (5) turn out to be stable in many practical applications. As in the unconstrained case, a complementary set of motion equations can be applied.

Theorem 5. *If the matrix (w_{ij}) is symmetric and positive definite, then F_{g3} is a Lyapunov function for the motion equations*

$$\dot{V}_i = \frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)} \Leftrightarrow V_i, \quad \text{where} \quad U_i = \sum_j w_{ij} V_j + I_i. \tag{25}$$

The proof is similar to the proof of theorem 3.

1.6 Other modifications of the classical continuous Hopfield model

In trying to be complete in our overview, we here mention some other Hopfield network modifications. Various still other modifications not discussed here, will be brought out at appropriate places in the rest of the paper.

1.6.1 Especial transfer functions

In [38] many applications of Hopfield networks in the area of constrained optimizations are discussed. At the same time, other types of neurons are introduced. In order to deal with undesirable oscillations, the 'hysteresis McCulloch-Pitts neuron' is proposed defined by

$$V_i = \begin{cases} 1 & \text{if } U_i > u \\ 0 & \text{if } U_i < l \\ \text{'unchanged'} & \text{otherwise,} \end{cases} \tag{26}$$

where u and l ($u > l$) are certain constants. In order to deal with constraints, the 'maximum neuron' is suggested defined by

$$V_i = \begin{cases} 1 & \text{if } U_i = \max_i(U_1, \ldots, U_n) \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

An interesting suggestion for dealing with mutually dependent constraints is done in [33] where, during relaxation of the neural network, the constraints are kept fulfilled constantly by normalizing both rows and columns of the permutation matrix conform the principle of equation (17) using the (iterative) 'Sinkhorn' algorithm.

1.6.2 Mean field annealing and chaos

As we have seen, relaxation of the Hopfield network towards an equilibrium state can be pursued using the motion equations (5), while keeping $V_i = g_i(U)$. The precise state that will be reached depends on (a) the cost function $E(V)$, (b) the initialization of the state vector V, and (c) the Hopfield term $E_h(V)$. Here, the cost function $E(V)$ follows from the mapping of the constrained optimization problem onto the Hopfield model. In trying to land in the global minimum, several initializations of the state vector V can be tried (in fact, expert knowledge can be exploited here). Finally, an annealing scheme called 'mean field annealing' (MFA) [28, 29, 15, 14] – a deterministic version of 'simulated annealing' [1] – can be applied: during updating, the temperature $T = 1/\beta$ of the system is lowered gradually effecting the Hopfield term and it is hoped that the system will settle down itself in the global minimum.

In another attempt to obtain (semi-)optimal solutions, an 'inhibitory self-loop' is incorporated in the Hopfield network yielding a chaotic Potts spin model [20]. Also in this approach, MFA is applied.

[5] Under the given conditions, the symmetric matrix J_g has only positive eigenvalues.

1.6.3 Higher order networks

In the classical Hopfield model, the energy expression $E(V)$ is confined to quadratic functions. In several papers, attempts have been reported with models that admit cost functions of degree three and higher, called 'higher order networks'. In [22], a cubic expression of $E(V)$ is proposed and applied on the travelling salesman problem. Although stability could not be guaranteed, the superiority of the new model over the classical one was demonstrated experimentally.

In [37], fourth order Hopfield networks are proposed in order to cope with the numerous constraints. Basically, a penalty approach was adopted here. Using gradient descent, the corresponding motion equations were derived. It is suggested that the usually occurring 'spurious states' [15, 14] are not present using this approach.

In [12], energy expressions of any order are introduced and conditions (concerning the symmetric properties of the weights) are formulated that guarantee stability of the updating scheme (5). Simulation results demonstrate that the higher models proposed outperform the classical Hopfield network by the empirical observation that the 'basin of attraction' [15] of the global optimal solution is enlarged as the order of the network is increased.

The idea of applying non-quadratic cost functions also emerged in other methods of neural net optimization like in the so-called 'multiscale optimization' technique [24]. In addition to the original fine-scale problem, an approximated coarse-scale problem is defined. Then, a general scheme is applied for alternating between the consecutive relaxation steps in both neural nets together with a two-way information flow between the nets.

Finally in [41], arbitrary energy functions are admitted in a hybrid 'Lagrange and Transformation' approach[6]. In addition, alternative formulations of the Hopfield term (called 'barrier functions') are proposed intended to avoid local minima in an efficient way.

1.6.4 Iterative methods

Instead of using the motion equations (5), various people suggested iterative methods in order to speed up relaxation of a given Hopfield net. Although the stability properties of the iterative updating schemes are usually different, these schemes have indeed proven to converge much faster in many cases. Quite common rules in the literature are of the type

$$V_i^{\text{new}} = g_i(U^{\text{old}}) = g_i(\sum_j w_{ij} V_j^{\text{old}} + I_i). \tag{28}$$

In the afore-mentioned paper of the previous subsection [41], a more complicated iterative scheme is proposed where besides the neural input and output values U_j and V_j, Lagrange multipliers are involved in the updating scheme. We further note that the Sinkhorn algorithm as mentioned is section 1.6.1 is also an iterative algorithm that is embedded in the usual updating schemes.

1.7 Outline of the rest of the paper

In the following section, we introduce a new and very general framework for describing continuous Hopfield network. Here, non-quadratic cost functions are admitted as well as constraints of all kind. We also present the result of certain simulations. In section 3, we shall dwell on the ways in which asymmetric linear and certain non-linear constraints can actually be built-in. After this, we analyze the Hopfield-Lagrange model and, in a separate section, we consider elastic networks relating them to the models discussed so-far. In the final section, we make our conclusions and do suggestions for further research. Especially we shall pay attention to the possibilities of transferring the ideas as emerged in the context of continuous Hopfield nets towards the area of discrete stochastic Hopfield models.

2 The generalized continuous Hopfield network

In the first part of this section, we introduce our generalization of the classical continuous Hopfield model, including some theorems. In the second part, we discuss some consequences of these theorems among which the phenomenon that the new model does incorporate the classical Hopfield model and many of its modifications, as a special case. Finally we present the outcomes of several simulations.

[6] In this paper, we shall deal with Lagrange multipliers in section 4.

2.1 Some theorems

It is remarkable that the motion equations (5) of the continuous unconstrained model may still be applied using the constrained model. This poses the question whether those motion equations can still be applied if an arbitrary transfer function of the form $V_i = g_i(U) = g_i(U_1, U_2, \ldots, U_n)$ is used. At the same time, we change $U_i = \sum_j w_{ij} V_j + I_i$ into a generalized, mostly non-linear 'summation function' of type $U_i = h_i(V)$, where an external input I_i is still admitted. By this, the so-called *generalized continuous Hopfield network* is created, a visualization of which is given in figure 2. It is quite important to understand the equilibrium and stability properties of this generalized network. The first of the following two theorems considers the issue concerning equilibrium.

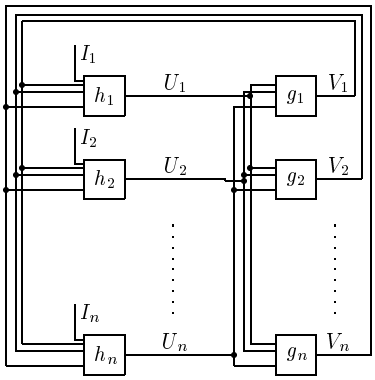

Figure 2: The generalized continuous Hopfield network.

Theorem 6. *Let $G(U) = G(U_1, U_2, \ldots, U_n)$ be a function for which*

$$\forall i : \frac{\partial G(U)}{\partial U_i} = g_i(U), \tag{29}$$

and let, in the same way, $H(V) = H(V_1, V_2, \ldots, V_n)$ be a function for which

$$\forall i : \frac{\partial H(V)}{\partial V_i} = h_i(V), \tag{30}$$

then any stationary point of the energy

$$F_{\mathrm{g}}(U, V) = \Leftrightarrow H(V) + \sum_i U_i V_i \Leftrightarrow G(U) \tag{31}$$

coincides with an equilibrium state of the generalized continuous Hopfield network, defined by

$$\forall i : V_i = g_i(U) \quad \wedge \quad U_i = h_i(V). \tag{32}$$

Proof. Resolving

$$\forall i : \partial F_{\mathrm{g}}/\partial U_i = 0 \ \wedge \ \partial F_{\mathrm{g}}/\partial V_i = 0, \tag{33}$$

the set of equilibrium conditions (32) is found. □

The following theorem considers the above-mentioned second issue concerning stability.

Theorem 7. *If $F_{\mathrm{g}}(U, V)$ is bounded below, then the next statements hold:*
(a) *If, during updating, the Jacobian matrix $J_{\mathrm{g}} = (\partial g_i/\partial U_j)$ first is (or becomes) and then remains positive definite, then the energy function F_{g} is a Lyapunov function for the motion equations*

$$\dot{U}_i = \Leftrightarrow \frac{\partial F_g}{\partial V_i} = h_i(V) \Leftrightarrow U_i, \text{ where } V_i = g_i(U). \tag{34}$$

(b) *If, during updating, the Jacobian matrix $J_{\mathrm{h}} = (\partial h_i/\partial V_j)$ first is (or becomes) and then remains positive definite, then the energy function F_{g} is a Lyapunov function for the motion equations*

$$\dot{V}_i = \Leftrightarrow \frac{\partial F_g}{\partial U_i} = g_i(U) \Leftrightarrow V_i, \text{ where } U_i = h_i(V). \tag{35}$$

(c) *If, during updating, the Jacobian matrices* J_{g} *and* J_{h} *first are (or become) and then remain positive definite, then the energy function* F_{g} *is a Lyapunov function for the motion equations*

$$\dot{U}_i = \Leftrightarrow\frac{\partial F_g}{\partial V_i} = h_i(V) \Leftrightarrow U_i \quad \wedge \quad \dot{V}_i = \Leftrightarrow\frac{\partial F_g}{\partial U_i} = g_i(U) \Leftrightarrow V_i. \tag{36}$$

Proof. Assuming that the conditions as mentioned in (c) hold, we obtain

$$\begin{aligned} \dot{F}_{\mathrm{g}} &= \sum_i (\Leftrightarrow h_i(V) + U_i) \sum_j \frac{\partial V_i}{\partial U_j}\dot{U}_j + \sum_i (V_i \Leftrightarrow g_i(U)) \sum_j \frac{\partial U_i}{\partial V_j}\dot{V}_j \\ &= \Leftrightarrow\sum_i \dot{U}_i \sum_j \frac{\partial V_i}{\partial U_j}\dot{U}_j \Leftrightarrow\sum_i \dot{V}_i \sum_j \frac{\partial U_i}{\partial V_j}\dot{V}_j \\ &= \Leftrightarrow\dot{U}^T J_{\mathrm{g}} \dot{U} \Leftrightarrow\dot{V}^T J_{\mathrm{h}} \dot{V} \leq 0. \end{aligned} \tag{37}$$

So, the boundedness of F_{g} is sufficient to guarantee convergence to a state where $\forall i : \dot{U}_i = \dot{V}_i = 0$, implying fulfillment of the equilibrium condition

$$\forall i : U_i = h_i(V) \quad \wedge \quad V_i = g_i(U). \tag{38}$$

The proofs of the parts (a) and (b) can be performed in the same way as set out in the last part of the proof of theorem 4. □

2.2 Discussing the generalized model

2.2.1 Existing models that fit

It is not difficult to see that the classical Hopfield network of section 2 is a special case of the generalized model introduced in this section. First of all, it is easy to verify that the original quadratic cost function $E(V)$ of equation (2) fulfills the condition (30) since in that case

$$\forall i : \frac{\partial H(V)}{\partial V_i} = \frac{\partial E(V)}{\partial V_i} = \sum_j w_{ij} V_j + I_i = h_i(V). \tag{39}$$

Similarly, the function $G(U)$ corresponding to the classical Hopfield model and defined by equation (10), does fulfill the condition (29) since

$$\forall i : \frac{\partial G(U)}{\partial U_i} = \frac{\partial (\sum_i \int_0^{U_i} g(u)\mathrm{d}u)}{\partial U_i} = g(U_i) = g_i(U). \tag{40}$$

Therefore, theorem 6 holds for the original Hopfield network yielding for the stationary points of F_g the states defined by equation (32) which, in this case, coincide with the equilibrium states (6).

In a direct line of this we note that the conditions as given in theorem 1 are also sufficient to proof stability of the classical network by means of theorem 7: If the matrix (w_{ij}) is symmetric and if $V_i = g(U_i)$ is monotone increasing and differentiable, then all non-diagonal elements of J_g are zero and all diagonal elements are positive, making this matrix positive definite. So, stability of the classical model has been proven conform theorem 7(a).

It is easier still to see that the Potts glasses model (which, because of the corresponding physical phenomenon, is the usual name of the model of section 1.5) fits in the most general framework of this section: The expression (19) of F_{g3} is a special case of the expression (31) of F_g.

We finally note here that many other modifications of the classical Hopfield model as mentioned in subsection 1.6 like [12, 37], fit in the framework proposed, simply because the corresponding motion equations are derived using a gradient descent in conformity with equation (34). In some cases, even stability can be proven using theorem 7.

2.2.2 How to use the generalized model

Within the generalization introduced, much more freedom exists for configuring continuous Hopfield networks. Modelling an energy expression $H(V)$ is rather simple, since the corresponding summation functions $h_i(V)$ can be found by taking the partial derivatives conform definition (30). As mentioned before, it can also be tried to proof stability of the motion equations using theorem 7.

Building-in constraints however, is much more difficult. The transfer functions g_i should meet several requirements:

1. g_i should be chosen in conformity with condition (29).
2. The values of $g_i(U)$ should imply the validity of the constraints $C_\alpha(V) = 0$ to be incorporated, at least in the equilibrium states.
3. At 'low temperatures', the energy landscape of the cost function $E(V)$ should not be disturbed by the Hopfield term $E_h(V)$ (the expression of which strongly relates to the transfer functions g_i at hand).

The first and second requirement are self-evident. The third requirement follows from an inductive argument: it also holds for the original continuous Hopfield model as well as for the constrained Hopfield model as described in sections 1.2 and 1.5 respectively[7]. In the next subsection we shall show experimentally that there exist other transfer functions which appear to fulfill the afore-mentioned three requirements.

2.3 Simulation results concerning the generalized model

In the afore-mentioned literature on extensions of the classical Hopfield model (section 1.6), several examples can be found of the capabilities of the generalized model. E.g., the in [12] discussed higher order neural networks appear to represent a strong heuristic for solving the Ising Spin (checkerboard pattern) problem. In addition, it is argued in [37] that the use of higher order couplings between the neurons helps to avoid the usual 'spurious states' [15] of the system. In [22] it is found that especially for large problem instances the extended Hopfield model works better: this improved performance is attributed to the convergence with 'frustration' [34], a notion from the theory of statistical mechanics. But in spite of these (experimental) results, we do not have a full understanding of the behavior of the generalized Hopfield model yet. Therefore, the object of presenting the computational results of some simulations here is simply to show that the derived general theories are not falsified by these elementary tests. At the same time, these tests yield certain encouraging and startling outcomes opening new perspectives for future research.

2.3.1 A toy problem

The first problem concerns a simple test whether *non-quadratic* cost functions can be tackled using the general framework. Consider the following toy problem:

$$\text{minimize} \quad \Leftrightarrow V_1^2 V_2^3 + V_2^5 \quad \text{subject to:} \quad V_1 + V_2 = 1. \tag{41}$$

The corresponding motion equations are

$$\dot{U}_1 = 2V_1 V_2^3 \Leftrightarrow U_1, \tag{42}$$

$$\dot{U}_2 = 3V_1^2 V_2^2 \Leftrightarrow 5V_2^4 \Leftrightarrow U_2, \tag{43}$$

where the corresponding V_i's are determined using equation (17). Taking $\beta = 0.001$, the high temperature solution encountered is $V_1 = 0.5001$, $V_2 = 0.4999$. Choosing $\beta = 50$, $V_1 = 0.617$, $V_2 = 0.383$ is found, which approach the exact solution in $[0, 1]$, being $V_1 = 0.625$ and $V_2 = 0.375$.

2.3.2 The n-rook problem

The second test is done to get some idea on the differences of convergence time between updating schemes based on the penalty approach and those based on the incorporation of constraints.

We consider the n-rook problem: given an $n \times n$ chess-board the goal is to place n non-attacking rooks on the board. The problem is the same as the 'crossbar switch scheduling' problem[8]. We may map the problem on the continuous Hopfield network as follows: if V_{ij} represents whether a rook is placed on the square of the chess-board with row number i and column number j, we search for a combination of V_{ij}-values such that the following constraints are fulfilled:

$$C_1 = \sum_{i,j} \sum_{k>j} V_{ij} V_{ik} = 0, \quad C_2 = \sum_{j,i} \sum_{k>i} V_{ij} V_{kj} = 0, \quad C_3 = \tfrac{1}{2} (\sum_{i,j} V_{ij} \Leftrightarrow n)^2 = 0. \tag{44}$$

[7] From the statistical mechanical perspective the third requirement does raise a fundamental question: "Which conditions should the transfer functions $g_i(U)$ fulfill in order to guarantee that the generalized continuous Hopfield network can (still) be considered as a mean field approximation of the corresponding stochastic Hopfield net?". Up till now, the answer to this question has not been given.

[8] The crossbar switch scheduling problem has also been resolved by Takefuji [38], although he applied a network having another type of neurons.

$C_1 = 0$ implies that in any row at most one $V_{ik} \neq 0$, $C_2 = 0$ implies that in any column at most one $V_{kj} \neq 0$. $C_3 = 0$ in combination with $C_1 = C_2 = 0$ implies that precisely n rooks are placed on the board. The expressions for C_1, C_2, and C_3 can be used as penalty terms since they fulfill requirement (7).

Applying the penalty approach of subsection 1.3 while choosing $\forall \alpha : c_\alpha = 1$, convergence is present provided Δt is chosen to be small enough. E.g., in case of $n = 25$, $\beta = 1000$, $\Delta t = 0.0001$, we invariably found convergence to one of the approximately 1.55×10^{25} solutions. Taking $n = 100$ with $\Delta t = 0.00005$, the system also turns out to be stable. However, the calculation time now becomes an issue (several hours), since the neural network involved consists of 10 000 neurons, which have to be updated sequentially.

We also solved the n-rook problem by means of a constrained model having partially built-in symmetric linear constraints. By this, the space of admissible states is reduced from 2^{n^2} to $2^{n \log_2 n}$. The V_{ij} are chosen such that

$$\forall i : \sum_j V_{ij} = 1, \tag{45}$$

implying that in every row, the sum of occupied squares of the chess-board equals one. It now suffices to minimize the cost function

$$F_{\mathrm{c,nr}}(V) = c_2 C_2(V) + E_h(V). \tag{46}$$

The experimental outcomes confirm the conjecture that the constrained network behaves much better. At low temperatures ($\beta > 0.5$), the effect of noise is small and the final neural outputs are close to 0 or 1. If next the temperature is slightly increased, a rapid so-called 'phase transition' [15, 34] occurs: for $\beta = 0.3$, the solution values become almost equal conform $V_{ij} \approx 0.2500$. The convergence time is invariably only *a small fraction* of the convergence time of the pure penalty method.

The values of the neurons initially seem to change in a chaotic way: the value of the $F_{\mathrm{c,nr}}$ strongly oscillates in an unclear way. However, after a certain period, the network suddenly finds its way to a stable minimum, at the same time rapidly minimizing the value of the cost function. This outcome agrees with our conjecture on stability of the constrained model as mentioned in section 1.5.

We finally tried to resolve the n-rook problem using the Sinkhorn algorithm as mentioned in subsection 1.6.1. By this, the state of admissible states is further reduced to $n!$. The V_{ij} are chosen such that

$$\forall i : \sum_j V_{ij} = 1, \quad \forall j : \sum_i V_{ij} = 1, \tag{47}$$

implying that both in every row and in every column, the sum of occupied squares of the chess-board equals one. It now suffices to minimize the cost function....[9]

3 Incorporating constraints

The way how symmetric linear constraints can be built-in has already been discussed in section 1.5. Here, we try to generalize this idea in case of having asymmetric linear and non-linear constraints. Beforehand we make the important observation that our attempts to *directly* built-in *asymmetric* constraints all failed because of the empirical experience that the third requirement for the transfer functions g_i (concerning no disturbance of the cost function at low temperatures; see section 2.2.2) could not be fulfilled.

3.1 Incorporating asymmetric linear constraints

The optimization problem having one asymmetric linear constraint can be stated as

$$\text{minimize} \quad E(V) \quad \text{subject to :} \quad \sum_i v_i V_i = c, \tag{48}$$

where $c \in \Re$ and where every v_j is supposed to be a non-zero integer[10].

[9] This will be elaborated in the final version of the paper.

[10] In case of having real-valued v_j's, they can be approximated by a rational fraction. Then, the constraint in (48) can be rewritten as a constraint having only integer-valued v_j's using a multiplication by the least

3.1.1 General scheme

The basic idea behind the new approach is to realize a *transformation* from the given problem having one asymmetric constraint to a problem having several symmetric constraints *without* increasing the number of the motion equations of the system. The general scheme can be summarized as follows.

1. In the expression of $E(V) = E(V_1, V_2, \ldots, V_n)$, any V_i is replaced by $\mid v_i \mid$ *identical* neurons $V_{i,j}$ $(j = 1, \ldots, \mid v_i \mid)$ conform
$$V_i = \frac{1}{\mid v_i \mid}(V_{i,1} + V_{i,2} + \cdots + V_{i,|v_i|}), \tag{49}$$
yielding the cost function $E(V') = E(V_{1,1}, \ldots, V_{1,|v_1|}, V_{2,1}, \ldots, V_{2,|v_2|}, \ldots, V_{n,1}, \ldots, V_{n,|v_n|})$.
2. Using this expression $E(V')$, a set of motion equations is determined by elaborating
$$\dot{U}_{i,j} = \Leftrightarrow\frac{\partial E(V')}{\partial V_{i,j}} \Leftrightarrow U_{i,j}. \tag{50}$$
3. Finally, this set of motion equations is reduced to the number of variables of the original problem by re-substitution of V_i for any $V_{i,j}$ and by re-substitution of U_i for any $U_{i,j}$. The corresponding transfer function is given by
$$V_i = \frac{c\ \exp(\beta U_i)}{\sum_l \mid v_l \mid \exp(\beta U_l)}, \tag{51}$$
which implies the permanent fulfillment of the constraint (48).

The final set of n motion equations as found in the last step, should be used to solve the given constrained minimization problem. A formal proof of the correctness of this procedure is given in appendix A.

3.1.2 A toy example

To illustrate the above-given scheme, we consider the following toy minimization problem:

$$\text{minimize} \quad E(V) = V_1^2 \Leftrightarrow V_1 V_2 \quad \text{subject to}: \ 2V_1 + 3V_2 = 1. \tag{52}$$

In the first step, we replace $2V_1$ by the sum of two identical neurons $V_{1,1}$ and $V_{1,2}$ and, similarly, $3V_2$ by the sum of three identical neurons $V_{2,1}$, $V_{2,2}$, and $V_{2,3}$. Then, the problem can be formulated as

$$\text{minimize} \quad E(V') = (\tfrac{1}{2}V_{1,1} + \tfrac{1}{2}V_{1,2})^2 \Leftrightarrow (\tfrac{1}{2}V_{1,1} + \tfrac{1}{2}V_{1,2})(\tfrac{1}{3}V_{2,1} + \tfrac{1}{3}V_{2,2} + \tfrac{1}{3}V_{2,3}) \tag{53}$$

$$\text{subject to}: \ V_{1,1} + V_{1,2} + V_{2,1} + V_{2,2} + V_{2,3} = 1, \tag{54}$$

$$V_{1,1} = V_{1,2} \quad \text{and} \quad V_{2,1} = V_{2,2} = V_{2,3}. \tag{55}$$

Note that the new cost function $E(V)$ has the same value as the old one. In the second step, the corresponding motion equations of type (5) are determined:

$$\dot{U}_{1,j} = \Leftrightarrow\frac{\partial E(V')}{\partial V_{1,j}} \Leftrightarrow U_{1,j} = \Leftrightarrow(\tfrac{1}{2}V_{1,1} + \tfrac{1}{2}V_{1,2}) + \tfrac{1}{2}(\tfrac{1}{3}V_{2,1} + \tfrac{1}{3}V_{2,2} + \tfrac{1}{3}V_{2,3}) \Leftrightarrow U_{1,j}, \tag{56}$$

$$\dot{U}_{2,j} = \Leftrightarrow\frac{\partial E(V')}{\partial V_{2,j}} \Leftrightarrow U_{2,j} = \tfrac{1}{3}(\tfrac{1}{2}V_{1,1} + \tfrac{1}{2}V_{1,2}) \Leftrightarrow U_{2,j}, \tag{57}$$

where

$$V_{i,j} = \frac{\exp(\beta U_{i,j})}{\sum_{l=1}^{2} \exp(\beta U_{1,l}) + \sum_{l=1}^{3} \exp(\beta U_{2,l})}. \tag{58}$$

Since for any i all $V_{i,j}$ are equal, it suffices in practice to calculate only one member of every set of the motion equations (56) and (57). In the third step, we can therefore simply re-substitute V_i for any $V_{i,j}$ and U_i for any $U_{i,j}$ in these equations, yielding

$$\dot{U}_1 = \Leftrightarrow V_1 + \tfrac{1}{2}V_2 \Leftrightarrow U_1, \tag{59}$$

$$\dot{U}_2 = \tfrac{1}{3}V_1 \Leftrightarrow U_2, \tag{60}$$

common multiple of all denominators.

where

$$V_i = \frac{\exp(\beta U_i)}{2\exp(\beta U_1) + 3\exp(\beta U_2)}. \tag{61}$$

Choosing $\beta = 0.001$ in the simulation, the solution values $V_1 = 0.1999$ and $V_2 = 0.2000$ were found, corresponding to solution values at high temperature. By choosing a low temperature $\beta = 1000$, the solution $V_1 = 0.1006$, $V_2 = 0.2663$ is found, which approximates the exact one $V_1 = 0.1$, $V_2 = 4/15$. Other simulations showed similar results, all in agreement with the theory.

3.2 Incorporating non-linear constraints

It can also be tried to incorporate non-linear constraints by choosing appropriate transfer functions $g_i(U)$ that meet all the requirements as summed up in subsection 2.2.2. Starting with the second requirement, we easily discovered non-linear transfer functions that implement certain constraints, but, when trying to integrate these functions, we never managed to find the corresponding analytical expression of $G(U)$. This also blockaded the possibility of checking the fulfillment of the third requirement. So, up till now we were forced to confine ourselves to experimental checks. We here present a successful example. Consider the toy problem:

$$\text{minimize} \quad \tfrac{1}{2}(V_1 \Leftrightarrow 2V_2)^2 \qquad \text{subject to:} \quad V_1^2 + V_2^2 = 1. \tag{62}$$

It is easy to check that the exact solution of this problem is given by $V_1 = 2\sqrt{0.2} \approx 0.8944$, $V_2 = \sqrt{0.2} \approx 0.4472$. We further note that the subspace of $[0,1]^2$ describing the given constraint, consists of a curved line (instead of a right one in case of incorporating linear constraints). Using (34), the corresponding motion equations appear to be

$$\dot{U}_1 = \Leftrightarrow V_1 + 2V_2 \Leftrightarrow U_1, \tag{63}$$

$$\dot{U}_2 = 2V_1 \Leftrightarrow 4V_2 \Leftrightarrow U_2. \tag{64}$$

We choose the V_i's conform

$$V_i = \sqrt{\frac{\exp(\beta U_i)}{\sum_l \exp(\beta U_l)}}. \tag{65}$$

Choosing $\beta = 0.001$, the solution $V_1 = 0.7075$, $V_2 = 0.7067$ is found: the solution values are almost identical (as expected at a high temperature), approximating $\frac{1}{2}\sqrt{2}$. Taking $\beta = 1000$, the solution $V_1 = 0.8943$, $V_2 = 0.4474$ is encountered: these values approximate the above-mentioned exact solutions.

Several other toy problems showed similar behavior implying that the generalized Hopfield network has indeed certain capabilities of directly incorporating non-linear constraints, although we do not have a full understanding of these capabilities yet. It is further conjectured that asymmetric non-linear constraints can be tackled in the same way as has been done in case of dealing with asymmetric linear constraints (section 3.1) by performing an appropriate transformation of the original problem to one having symmetric constraints.

4 The Hopfield-Lagrange model

Another way to cope with constraints is by using Lagrange multipliers, the first example of which appeared in 1988 [31]. The constrained optimization problem at hand is converted into an extremization problem. In this first paper however, the Hopfield term was not part of the model. In a paper [39] from 1989, the full integration with the continuous Hopfield model took place in what we call the Hopfield-Lagrange model. Contrary to the requirement (7) used in the penalty approach, the constraints should now be formulated conform their original definition (1):

$$\forall \alpha : C_\alpha(V) = 0 \quad \Leftrightarrow \quad V \text{ represents a feasible solution.} \tag{66}$$

The energy of the model is given by

$$E_{\mathrm{hl}}(V,\lambda) = E(V) + \sum_\alpha \lambda_\alpha C_\alpha(V) + E_{\mathrm{h}}(V) \tag{67}$$

having the following corresponding set of differential equations

$$\dot{U}_i = \Leftrightarrow \frac{\partial E_{\mathrm{hl}}}{\partial V_i} = \sum_j w_{ij} V_j + I_i \Leftrightarrow \sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i, \tag{68}$$

$$\dot{\lambda}_\alpha = + \frac{\partial E_{\mathrm{hl}}}{\partial \lambda_\alpha} = C_\alpha(V), \tag{69}$$

where continuously $V_i = g(U_i)$. Thus, the values of the Lagrange multipliers can be estimated by the system itself by applying a gradient *ascent*.

4.1 Stability analysis

4.1.1 Background

Let us first take a simple problem in order to try to understand the trick of gradient ascent in (69). The toy problem used is:

$$\begin{aligned} &\text{minimize } E(V) = V_1^2, \\ &\text{subject to : } V_1 \Leftrightarrow 1 = 0. \end{aligned} \tag{70}$$

Using the Hopfield-Lagrange model with the sigmoid as transfer function, the energy function of the type (67) here equals

$$E_{\mathrm{hl,t}}(V, \lambda) = V_1^2 + \lambda_1 (V_1 \Leftrightarrow 1) + \frac{1}{\beta}((1 \Leftrightarrow V_1)\ln(1 \Leftrightarrow V_1) + V_1 \ln V_1). \tag{71}$$

At low temperatures, this energy expression simply reduces to

$$E_{\mathrm{pb,t}}(V, \lambda) = V_1^2 + \lambda_1 (V_1 \Leftrightarrow 1), \tag{72}$$

which is visualized in figure 3. To find the critical point $(V_1, \lambda_1) = (1, \Leftrightarrow 2)$ using a direct gradient

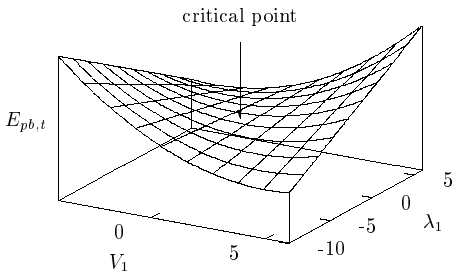

Figure 3: The energy landscape of $V_1^2 + \lambda_1(V_1 \Leftrightarrow 1)$.

method, we should apply a gradient *des*cent with respect to V_1 and, at the same time, a gradient *as*cent with respect to λ_1: the result is a spiral motion towards the critical point.

4.1.2 A potential Lyapunov function

We now analyze stability of the differential equations (68) and (69). To do so, we adopt the stability approach as proposed in the first paper [31]. Then, physics is our source of inspiration. We set up an expression of the sum of kinetic and potential energy. Taking the differential equations (68) and (69) together, one gets one second-order differential equation:

$$\ddot{U}_i = \Leftrightarrow \sum_j a_{ij} \frac{\mathrm{d}V_j}{\mathrm{d}U_j} \dot{U}_j \Leftrightarrow \dot{U}_i \Leftrightarrow \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i}, \tag{73}$$

where (a_{ij}) equals

$$a_{ij} = \Leftrightarrow w_{ij} + \sum_\alpha \lambda_\alpha \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j}. \tag{74}$$

Expression (73) coincides with the equation for a damped harmonic motion of a spring-mass system, where the mass equals 1, the spring constant equals 0, and the external force equals $\Leftrightarrow\sum_\alpha C_\alpha \partial C_\alpha/\partial V_i$.

Theorem 8. *If the matrix* (b_{ij}) *defined by*

$$b_{ij} = a_{ij}\frac{\mathrm{d}V_j}{\mathrm{d}U_j} + \delta_{ij} \tag{75}$$

(δ_{ij} being the Kronecker delta) first is or becomes and then remains positive definite, then the energy function

$$E_{\mathrm{kin+pot}} = \sum_i \tfrac{1}{2}\dot{U}_i^2 + \sum_{i,\alpha}\int_0^{U_i} C_\alpha \frac{\partial C_\alpha}{\partial V_i}\mathrm{d}u \tag{76}$$

is a Lyapunov function[11] *for the set of motion equations* (68) *and* (69).

Proof. Taking the time derivative of $E_{\mathrm{kin+pot}}$ and using (73) as well as the positive definiteness of (b_{ij}), we obtain

$$\begin{aligned}
\dot{E}_{\mathrm{kin+pot}} &= \sum_i \dot{U}_i\ddot{U}_i + \sum_{i,\alpha} C_\alpha\frac{\partial C_\alpha}{\partial V_i}\dot{U}_i \\
&= \sum_i \dot{U}_i\left(\Leftrightarrow\sum_j a_{ij}\frac{\mathrm{d}V_j}{\mathrm{d}U_j}\dot{U}_j \Leftrightarrow\dot{U}_i \Leftrightarrow\sum_\alpha C_\alpha\frac{\partial C_\alpha}{\partial V_i}\right) + \sum_{i,\alpha} C_\alpha\frac{\partial C_\alpha}{\partial V_i}\dot{U}_i \\
&= \Leftrightarrow\sum_{i,j}\dot{U}_i a_{ij}\frac{\mathrm{d}V_j}{\mathrm{d}U_j}\dot{U}_j \Leftrightarrow\sum_i \dot{U}_i^2 = \Leftrightarrow\sum_{i,j}\dot{U}_i b_{ij}\dot{U}_j \le 0.
\end{aligned} \tag{77}$$

Provided $E_{\mathrm{kin+pot}}$ is bounded below (which is expected to hold in view of its definition), its value constantly decreases until finally $\forall i : \dot{U}_i = 0$. From (68) we see that this normally implies that $\forall\alpha : \dot{\lambda}_\alpha = 0$ too. We then conclude from equations (68) and (69) that a stationary point of the Langrangian function $E_{\mathrm{hl}}(V,\lambda)$ must have been reached under those circumstances. Or, in other words, a constrained equilibrium point of the neural network is attained. □

Inspection of the derivation reveals why the gradient ascent is helpful in (69): only when the sign flip is applied, the two terms $\sum_i \dot{U}_i\sum_\alpha C_\alpha\partial C_\alpha/\partial V_i$ cancel each other. In order to prove stability, we should analyze the complicated matrix (b_{ij}) which in full equals

$$b_{ij} = \left(\Leftrightarrow w_{ij} + \sum_\alpha \lambda_\alpha\frac{\partial^2 C_\alpha}{\partial V_i\partial V_j}\right)\frac{\mathrm{d}V_j}{\mathrm{d}U_j} + \delta_{ij}. \tag{78}$$

Application of the Hopfield-Lagrange model to combinatorial optimization problems yields non-positive values for w_{ij}, so then $w'_{ij} \equiv \Leftrightarrow w_{ij} \ge 0$. If we confine ourselves to expressions C_α which are linear functions in V, then equation (78) reduces to

$$b_{ij} = w'_{ij}\frac{\mathrm{d}V_j}{\mathrm{d}U_j} + \delta_{ij}. \tag{79}$$

If the δ_{ij}-terms dominate, then (b_{ij}) is positive definite and stability is sure. However, it seems impossible to formulate general conditions which guarantee stability, since the matrix elements b_{ij} are a function of $\mathrm{d}V_j/\mathrm{d}U_j$ and thus change dynamically during the update of the differential equations. This observation explains why we called this subsection 'A *potential* Lyapunov function'. In practical applications, we can try to analyze matrix (b_{ij}). If this does not turn out successful, we may rely on experimental results. However, there is a way of escape, namely, by applying quadratic constraints. Under certain general conditions, they appear to guarantee stability in the long run at the cost of a degeneration of the Hopfield-Lagrange model to a type of penalty model.

[11] Since $E_{\mathrm{kin+pot}}$ is potential energy of the damped mass system, this function is a generalization of the Lyapunov function introduced in [31]. There, another equation was used with a quadratic potential energy term. This term cannot be used here because of the non-linear relationship $V_i = g(U_i)$. The quadratic term has to be modified in the integral as shown, while $\dot{V}_i$ is replaced by $\dot{U}_i$.

4.2 Degeneration to a dynamic penalty model

4.2.1 Non-unique multipliers

We consider the Hopfield-Lagrange model as defined in the beginning of this section.

Theorem 9. *Let W be the subspace of $[0,1]^n$ such that $V \in W \Rightarrow \forall\alpha : C_\alpha(V) = 0$ and let $V^0 \in W$. If the condition*

$$\forall\alpha, \forall i : C_\alpha = 0 \Rightarrow \frac{\partial C_\alpha}{\partial V_i} = 0 \tag{80}$$

holds, then there do not *exist unique numbers $\lambda_1^0, \ldots, \lambda_m^0$ such that $E_{\mathrm{hl}}(V,\lambda)$ has a critical point in (V^0, λ^0).*

Proof. The condition (80) implies that all $m \times m$ submatrices of the Jacobian (130) are singular. Conform the 'Lagrange Multiplier Theorem' of appendix B, uniqueness of the numbers $\lambda_1^0, \ldots, \lambda_m^0$ is not guaranteed. Moreover, in the critical points of E_{hl} as defined in (67), the following equations hold:

$$\sum_j w_{ij} V_j^0 + I_i \Leftrightarrow \sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i}(V^0) \Leftrightarrow U_i = 0. \tag{81}$$

Since $\forall\alpha : \partial C_\alpha / \partial V_i(V^0) = 0$, the multipliers λ_α may have *arbitrary* values in a critical point of $E_{\mathrm{hl}}(V,\lambda)$. □

In the literature [10, 15, 38, 39, 40] as well as in this paper, quadratic constraints are frequently encountered, often having the form

$$C_\alpha(V) = \tfrac{1}{2}(\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 = 0, \qquad \alpha = 1 \cdots m, \tag{82}$$

where any n_α equals some constant. Commonly, the constraints relate to only a subset of all V_i. So, for a constraint C_α, the index i_α passes through some subset N_α of $\{1, 2, \ldots, n\}$. We conclude that

$$\frac{\partial C_\alpha}{\partial V_i} = \begin{cases} \sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha & \text{if } i \in N_\alpha \\ 0 & \text{otherwise.} \end{cases} \tag{83}$$

It follows that condition (80) holds for the quadratic constraints (82). This implies that multipliers associated with those constraints are not uniquely determined in equilibrium points of the corresponding Hopfield-Lagrange model.

4.2.2 Stability yet

The question may arise how the Hopfield-Lagrange model deals with the non-determinacy of the multipliers[12]. To answer that question, we again consider (67), (68) and (69) and substitute the quadratic constraints (82). This yields

$$E_{\mathrm{hl,q}}(V,\lambda) = E(V) + \sum_\alpha \frac{\lambda_\alpha}{2}(\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 + E_{\mathrm{h}}(V), \tag{84}$$

$$\dot{U}_i = \Leftrightarrow \frac{\partial E}{\partial V_i} \Leftrightarrow \sum_{\alpha : i \in S_\alpha} \lambda_\alpha (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha) \Leftrightarrow U_i, \tag{85}$$

$$\dot{\lambda}_\alpha = \tfrac{1}{2}(\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2. \tag{86}$$

Theorem 10. *If $\forall i : V_i = g(U_i)$ is a differentiable and monotone increasing function, then the set of differential equations* (85) *and* (86) *is stable.*

Proof. We start by making the following crucial observations:

- As long as a constraint is not fulfilled, it follows from (86) that the corresponding multiplier increases:

$$\dot{\lambda}_\alpha > 0. \tag{87}$$

[12] This must be in a certain positive way, since the experiments as known from the literature were at least partially successful.

- If, at a certain moment, all constraint are fulfilled, then the motion equations (86) reduce to $\dot{\lambda}_\alpha = 0$ and the motion equations (85) reduce to (5) implying stability provided that the transfer function is differentiable and monotone increasing. This implies that instability of the system can *only* be caused by violation of one or more of the quadratic constraints.

We now consider the total energy $E_{\mathrm{hl,q}}$ of (84) and suppose that the system is initially unstable. One or more constraints must then be violated and the values of the corresponding multipliers will increase. If the instability endures, the multipliers will eventually become positive. It follows from (84) that the contribution

$$\sum_\alpha \frac{\lambda_\alpha}{2} (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 \tag{88}$$

to $E_{\mathrm{hl,q}}$ then consists of only convex quadratic forms, which correspond to various parabolic 'pits' or 'troughs' [13] in the energy landscape of $E_{\mathrm{hl,q}}$. As long as the multipliers grow, the pits become steeper and steeper. Eventually, the quadratic terms will dominate and the system will settle down in one of the created energy pits (whose location, we realize, is more or less influenced by E and E_{h}). In this way, the system will ultimately fulfill all constraints and will have become stable. □

Actually, for positive values of λ_α, the multiplier terms (88) fulfill the penalty term condition (7) and therefore act as penalty terms. As was sketched in the proof, the *system itself* always finds a feasible solution. This contrasts with the traditional penalty approach, where the experimenter may need a lot of trials to determine appropriate penalty weights. Moreover, the penalty terms might be 'as small as possible', having the additional advantage that the original cost function is minimally distorted. Since the penalty weights change dynamically on their journey to equilibrium, we have met with what we term a *dynamic penalty method*.

4.3 Stability analysis, the generalized Hopfield model

It is possible to extend the above-given stability analysis in case of combining the generalized continuous Hopfield model of section 2 with the multiplier approach of this section yielding an even more general framework. However, this will not be shown here, since the resulting matrix analogous to the matrix (75) appears to be very hard to analyze in practice [4].

4.4 Computational results, the unconstrained model

4.4.1 Simple optimization problems

We started by performing some simple experiments by trying various quadratic cost functions with linear constraints. The general form was

$$\begin{aligned} \text{minimize} \quad & E(V) = \tfrac{1}{2} \sum_{i=1}^{n} d_i (V_i \Leftrightarrow e_i)^2, \\ \text{subject to:} \quad & a_i^\alpha V_i \Leftrightarrow b_i^\alpha = 0, \ \alpha = 1, .., m, \end{aligned} \tag{89}$$

with positive values for d_i. The cost function was always chosen such that its minimum belongs to the state space $[0,1]^n$ and the constraints were always taken non-contradictory. Since for this class of problems

$$\frac{\partial^2 E}{\partial V_i \partial V_j} = d_i \delta_{ij} \quad \wedge \quad \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j} = 0, \tag{90}$$

the corresponding time derivative of the sum of kinetic and potential energy equals

$$\dot{E}_{\mathrm{kin+pot,s}} = \Leftrightarrow \sum_{i=1}^{n} (d_i \frac{\mathrm{d}V_i}{\mathrm{d}U_i} + 1) \dot{U}_i^2 \leq 0. \tag{91}$$

Using the sigmoid as the transfer function, we found convergence for all problem instances.

[13] If i_α passes through the whole set $\{1, 2, \ldots, n\}$, $(\sum_{i_\alpha} V_{i_\alpha} - n_\alpha)^2$ represents a n-dimensional parabolic pit. If, instead, i_α passes through a proper subset of $\{1, 2, \ldots, n\}$, this quadratic expression represents a trough in the energy landscape of $E_{\mathrm{hl,q}}$. However, in both cases, we shall speak of pits.

4.4.2 The weighted matching problem

We now report the results of the computations concerning the Weighted Matching Problem (WMP) [15]. An even number of points in some space is given, where the points are linked together in pairs. The goal is to find the configuration with the minimal total length of links. Interpreting $V_{ij} = 1$ ($V_{ij} = 0$) as if point i is (not) linked to point j, where $1 \leq i < j \leq n$, we tried several formulations of the constraints. Using linear constraints, the corresponding system turned out to be *un*stable. Therefore, we continued by trying quadratic ones since then, stability is generally guaranteed, as was pointed out before. The corresponding formulation of the problem is

$$\text{minimize} \quad E(V) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} V_{ij},$$

subject to:

$$C_{1,i}(V) = \tfrac{1}{2}(\sum_{j=1}^{i-1} V_{ji} + \sum_{j=i+1}^{n} V_{ij} \Leftrightarrow 1)^2 = 0, \quad C_{2,ij}(V) = \tfrac{1}{2} V_{ij}(1 \Leftrightarrow V_{ij}) = 0. \tag{92}$$

The right-hand constraints in (92) describe the requirement that finally, every V_{ij} must equal either 0 or 1 since each $C_{2,ij}$ corresponds to a concave function whose minima are boundary extrema. The corresponding multipliers are denoted by ν_{ij}. The multipliers corresponding to the constraints $C_{1,i}$

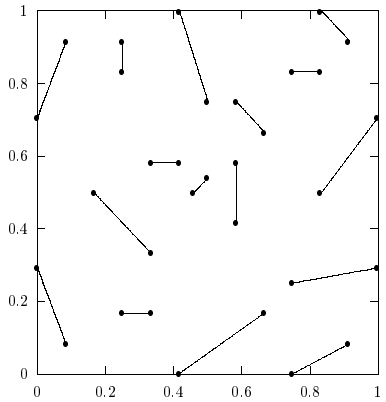

Figure 4: A solution of the WMP for $n = 32$.

are denoted by λ_i. All constraints (92) together enforce that every point is linked to precisely one other point. Again, the sigmoid was the selected transfer function. The experiments showed proper convergence. Using 32 points, the corresponding system consists of 1024 differential equations and 528 multipliers. The corresponding solution is visualized in figure 4. We repeated the experiment several times and always found solutions of similar quality.

In order to show how difficult the stability analysis can be when using theorem 8, we determined the matrix (75) in case of $n = 4$. Enumerating rows and columns in the order (1,2), (1,3), (1,4), (2,3), (2,4), (3,4), we found:

$$(b^{\mathrm{wm}}_{ij,kl}) = \begin{pmatrix} \Lambda_{12} & \lambda_1\Phi_{13} & \lambda_1\Phi_{14} & \lambda_2\Phi_{23} & \lambda_2\Phi_{24} & 0 \\ \lambda_1\Phi_{12} & \Lambda_{13} & \lambda_1\Phi_{14} & \lambda_3\Phi_{23} & 0 & \lambda_3\Phi_{34} \\ \lambda_1\Phi_{12} & \lambda_1\Phi_{13} & \Lambda_{14} & 0 & \lambda_4\Phi_{24} & \lambda_4\Phi_{34} \\ \lambda_2\Phi_{12} & \lambda_3\Phi_{13} & 0 & \Lambda_{23} & \lambda_2\Phi_{24} & \lambda_3\Phi_{34} \\ \lambda_2\Phi_{12} & 0 & \lambda_4\Phi_{14} & \lambda_2\Phi_{23} & \Lambda_{24} & \lambda_4\Phi_{34} \\ 0 & \lambda_3\Phi_{13} & \lambda_4\Phi_{14} & \lambda_3\Phi_{23} & \lambda_4\Phi_{24} & \Lambda_{34} \end{pmatrix}$$

where

$$\Lambda_{ij} = 1 + (\Leftrightarrow\nu_{ij} + \lambda_i + \lambda_j)\frac{\mathrm{d}V_{ij}}{\mathrm{d}U_{ij}} \quad \wedge \quad \Phi_{ij} = \frac{\mathrm{d}V_{ij}}{\mathrm{d}U_{ij}}. \tag{93}$$

In general, we can not prove convergence because the properties of the matrix b^{wm} change dynamically. However, stability in the initial and final states can easily be demonstrated. Initially, we set all multipliers equal to 0. Then, b^{wm} reduces to the unity matrix. On the other hand, if a feasible solution is found in the end, then $\forall i,j : V_{ij} \approx 0$ or $V_{ij} \approx 1$ implying that all $\Phi_{ij} \approx 0$. This again implies that b^{wm} reduces to the unity matrix. Since the unity matrix is positive definite, stability is guaranteed both at the start and in the end. However, during the updating process, the situation is much less clear.

4.4.3 The Travelling Salesman Problem

The TSP may be considered as an extension of the n-rook problem where, in addition to the satisfaction of the constraints of the permutation matrix, a cost function (namely, the tour length of the salesman) should be minimized. Using the Hopfield-Lagrange model in a similar way as in [39], we found proper convergence to feasible solutions although the quality for larger problem instances was very poor [4]. Inspired by the success with the WMP, we tried to solve the TSP using much more multipliers. This intervention is expected to improve the system performance because of an increased 'flexibility'. The modified problem is to find an optimal extremum of

$$\begin{aligned} E_{\mathrm{hl,u,tsp2}}(V,\lambda) &= \sum_{i,j,k} V_{ij} d_{ik} V_{kj+1} + \sum_i \frac{\lambda_i}{2} (\sum_k V_{ik} \Leftrightarrow 1)^2 + \\ &\quad \sum_j \frac{\mu_j}{2} (\sum_k V_{kj} \Leftrightarrow 1)^2 + \sum_{i,j} \frac{\nu_{ij}}{2} V_{ij} (1 \Leftrightarrow V_{ij}). \end{aligned} \tag{94}$$

The corresponding set of differential equations equals

$$\begin{aligned} \dot{U}_{ij} &= \Leftrightarrow \sum_k d_{ik} (V_{kj+1} + V_{kj-1}) \Leftrightarrow \lambda_i (\sum_k V_{ik} \Leftrightarrow 1) \Leftrightarrow \\ &\quad \mu_j (\sum_k V_{kj} \Leftrightarrow 1) \Leftrightarrow \nu_{ij} (\tfrac{1}{2} \Leftrightarrow V_{ij}) \Leftrightarrow U_{ij}, \end{aligned} \tag{95}$$

$$\dot{\lambda}_i = \sum_i \tfrac{1}{2} (\sum_k V_{ik} \Leftrightarrow 1)^2, \quad \dot{\mu}_j = \sum_j \tfrac{1}{2} (\sum_k V_{kj} \Leftrightarrow 1)^2, \quad \dot{\nu}_{ij} = \sum_{i,j} \tfrac{1}{2} V_{ij} (1 \Leftrightarrow V_{ij}). \tag{96}$$

Again, the experiments showed proper convergence. For very small problem instances we found

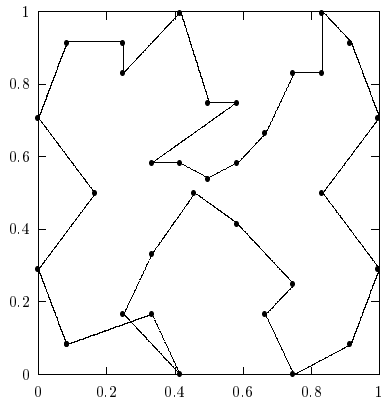

Figure 5: A solution of the TSP2 for $n = 32$.

optimal solutions. Large problem instances also yielded feasible, but usually sub-optimal solutions. An example is given in figure 5, where 32 cities and 1088 multipliers were used. The quality of the solution was certainly better than the afore-mentioned approach, although still not optimal.

4.5 Computational results, the constrained model

It is interesting to experiment with combinations of the constrained Hopfield and the Hopfield-Lagrange model. Which part of the constraints is built-in and which part is tackled with multipliers, strongly depends on the structure of the problem. E.g., in case of the WMP (section 4.4.2), the constraints are highly interweaved and the constrained model in its original form is not at all

applicable. In case of the TSP, the approach can be similar to that of the n-rook problem of section 2.3.2 using partially built-in constraints. Our experiments showed proper convergence but unfortunately, the quality of the solutions was not better than the quality of the already described solutions (for details, see [4]).

5 Elastic nets

For completeness we here mention some results concerning elastic nets. Elastic nets are neural networks with a very specific architecture designed to solve the TSP. The main reason to include this subject here is that in our opinion, contrary to what is suggested [35] and adopted [15, 42] in the literature, the classical Elastic Network Algorithm (ENA) [13] can not be derived from a constrained Hopfield network. In our view, the ENA should be considered as a *dynamic penalty method*. In addition to the analysis of the classical elastic net having a quadratic distance measure, we present *Non-equidistant* Elastic Net Algorithm (NENA) using a linear distance measure. Finally, a *Hybrid* Elastic Net Algorithm (HENA) is discussed.

5.1 Stochastic Hopfield and elastic neural networks

In order to prove our claim concerning the non-relationship between elastic and (constrained) Hopfield neural networks, we must first make a short side-step to discrete stochastic Hopfield networks and their mean field approximation. Like has been done in [35], we use Hopfield's discrete energy expression [17] multiplied by -1, i.e.

$$E(S) = \tfrac{1}{2} \sum_{ij} w_{ij} S_i S_j + \sum_i I_i S_i, \tag{97}$$

where $S \in \{0,1\}^n$ and all $w_{ij} \geq 0$. Making the units stochastic, the network can be analyzed applying statistical mechanics. We concentrate on the free energy [15, 26, 14]

$$F = \langle E(S) \rangle \Leftrightarrow T\mathcal{S}, \tag{98}$$

where $T = 1/\beta$ is the temperature, where $\langle E(S) \rangle$ represents the average energy, and where $\mathcal{S}$ equals the so-called entropy. A minimum of F corresponds to a thermal equilibrium state. We shall apply the next theorem [4, 6, 7] concerning an alternative energy expression for the continuous constrained Hopfield model[14]:

Theorem 11. *In mean field approximation, the free energy F_c of constrained stochastic binary Hopfield networks, submitted to the constraint $\sum_i S_i = 1$ equals*

$$F_c(V) = \Leftrightarrow \tfrac{1}{2} \sum_{ij} w_{ij} V_i V_j \Leftrightarrow \tfrac{1}{\beta} \ln[\sum_i \exp(\Leftrightarrow \beta(\sum_j w_{ij} V_j + I_i))]. \tag{99}$$

The stationary points of F_c are found at state space points where

$$V_i = \mathrm{P}(S_i = 1 \ \wedge \ \forall j \neq i : S_j = 0) = \frac{\exp(\Leftrightarrow \beta(\sum_j w_{ij} V_j + I_i))}{\sum_l \exp(\Leftrightarrow \beta(\sum_j w_{lj} V_j + I_l))}. \tag{100}$$

Note that in mean field approximation, the free energy is a continuous function in V, precisely like the energy F of the continuous Hopfield networks in our previous sections.

We now look at the specific mapping of the TSP onto the stochastic discrete Hopfield network as used in [35]. Let S_p^i denote whether the salesman at time i occupies space-point p ($S_p^i = 1$) or not ($S_p^i = 0$), then the corresponding cost function may be stated as

$$E(S) = \tfrac{1}{4} \sum_i \sum_{pq} d_{pq}^2 S_p^i (S_q^{i+1} + S_q^{i-1}) + \tfrac{\alpha}{4} \sum_i \sum_{pq} d_{pq}^2 S_p^i S_q^i. \tag{101}$$

The first term represents the sum of distance-squares (which strongly relates to the total tour length of the salesman), the second term penalizes the simultaneous presence of the salesman at more than

[14] For a detailed comparison between the various free energy expressions, we refer to [4, 6, 7].

one position. The other constraints, which should guarantee that every city is visited once, can be built-in 'strongly' using $\forall i : \sum_j S_{ij} = 1$.

By application of theorem 11 using expression (101) for $E(S)$, one finds [35, 8] the free energy

$$\begin{aligned} F_{\rm tsp}(V) &= \Leftrightarrow\tfrac{1}{4}\sum_i\sum_{pq} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}) \Leftrightarrow \tfrac{\alpha}{4}\sum_i\sum_{pq} d_{pq}^2 V_p^i V_q^i \Leftrightarrow \\ &\quad \tfrac{1}{\beta}\sum_p \ln\big[\sum_i \exp(\Leftrightarrow\tfrac{\beta}{2}\sum_q d_{pq}^2(\alpha V_q^i + V_q^{i+1} + V_q^{i-1}))\big]. \end{aligned} \tag{102}$$

On the other hand, the 'elastic net' algorithm [13, 15] uses the cost function

$$E_{\rm en}(x) = \tfrac{\alpha_2}{2}\sum_i \mid x^{i+1} \Leftrightarrow x^i \mid^2 \Leftrightarrow \tfrac{\alpha_1}{\beta}\sum_p \ln\sum_j \exp(\tfrac{-\beta^2}{2} \mid x_p \Leftrightarrow x^j \mid^2). \tag{103}$$

Here, x^i represents the i-th elastic net point and x_p represents the location of city p. Application of gradient descent on (103) yields the updating rule

$$\Delta x^i = \tfrac{\alpha_2}{\beta}(x^{i+1} \Leftrightarrow 2x^i + x^{i-1}) + \alpha_1 \sum_p \Lambda^p(i)(x_p \Leftrightarrow x^i), \tag{104}$$

where

$$\Lambda^p(i) = \frac{\exp(\Leftrightarrow\frac{\beta^2}{2} \mid x_p \Leftrightarrow x^i \mid^2)}{\sum_l \exp(\Leftrightarrow\frac{\beta^2}{2} \mid x_p \Leftrightarrow x^l \mid^2)} \tag{105}$$

and where the time-step $\Delta t = 1/\beta$ equals the current temperature T.

5.2 Why the ENA is a dynamic penalty method

In [35] a derivation of (103) from (102) is proposed while choosing $\alpha_1 = \alpha_2 = 1$. We here formulate three objections against this derivation. To do so, we repeat the main steps of the proof adding our comments after each step. First, in order to derive a free energy expression in the standard form (98), a Taylor series expansion on the last term of (102) is applied. We try to do the same. Taking

$$f(x) = \sum_p \ln\big[\sum_i \exp(x_p^i)\big], \tag{106}$$

$$a_p^i = \Leftrightarrow\beta\tfrac{\alpha}{2}\sum_q d_{pq}^2 V_q^i, \quad \text{and} \quad h_p^i = \Leftrightarrow\beta\tfrac{1}{2}\sum_q d_{pq}^2 (V_q^{i+1} + V_q^{i-1}), \tag{107}$$

and using (100) (adapted to the TSP, with $\alpha \gg 1$), we obtain [8]

$$\begin{aligned} f(a+h) &= \sum_p \ln\big[\sum_i \exp(a_p^i)\big] + \sum_{ip} h_p^i \frac{\partial f}{\partial x_p^i}(a_p^i) + \mathcal{O}(h^2) \\ &\approx \sum_p \ln\sum_i \exp\big(\Leftrightarrow\beta\tfrac{\alpha}{2}\sum_q d_{pq}^2 V_q^i\big) \Leftrightarrow \tfrac{\beta}{2}\sum_i\sum_{pq} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}). \end{aligned} \tag{108}$$

Substitution of this result in (102) yields:

$$\begin{aligned} F_{\rm app}(V) &= \tfrac{1}{4}\sum_i\sum_{pq} d_{pq}^2 V_p^i (V_q^{i+1} + V_q^{i-1}) \Leftrightarrow \tfrac{\alpha}{4}\sum_i\sum_{pq} d_{pq}^2 V_p^i V_q^i \Leftrightarrow \\ &\quad \tfrac{1}{\beta}\sum_p \ln\sum_i \exp\big(\Leftrightarrow\beta\tfrac{\alpha}{2}\sum_q d_{pq}^2 V_q^i\big). \end{aligned} \tag{109}$$

Objection 1. Since h_p^i is proportional to β, the Taylor-approximation (108) does not hold for high values of β. This is a fundamental objection because during the execution of the ENA, β is increased step by step. □

Second, a 'decomposition of the particle (salesman) trajectory' is performed in [35]:

$$x^i = \; <x(i)> \; = \sum_p x_p <S_p^i> = \sum_p x_p V_p^i. \tag{110}$$

$x(i)$ is the stochastic and x^i the average position of the salesman at time i. Using the decomposition, one writes

$$\sum_q d_{pq}^2 V_q^i = \mid x_p \Leftrightarrow x^i \mid^2 . \tag{111}$$

By this, a crucial transformation from a linear function in V_p^i into a quadratic one in x^i is achieved. Substitution of the result in (109) (with $\alpha = \beta$), neglect of the second term, and application of the decomposition (110) on the first term of (109) finally yield (103).

Objection 2. Careful analysis [4, 8] shows that in general

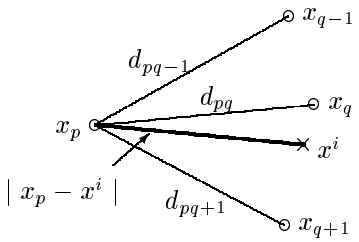

Figure 6: An elucidation of the inequality in (112).

$$\sum_q d_{pq}^2 V_q^i = \sum_q (x_p \Leftrightarrow x_q)^2 V_q^i \neq \mid x_p \Leftrightarrow x^i \mid^2 . \tag{112}$$

The explanation is as follows. The left-hand side of inequality (112) represents the expected sum of the distance squares between city point p and the particle position at time i, while the right-hand side represents the square of the distance between city point p and the expected particle position at time i. Under special conditions (e.g., if the constraints are fulfilled), the inequality sign must be replaced by the equality sign, but in general, the inequality holds (see also figure 6). □

Third, energy expression (102) is a special case of a generalized free energy of type (99), whose stationary points are solutions of

$$V_p^i = \frac{\exp(\Leftrightarrow \beta \sum_{jq} w_{pq}^{ij} V_q^j)}{\sum_l \exp(\Leftrightarrow \beta \sum_{jq} w_{pq}^{lj} V_q^j)}. \tag{113}$$

So, whatever the temperature may be, these stationary points are found at states where on average, all strongly submitted (i.e., all directly built-in) constraints are fulfilled. Moreover, the stationary points of a free energy of type (99) are often maxima [7, 8].

Objection 3. An analysis of the free energy of the ENA (section 3) yields a very different view because both terms of (103) create a set of minima. Therefore, a *competition* takes place between feasibility and optimality, where the current temperature determines the overall effect. This corresponds to the classical penalty method. A difference from that approach is that here – like in the Hopfield-Lagrange model of section 4 – the penalty weights change dynamically. Consequently, we consider the ENA a *dynamic penalty method*. □

The last observation corresponds to the theory of so-called deformable templates [30, 42], where the corresponding Hamiltonian equals

$$E_{\mathrm{dt}}(S, x) = \tfrac{\alpha_2}{2} \sum_i \mid x^{i+1} \Leftrightarrow x^i \mid^2 + \sum_{pj} S_p^j \mid x_p \Leftrightarrow x^j \mid^2 . \tag{114}$$

A statistical analysis of E_{dt} yields the free energy (103). A comparison between (103) and (114) clarifies that the first energy expression is derived from the second one by adding noise *exclusively* to the penalty terms. This completes our list of arguments against the suggested relationship [35] between Hopfield and elastic neural nets.

5.3 An analysis of the ENA

We can analyze the ENA by inspection of the energy landscape [8]. The general behavior of the algorithm leads to large-scale, global adjustments early on. Later on, smaller, more local refinements occur. *Equidistance* is enforced by the first, 'elastic ring term' of (103), which corresponds to parabolic pits in the energy landscape. *Feasibility* is enforced by the second, 'mapping term' corresponding to pits whose width and depth depend on T. Initially, the energy landscape appears to shelve slightly and is lowest in regions with high city density. On lowering the temperature a little, the mapping term becomes more important: it creates steeper pits around cities. By this, the elastic net starts to stretch out.

We next consider two potential, nearly final states of a problem instance with 5 *permanently* fixed cities and 5 variable elastic net points, of which 4 are *temporarily* fixed. The energy landscape of the remaining elastic net point is displayed. Figure 1 shows the case where 4 of the 5 cities have already caught an elastic net point. The landscape of the 5-th ring point (that is completely dominated by

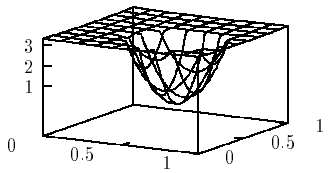

fig. 1. The energy landscape, a non-feasible state.

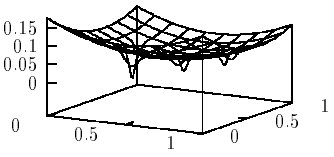

fig. 2. The energy landscape, an almost feasible state.

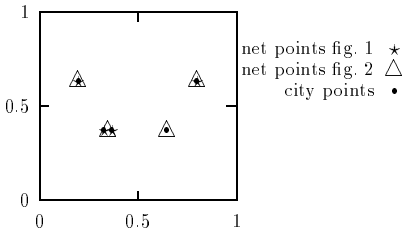

fig. 3. Net point and city point positions.

the mapping term) exhibits a relatively large pit situated above the only non-visited city. If the point is not too far away from the non-visited city, it can still be caught by it. It demonstrates, that a too rapid lowering of the temperature may lead to a non-valid solution.

In figure 2, an almost feasible, final solution is shown, where 3 net points coincide with 3 cities. A 4-th elastic net point is precisely in the middle between the two close cities. Now, the mapping term only produces some small pits. By this, the quadratic elastic ring term has become perceptible too. Hence, the remaining elastic net point is most probably forced to a point in the middle of its neighbors making the final state more or less equidistant, but not feasible!

Summarizing, it is possible to end up in a non-feasible solution if (a) the parameter T is lowered too rapidly or if (b) two close cities have caught the same net point. It is further quite interesting to note that in optimal annealing [1], the temperature is decreased carefully to *escape* from local minima. In this case however, annealing is applied carefully just to *end up* in a local (namely, a constrained) minimum. For much more details on the analysis of the ENA, we refer to [36, 4, 8].

5.4 Alternative elastic net algorithms

Various adaptations of the original ENA has been proposed, see, e.g., [11]. In order to use a correct distance measure and at the same time, to get rid of the equidistance property, we here propose an alternative elastic net algorithm having a *linear* distance measure in the energy expression (103):

$$F_{\mathrm{lin}}(x) = \alpha_2 \sum_i \mid x^{i+1} \Leftrightarrow x^i \mid \Leftrightarrow \frac{\alpha_1}{\beta} \sum_p \ln \sum_j \exp(\frac{-\beta^2}{2} \mid x_p \Leftrightarrow x^j \mid^2). \tag{115}$$

Applying gradient descent, the corresponding motion equations are found [8]. A self-evident analysis shows that, like in the original ENA, the elastic net forces try to push elastic net points onto a straight line. There is, however, an important difference: once a net point is situated in *any* point on the straight line between its neighboring net points, it no longer feels an elastic net force. Equidistance is not pursued anymore and the net points have more freedom to move towards cities. We therefore conjecture that the 'non-equidistant elastic net algorithm' (NENA) will find feasible solutions more easily.

Since the elastic net forces are normalized by the new algorithm, a tuning problem arises. To solve this problem, all elastic net forces are multiplied by a certain factor in order to compensate

for the normalization. The final updating rule becomes:

$$\Delta x^i = \frac{\alpha_2}{\beta}\frac{1}{m}\sum_{1}^{m} \mid x^{i+1} \Leftrightarrow x^i \mid \left(\frac{x^{i+1} \Leftrightarrow x^i}{\mid x^{i+1} \Leftrightarrow x^i \mid} + \frac{x^{i-1} \Leftrightarrow x^i}{\mid x^{i-1} \Leftrightarrow x^i \mid} \right) + \alpha_1 \sum_{p} \Lambda^p(i)(x_p \Leftrightarrow x^i).$$

In a last try to improve performance, we merged the ENA and the NENA into a 'hybrid elastic net algorithm' (HENA): the algorithm starts using ENA in order to get a balanced stretching out and, after a certain number of steps, switches to NENA in order to try to guarantee feasibility.

5.5 Experiments

We started using the 5-city configuration of section 3. Using 5 up to 12 elastic net points, the original ENA produced only non-feasible solutions. Using 15 elastic net points, the optimal feasible solution is always found. Using 5 elastic net points, the new NENA already produced the optimal solution in some cases. A gradual increase of the number of elastic net points results into a rise of the percentage of optimal solutions found. Using only 10 elastic net points, we obtained a 100% score. We concluded that for small problem instances up till 25 cities, the here proposed NENA performs better than the original ENA.

However, the picture started to change having 30-city problem instances. As a rule, both algorithms are equally disposed to find a valid solution, but the quality of the solutions of the original ENA is generally better. Trying even larger problem instances, the NENA more frequently found a non-valid solution: inspection shows a strong *lumping* effect of net points around cities and sometimes a certain city is completely left out. Apparently, by disregarding the property of equidistance, a new problem (that of clotting of elastic net points) has originated.

At this point, the hybrid approach of HENA comes to mind. Up to 100 cities however, we were unable to find parameters which yield better solutions than the original ENA.

Summarizing we may say that the original ENA turns out to be a relatively very well balanced dynamic penalty term algorithm.

6 Discussion and Outlook

A generalized framework for continuous Hopfield networks has been formulated in this paper. It can be used for resolving constrained optimization problems: non-quadratic cost functions are admitted and under certain conditions, the constraints can directly be incorporated in the neural network (which strongly reduces the size of the search space). If it turns out to be impossible to directly build-in the constraints, they can be grappled with using Lagrange multipliers resulting in what we have called the Hopfield-Lagrange model. A last possibility to handle the constraints is to use dynamic penalty terms. In all these cases, annealing can be applied. When dynamic penalty terms are used, it is even possible to apply the annealing scheme exclusively to the penalty terms. In this special case, annealing is applied to enforce relaxation to a valid solution.

In this article, we mainly focussed on mathematical analyses. In order to get a real understanding of the performance of the models in practice, more work should be done. The theorems given here may be helpful to analyze the stability properties of the motion equations chosen. In case of dealing with 'constrained satisfaction problems' (without a function to be optimized) we think Hopfield models have already proven their strength. However, in case of dealing with harder problems such as constrained optimization problems, a lot of fine tuning is expected to be necessary in order to really end up in high quality solutions. We would like to underline the role of expert knowledge here. Modelling a cost function often strongly depends on the application domain. In addition, the initialization of the network should preferably not be done at random, but inspired by the type of solution expected (remember, e.g., the initialization of the elastic net algorithm).

Within the area of combinatorial optimization many specific domains exist like scheduling [27, 37], optimal routing [27], memory association, and image processing (segmentation and restauration) [21, 25]. Using the (new) capabilities of the generalized model, it seems worthwhile to investigate new mappings of the cost function in all those areas. By applying domain knowledge, it is expected that more natural cost functions can be constructed. In particular, reducing the number of local minima in the energy landscape and-or enlargement of the 'basins of attraction' should be tried. In order to further limit the effect of local minima, 'mean field annealing' can be applied exploiting its 'smoothing effect' on the original cost function. It is also appropriate to consider alternative 'barrier functions' [41].

It can further be tried to apply iterative updating rules like the ones mentioned in section 1.6.4. The stability properties of these schemes are often hard to analyze and sometimes worse than the ones we considered. But in cases where these motion equations appear to be stable, their convergence rate is often much higher.

The final results of the all these schemes should of course be compared to the results achieved with other optimization methods both inside the area of neural networks (like 'multi-scale optimization' [24]) and 'deformable templates' [42]) and outside this field (like local search [2]).

Besides doing all this, something still different can be done namely trying to transfer the ideas as emerged here (in the area of *continuous* neural networks) towards the world of *discrete* (and stochastic) neural networks like Boltzmann machines [15, 14]. Non-quadratic cost functions and the incorporation of constraints are probably the first ones to consider. Then, statistical mechanics seems to be the most appropriate tool to analyze these *generalized discrete Hopfield networks*. For proving stability for example, the criterium of the '(detailed) balance condition' [26] can be used to investigate under which conditions the generalized updating rules do guarantee relaxation to the Boltzmann-Gibbs distribution [15]. In addition, various versions of simulated annealing [1] are a natural choice for trying to find high-quality solutions of constrained optimization problems using this type of neural networks.

A The proof of the correctness of the scheme of section 3.1.1

We will proof the correctness of the general scheme for building-in asymmetric linear constraints as proposed in section 3.1.1. The original problem (48) to be solved is:

$$\text{minimize} \quad E(V) \quad \text{subject to:} \quad \textstyle\sum_{i=1}^{n} v_i V_i = c, \tag{116}$$

where $V = (V_1, V_2, \ldots, V_n)$ and every v_i is an integer. By application of the mentioned scheme, this problem is transformed into a new one defined by

$$\text{minimize} \quad E(V') \quad \text{subject to:} \quad \textstyle\sum_{i=1}^{n} \frac{v_i}{|v_i|} \sum_{j=1}^{|v_i|} V_{i,j} = c \quad \wedge \quad \forall i,j,k : V_{i,j} = V_{i,k}, \tag{117}$$

where

$$V' = (V_{1,1}, \ldots, V_{1,|v_1|}, V_{2,1}, \ldots, V_{2,|v_2|}, \ldots, V_{n,1}, \ldots, V_{n,|v_n|}). \tag{118}$$

In order to determine the constrained minima of the problems (116) and (117), we apply the technique of *Lagrange multipliers*. By doing so, two solutions sets are found which we denote as $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively.

Theorem 12. *The solution sets $\mathcal{S}_1$ and $\mathcal{S}_2$ coincide.*

Proof. Applying the multiplier technique, the first problem (116) can be formulated as finding the extrema of the Lagrangian function

$$L(V, \lambda) = E(V) + \lambda(\sum_i v_i V_i \Leftrightarrow c). \tag{119}$$

The extrema are found by resolving the set of equations

$$\frac{\partial L(V,\lambda)}{\partial V_i} = \frac{\partial E(V)}{\partial V_i} + \lambda v_i = 0, \tag{120}$$

$$\frac{\partial L(V,\lambda)}{\partial \lambda} = \sum_i v_i V_i \Leftrightarrow c = 0. \tag{121}$$

The second problem, given by (117), can be reformulated as finding the extrema of the Lagrangian function

$$L(V', \lambda) = E(V') + \lambda(\sum_i \frac{v_i}{\mid v_i \mid} \sum_j V_{i,j} \Leftrightarrow c). \tag{122}$$

Here, the extrema are found by resolving the set of equations

$$\frac{\partial L(V',\lambda)}{\partial V_{i,j}} = \frac{\partial E(V')}{\partial V_{i,j}} + \lambda \frac{v_i}{\mid v_i \mid} = 0, \tag{123}$$

$$\frac{\partial L(V',\lambda)}{\partial \lambda} = \sum_i \frac{v_i}{\mid v_i \mid} \sum_j V_{i,j} \Leftrightarrow c = 0. \tag{124}$$

We are now able to proof that the solutions of the last set of equations (123) and (124) precisely coincide with the solutions of the set of equations (120) and (121). Using the chain rule of differentiation, it follows from the definition of equation (49) that

$$\frac{\partial E(V')}{\partial V_{i,j}} = \frac{1}{\mid v_i \mid} \frac{\partial E(V)}{\partial V_i}. \tag{125}$$

By substitution of this result in equation (123) and by substitution of V_i for any $V_{i,j}$ in equation (124) as well (conform step 3 of the scheme, where j runs from 1 to $\mid v_i \mid$), the following set of equations is found:

$$\frac{\partial L(V', \lambda)}{\partial V_{i,j}} = \frac{1}{\mid v_i \mid} \frac{\partial E(V)}{\partial V_i} + \lambda \frac{v_i}{\mid v_i \mid} = 0 \tag{126}$$

$$\frac{\partial L(V', \lambda)}{\partial \lambda} = \sum_i v_i V_i \Leftrightarrow c = 0. \tag{127}$$

Comparing the last two equations (126) and (127) to the set of equations (120) and (121), we see that they are the same and that they therefore yield the same set of solutions. In other words, the solution sets $\mathcal{S}_1$ and $\mathcal{S}_2$ coincide. □

Finally, we observe that by application of the motion equations (50) together with the transfer functions (51), a constrained local minimum of the (transformed) problem (117) is calculated. By application of theorem 12, it is clear that this minimum is a local minimum of the *original* problem (116) as well. This proves the correctness of the general scheme introduced.

B Lagrange multipliers

The Lagrange multiplier method is a method for analyzing constrained optimization problems defined by

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to :} & C_\alpha(x) = 0, \quad \alpha = 1, \ldots, m, \end{array} \tag{128}$$

where $x = (x_1, x_2, \ldots, x_n)$, $f(x)$ is called the objective function, and the equations $C_\alpha(x) = 0$ are the m side conditions or constraints. The *Lagrangian function* L is defined by a linear combination of the objective function f and the m constraint functions C_α conform

$$L(x, \lambda) = f(x) + \sum_\alpha \lambda_\alpha C_\alpha(x), \tag{129}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ and where the λ_i's are called *Lagrange multipliers*. The class of functions whose partial derivatives are continuous, we denote by $\mathcal{C}^1$. The following 'Lagrange multiplier theorem' [3] gives a *necessary* condition for f to have a local extremum subject to the constraints (128):

Theorem 13. *Let $f \in \mathcal{C}^1$ and all functions $C_\alpha \in \mathcal{C}^1$ be real functions on an open set T of R^n. Let W be the subset of T such that $x \in W \Rightarrow \forall \alpha : C_\alpha(x) = 0$. Assume further that $m < n$ and that some $m \times m$ submatrix of the Jacobian associated with the constraint functions C_α is nonsingular at $x^0 \in W$, that is, we assume that the following Jacobian is nonsingular at x^0*

$$J(x^0) = \begin{pmatrix} C_1^1(x^0) & C_1^2(x^0) & \ldots & C_1^m(x^0) \\ \vdots & \vdots & & \vdots \\ C_m^1(x^0) & C_m^2(x^0) & \ldots & C_m^m(x^0) \end{pmatrix} \tag{130}$$

where $C_\alpha^j(x^0) = \partial C_\alpha(x^0)/\partial x_j, \ j \in \{1, \ldots, n\}$.
If f assumes a local extremum at $x^0 \in W$, then there exist real and unique numbers $\lambda_1^0, \ldots, \lambda_m^0$ such that the Lagrangian $L(x, \lambda)$ has a critical point in (x^0, λ^0).

The proof of this theorem [3] rests on the 'implicit function theorem'. The condition of the nonsingularity of the defined Jacobian matrix makes the vector λ^0 unique.

References

[1] E.H.L. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, 1989.

[2] E.H.L. Aarts and J.K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, 1997.

[3] A. Benavie. *Mathematical Techniques for Economic Analysis*. Prentice-Hall, 1972.

[4] J. van den Berg. Neural Relaxation Dynamics, Mathematics and Physics of Recurrent Neural Networks with Applications in the Field of Combinatorial Optimization, *PhD thesis*, Erasmus University Rotterdam, 1996.

[5] J. van den Berg. Sweeping generalizations of continuous Hopfield and Hopfield-Lagrange networks. *Proceedings of the 1996 International Conference on Neural Information Processing* (ICONIP'96), 1, 673-678, Hong Kong, 1996.

[6] J. van den Berg and J.C. Bioch. On the (Free) Energy of Hopfield Networks. In: *Neural Networks: The Statistical Mechanics Perspective*, Proceedings of the CTP-PBSRI Joint Workshop on Theoretical Physics, eds. J.H. Oh, C. Kwon, S. Cho, 233–244, World Scientific, 1995.

[7] J. van den Berg and J.C. Bioch. Some Theorems Concerning the Free Energy of (Un)Constrained Stochastic Hopfield Neural Networks. *Lecture Notes in Artificial Intelligence* 904: 298–312, 2-nd European Conference on Computational Learning Theory, EuroCOLT'95, Springer, 1995.

[8] J. van den Berg and J.H. Geselschap. An analysis of various elastic net algorithms. *Technical Report EUR-CS-95-06*, Erasmus University Rotterdam, Comp. Sc. Dept., Faculty of Economics, 1995.

[9] J. van den Berg and J.H. Geselschap. A non-equidistant elastic net algorithm. In: *Mathematics of Neural Networks: Models, Algorithms and Applications*, eds. S.W. Ellacott, J.C. Mason, and I.J. Anderson, 101–106, Kluwer Academic Operations Research/Computer Science Interfaces series, Boston, 1997.

[10] D.E. Van den Bout and T.K. Miller. Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing. *Biological Cybernetics* 62, 129–139, 1989.

[11] M. Budinich and B. Rosario. A Neural Network for the Travelling Salesman Problem with a Wel Behaved Energy Function, In: *Mathematics of Neural Networks: Models, Algorithms and Applications*, eds. S.W. Ellacott, J.C. Mason, and I.J. Anderson, 134–136, Kluwer Academic Operations Research/Computer Science Interfaces series, Boston, 1997.

[12] B.S. Cooper. Higher Order Neural Networks for Combinatorial Optimisation - Improving the Scaling Properties of the Hopfield Network, *Proceedings of the IEEE International Conference on Neural Networks* (ICNN'95), 1855–1860, Perth, Western Australia, 1995.

[13] R. Durbin and D. Willshaw. An Analogue Approach of the Travelling Salesman Problem Using an Elastic Net Method. *Nature*, 326:689–691, 1987.

[14] S. Haykin. *Neural Networks, A Comprehensive Foundation*, MacMillan, 1994.

[15] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.

[16] G.E. Hinton. Deterministic Boltzmann Learning Performs Steepest Descent in Weight-Space, *Neural Computation* 1, 143–150, 1989.

[17] J.J. Hopfield. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences, USA*, 79, 2554–2558, 1982.

[18] J.J. Hopfield. Neurons with Graded Responses Have Collective Computational Properties Like Those of Two-State Neurons, *Proceedings of the National Academy of Sciences, USA*, 81, 3088–3092, 1984.

[19] J.J. Hopfield and D.W. Tank. "Neural" Computation of Decisions in Optimization Problems. *Biological Cybernetics* 52, 141–152, 1985.

[20] S. Ishii. Chaotic Potts Spin. *Proceedings of the IEEE International Conference on Neural Networks* (ICNN'95), 1578–1583, Perth, Western Australia, 1995.

[21] B. Kosko. *Neural Networks for Signal Processing*, Prentice-Hall, 1992.

[22] N. Matsui and K. Nakabayashi. Minimum Searching by Extended Hopfield Model, *Proceedings of the IEEE International Conference on Neural Networks* (ICNN'95), 2648–2651, Perth, Western Australia, 1995.

[23] E. Mjolsness and C. Garrett. Algebraic Transformations of Objective Functions, *Neural Networks* 3, 651–669, 1990.

[24] E. Mjolsness, C. Garrett, and W.L. Miranker. Multiscale Optimization in Neural Nets, *IEEE Transactions on Neural Networks* 2, no. 2, 263–274, March 1991.

[25] M. Muneyasu, K. Hotta and T. Hinamoto. Image Restauration by Hopfield Networks Considering the Line Process, *Proceedings of the IEEE International Conference on Neural Networks* (ICNN'95), 1703–1707, Perth, Western Australia, 1995.

[26] G. Parisi. *Statistical Field Theory*. Frontiers in Physics, Addison-Wesley, 1988.

[27] Y-K. Park and G. Lee. Applications of Neural Networks in High-Speed Communication Networks, *IEEE Communications*, 33-10, 68–74, 1995.

[28] C. Peterson and J.R. Anderson. A Mean Field Theory Learning Algorithm for Neural Networks. *Complex Systems* 1, 995–1019, 1987.

[29] C. Peterson and B. Söderberg. A New Method for Mapping Optimization Problems onto Neural Networks, *International Journal of Neural Systems* 1, 3–22, 1989.

[30] C. Peterson and B. Söderberg. Artificial Neural Networks and Combinatorial Optimization Problems, In: *Local Search in Combinatorial Optimization*, eds. E.H.L. Aarts and J.K. Lenstra, John Wiley & Sons, 1997.

[31] J.C. Platt and A.H. Barr. Constrained Differential Optimization. *Proceedings of the IEEE 1987 NIPS Conference*, 612–621, 1988.

[32] *Proceedings of the IEEE International Conference on Neural Networks* (ICNN'96), sessions Optimization and Associative Memory I-V, Washington, DC, USA, 1996.

[33] A. Rangarajan, A. Yuille, S. Gold and E. Mjolsness. A Convergence Proof for the Softassign Quadratic Assignment Algorithm, *Advances in Neural Information Processing Systems 9*, eds. M.C. Mozer, M.I. Jordan, T. Petsche, MIT Press, 1997.

[34] D. Sherrington. Neural networks: The Spin Glass Approach. In *Mathematical Approaches to Neural Networks*, ed. J.G. Taylor, 261–292, Elsevier, 1993.

[35] P.D. Simic. Statistical Mechanics as the Underlying Theory of 'Elastic' and 'Neural' Optimisations. *Network* 1, 88–103, 1990.

[36] M.W. Simmen. *Neural Network Optimization*. Dept. of Physics Technical Report 92/521, PhD Thesis, University of Edinburgh, 1992.

[37] J. Starke, N. Kubota, and T. Fukuda. Combinatorial Optimization with Higher Order Neural Networks - Cost Oriented Competing Processes in Flexible Manufactoring Systems, *Proceedings of the IEEE International Conference on Neural Networks* (ICNN'95), 1855–1860, Perth, Western Australia, 1995.

[38] Y. Takefuji. *Neural Network Parallel Computing*, Kluwer Academic Publishers, 1992.

[39] E. Wacholder, J. Han, and R.C. Mann. A Neural Network Algorithm for the Traveling Salesman Problem. *Biological Cybernetics* 61, 11–19, 1989.

[40] G.V. Wilson and G.S. Pawley. On the Stabilitity of the Travelling Salesman Problem Algorithm of Hopfield and Tank. *Biological Cybernetics* 58, 63–70, 1988.

[41] L. Xu. On The Hybrid LT Combinatorial Optimization: New U-Shape Barrier, Sigmoid Activation, Least Leaking Energy and Maximum Entropy. *Proceedings of the 1995 International Conference on Neural Information Processing* (ICONIP'95), 1, 309-312, Beijing, 1995.

[42] A.L. Yuille. Generalized Deformable Models, Statistical Physics, and Matching Problems. *Neural Computation* 2, 1–24, 1990.