Constrained Optimization with a Continuous Hopfield-Lagrange Model

Jan van den Berg
berg@cs.few.eur.nl

Jan C. Bioch
bioch@cs.few.eur.nl

Department of Computer Science
Erasmus University Rotterdam

September 6, 1993

Abstract

In this paper, a generalized Hopfield model with continuous neurons using Lagrange multipliers, originally introduced in [12], is thoroughly analysed. We have termed the model the Hopfield-Lagrange model. It can be used to resolve constrained optimization problems.

In the theoretical part, we present a simple explanation of a fundamental energy term of the continuous Hopfield model. This term has caused some confusion as reported in [11]. It led to some misinterpretations which will be corrected. Next, a new Lyapunov function is derived which, under some dynamical conditions, guarantees stability of the the system. We explain why a certain type of frequently used quadratic constraints can degenerate the Hopfield-Lagrange model to a penalty method. Furthermore, a difficulty is described which may arise if the method is applied to problems with 'hard constraints'.

The theoretical results suggest a method of using the Hopfield-Lagrange model. This method is described and applied to several problems like Weighted Matching, Crossbar Switch Scheduling and the Travelling Salesman Problem. The relevant theoretical results are applied and compared to the computational ones. Various formulations of the constraints are tried, of which one is a new approach, where a multiplier is used for every single constraint.

1 Introduction

1.1 Motivation and Results

Since the appearance of Hopfield and Tank's article [7], many researchers have tried to solve combinatorial optimization problems using artificial neural networks. We shall concentrate on approaches that use some kind of Hopfield

model. Within this class of approaches, often a 'penalty method' is applied: constraints are formulated as cost functions and added to the energy funtion to be minimized. By giving enough weight to these penalty terms, the system must favour stable states that correspond to feasible solutions. Unfortunately, these approaches did not work well: the choice of the various weights appeared to be highly sensitive, especially in the case of larger problem instances. E.g., for the Travelling Salesman Problem severe scalability problems are encountered.

A classical approach in the field of constrained optimization is Lagrange method using multipliers. Only a few neural net researchers have explored this method. Although some successful applications have been reported, the theoretical reasons for these successes have, as far as we know, not been explained to the full. This, together with the feeling that the Lagrange method is a more natural way to attack combinatorial optimization problems than the penalty method, motivated our research.

The structure of this article is as follows. In this section the theoretical starting points are given. We give a description of the continuous Hopfield model, the ways in which combinatorial optimization problems are solved and we introduce the Hopfield-Lagrange model. In section 2 we present the theoretical results starting with an explanation of the Hopfield term. In the differential equation, this term corresponds to what Takefuji calls the 'decay term'. In contrast to his conclusion, we prove that this term is *not* harmful. After that, we propose and analyze a Lyapunov function. In the fourth subsection we prove why some type of frequently used, quadratic constraints guarantee stability. In general, this goes hand in hand with a 'degeneration' of the Hopfield-Lagrange model. In the final theoretical subsection a 'toy problem' is analyzed, which clarifies a difficulty related to quadratic constraints. In section 3 we report experimental results. In each case, we have tried to analyze the exact stability properties using the derived Lyapunov function. The way the constraints are formulated turns out to be crucial. How this is related to the properties of convergence of the system is discussed.

1.2 The Hopfield Model

In 1982 Hopfield introduced the idea of an 'energy function' into neural network theory using an asynchronous updating rule and binary units [5]. In 1984, he generalized the stochastic model to a deterministic continuous one [6], which is essentially a *parallel gradient descent* method. The corresponding updating rule equals

$$\frac{dU_i}{dt} = \Leftrightarrow \frac{\partial E_{HM}(\vec{V})}{\partial V_i} = \sum_j w_{ij} V_j + I_i \Leftrightarrow U_i, \tag{1}$$

and the energy function $E_{HM}(\vec{V})$ equals

$$-\frac{1}{2}\sum_{ij} w_{ij} V_i V_j - \sum_i I_i V_i + \sum_i \int_0^{V_i} g^{-1}(V) dV, \tag{2}$$

where $\vec{V}$ is the state vector $(V_1, \cdots, V_n)$ of the neural net and V_i the output of neuron i, w_{ij} is the interconnection strength from neuron j to neuron i, $\sum_j w_{ij}V_j$ is the internal input of neuron i, I_i is the (constant) external input, $U_i = \sum_j w_{ij}V_j + I_i$ is the total input of neuron i, $V_i = g(U_i)$ is the activation function, usually non-linear, bounded and monotone increasing. We use for g the sigmoid function

$$g(U) = \frac{1}{1 + e^{-\beta U}}, \tag{3}$$

where β is a positive constant.

The energy function appears to be a Lyapunov function: Hopfield showed that the time derivative of E_{HM} is monotone non-increasing if $w_{ij} = w_{ji}$ and g is monotone increasing. Finally, the system reaches an equilibrium point. At that moment, the system state hopefully corresponds to the (optimal) solution of the problem.

The term $\sum_i \int_0^{V_i} g^{-1}(V)dV$, which we shall call the 'Hopfield term' denoted by $E_H(\vec{V})$, has been explained by Hopfield mainly in a qualitative way: its global effect is a displacement of solutions towards the interior of the state space, so finally $\forall i : 0 < V_i < 1$ will hold and, therefore, the corresponding U_i-values will be *finite*.

1.3 Constrained Combinatorial Optimization

Various researchers have tried to solve combinatorial optimization problems using artificial neural networks. References can be found in [4]. We start with a general formulation of the problem:

$$\begin{aligned} &\text{minimize } E(\vec{V}) \\ &\text{subject to : } C_\alpha(\vec{V}) = 0, \quad \alpha = 1 \cdots m, \end{aligned} \tag{4}$$

where $E(\vec{V})$ is the energy function to be minimized and the terms $C_\alpha(\vec{V}) = 0$ are the constraints, describing a subspace of the state space $[0, 1]^n$.

There exist different ways in treating the constraints. One approach is applying a so-called 'penalty method'. This method adds extra terms to the energy function, which penalize violation of constraints [10]. The formulation of the constraints must be such that

$$\begin{aligned} &\sum_{\alpha=1}^{m} c_\alpha . C_\alpha(\vec{V}) \text{ has a minimum value} \Leftrightarrow \\ &\qquad \vec{V} \text{ represents a feasible solution.} \end{aligned} \tag{5}$$

Using a Hopfield type neural network, the general problem (4) is converted into:

$$\text{minimize } E_P(\vec{V}) = E(\vec{V}) + \sum_{\alpha=1}^{m} c_\alpha . C_\alpha(\vec{V}) + E_H(\vec{V}). \tag{6}$$

The corresponding update rule is:

$$\frac{dU_i}{dt} = \Leftrightarrow\frac{\partial E_P}{\partial V_i} = \Leftrightarrow\frac{\partial E}{\partial V_i} \Leftrightarrow \sum_\alpha c_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i. \tag{7}$$

The influence of the Hopfield term $E_H(\vec{V})$ may be small, as will be shown. Ignoring this term, the energy function E_P is a *weighted* sum of $m+1$ terms and hence a difficulty arises in determining correct weights c_α. The minimum of E_P will be a *compromise* between fulfilling the constraints and minimizing the original target function. This explains why the method did not appear to be very successful, when it was applied to the Travelling Salesman Problem [7, 13, 2]. The weights had to be determined by trial and error and appeared to be very sensitive when larger problem instances were tried. On the other hand, one may expect that for pure combinatorial problems (without a target function to be minimized) the penalty method can be useful.

In a second approach, the features of the neural net are changed. The alteration is usually such, that some constraints are automatically fulfilled (see e.g. [3] and [11]). As an example, observe the following constraint

$$\sum_i^n V_i \Leftrightarrow 1 = 0. \tag{8}$$

This constraint expresses the requirement that the sum of the n values V_i should be 1. In solving combinatorial problems, this implies that one value V_i should be (approximately) one and the value of all the others should be (approximately) 0. A disadvantage of this approach is that the implementation of this type of 'constraint satisfying properties' in the neural net can be difficult, especially when there are many mutually dependent constraints.

Another way of changing the features of the neural net was adopted in [8]. Here, two layers are used. The first one is based on city adjacency, the second detects closed subtours in intermediate solutions. However, the implementation of the second layer is not straightforward.

A third way of treating the constraints has been proposed by Platt and Barr in 1988 [10]. They used the Lagrange multiplier method to convert constrained optimization problems into unconstrained *extremization* ones. A solution of the general problem (4) is also a *critical* point of

$$E_{PB}(\vec{V}, \vec{\lambda}) = E(\vec{V}) + \sum_{\alpha=1}^{m} \lambda_\alpha . C_\alpha(\vec{V}), \tag{9}$$

where $\vec{\lambda}$ is the vector of multipliers $(\lambda_1, \cdots, \lambda_m)$. Now, the formulation of the constraints must be such that

$$\begin{aligned} &\forall\alpha : C_\alpha(\vec{V}) = 0 \Leftrightarrow \\ &\vec{V} \text{ represents a feasible solution.} \end{aligned} \tag{10}$$

Platt and Barr used the so-called Basic Differential Multiplier Method (BDMM) to estimate the values of the Lagrange multipliers: a gradient *descent* applied to (9) was used to find the values of the state variables V_i together with a gradient *ascent* to estimate the Lagrange multipliers (the reason of this sleight will be explained in section 2.3):

$$\frac{dV_i}{dt} = \Leftrightarrow\frac{\partial E_{PB}}{\partial V_i} = \Leftrightarrow\frac{\partial E}{\partial V_i} \Leftrightarrow\sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i}, \tag{11}$$

$$\frac{d\lambda_\alpha}{dt} = +\frac{\partial E_{PB}}{\partial \lambda_\alpha} = C_\alpha(\vec{V}). \tag{12}$$

Note that in formula (11) gradient descent is applied with differentiation of V_i and not of U_i as is done using the Hopfield model. Also, in (9) the Hopfield term $E_H(\vec{V})$ is lacking. Platt and Barr found an appropriate Lyapunov function:

$$E_{kin} + E_{pot} = \sum_i \frac{1}{2}(\frac{dV_i}{dt})^2 + \sum_\alpha \frac{1}{2} C_\alpha^2(\vec{V}). \tag{13}$$

The time derivative of the sum of kinetic and potential energy is monotone decreasing under some conditions.

In 1989, Wacholder, Han and Mann [12] applied the work of Platt and Barr to a concrete neural network with continuous units. They added the Hopfield term $E_H(\vec{V})$ to the energy function (9), so basicly they used

$$E_{HL}(\vec{V}, \vec{\lambda}) = E(\vec{V}) + \sum_\alpha \lambda_\alpha . C_\alpha(\vec{V}) + E_H(\vec{V}) \tag{14}$$

with corresponding set of differential equations

$$\frac{dU_i}{dt} = \Leftrightarrow\frac{\partial E_{HL}}{\partial V_i} = \Leftrightarrow\frac{\partial E}{\partial V_i} \Leftrightarrow\sum_\alpha \lambda_\alpha \frac{\partial C_\alpha}{\partial V_i} \Leftrightarrow U_i, \tag{15}$$

$$\frac{d\lambda_\alpha}{dt} = +\frac{\partial E_{HL}}{\partial \lambda_\alpha} = C_\alpha(\vec{V}). \tag{16}$$

In equation (15), the variable U_i has returned. In fact, this formulation is a generalization of the Hopfield model because no special assumptions are made about the form of the cost function $E(\vec{V})$. Furthermore, it uses Lagrange multipliers. This explains why we term this model the *Hopfield-Lagrange Model*.

Wacholder et al. applied their model to small scaled (Multiple) Travelling Salesman Problems using a formulation related to the one of Hopfield and Tank with quadratic constraints. We repeated their experiments and also obtained convergence, but the quality of the solutions was poor. We tried to analyze this fact: we claim that for this case the Hopfield-Lagrange method behaves like a penalty method (subsection 2.4 and section 3).

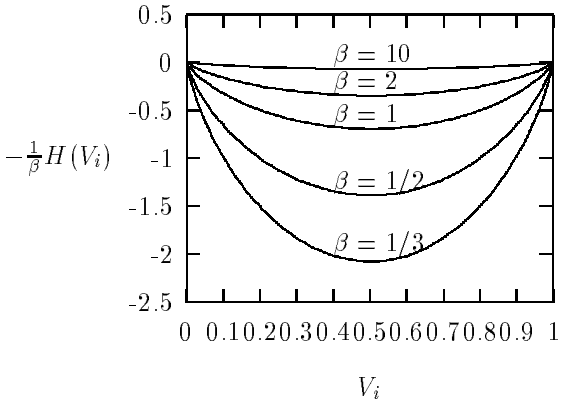

Figure 1: The term $-\frac{1}{\beta}H(V_i)$ for various values of β

2 Theoretical Results

2.1 The Hopfield Term

To simplify the discussion, consider an *unconstrained* minimization problem with total energy

$$E_{HL}(\vec{V}) = E(\vec{V}) + E_H(\vec{V}), \tag{17}$$

where $E(\vec{V})$ is the cost function to be minimized. The corresponding updating rule is:

$$\frac{dU_i}{dt} = \Leftrightarrow\frac{\partial E_{HL}}{\partial V_i} = \Leftrightarrow\frac{\partial E}{\partial V_i} \Leftrightarrow U_i. \tag{18}$$

We want to understand the influence of the Hopfield term $E_H(\vec{V}) = \sum_i \int_0^{V_i} g^{-1}(V)dV$. From (3) it follows, that

$$U = \frac{=1}{\beta}\ln(\frac{1 \Leftrightarrow V}{V}) = g^{-1}(V). \tag{19}$$

Using standard calculus, one finds:

$$\begin{aligned}\int_0^{V_i} g^{-1}(V)dV &= \tfrac{1}{\beta}[(1 \Leftrightarrow V_i)\ln(1 \Leftrightarrow V_i) + V_i \ln V_i] \\ &= \Leftrightarrow\tfrac{1}{\beta}H(V_i),\end{aligned} \tag{20}$$

where $H(V_i)$ equals the well known formula in Information Theory of the entropy of a binary source.

We make the following *mathematical* [1] observations (see also figure 1):

[1] A *physical* explanation using 'mean field theory' is also possible: introducing stochastic

- $\forall i : V_i \in [0,1] : E_H = \sum_i \int_0^{V_i} g^{-1}(V) dV \leq 0$.
- $-\frac{1}{\beta} H(V_i)$ has its minimum if $V_i = \frac{1}{2}$.
- For $\beta \to \infty$, $E_H(\vec{V}) = -\frac{1}{\beta} \sum_i H(V_i) \to 0$, so in the limit the Hopfield term does not influence the extrema of $E(\vec{V})$, as also Hopfield concluded.
- For finite values of β, $E_H(\vec{V})$ has a certain contribution: solutions, which are situated *in a corner of the hypercube* and of which $\partial E / \partial V_i = 0$ for all i, are displaced toward the interior (see figure 2). This is true for *any* finite value of β because the partial derivative $\partial E_H / \partial V_i$ for $V_i = 0$ and $V_i = 1$ respectively, equal -∞ and $+\infty$: the smaller the value of β the larger the displacement toward the interior. Even if an extremum in a corner of the hypercube appears to be a boundary extremum, the mentioned displacement will take place as long as the corresponding partial derivatives of $E(\vec{V})$ are finite.

  This displacement is often a pretty feature of the model. In a combinatorial optimization problem, solutions imply V_i-values equal to 0 or 1 with corresponding U-values $-\infty$ and $+\infty$. The Hopfield term is responsible for changing the values of V to ϵ and $1-\epsilon$, so the corresponding U-values then become *finite*. For high values of β these U-values will even be close to zero!
- Because the state space $\vec{V}$ is the entire N-dimensional hypercube $[0,1]^N$, it is possible that in the continuous model the minima of E are *in the interior of the hypercube*. Mostly, there will also be a displacement of solutions toward the interior but generally it will be a much smaller one.

There appears to be some confusion about the Hopfield term $E_H(\vec{V})$. As we have seen, it relates directly to the U_i-terms in the corresponding updating rules (1): $U_i = \partial E_H / \partial V_i$. Takefuji considers the 'decay term' U_i 'harmful' [11]. He concludes:

"Hopfield gives the motion equation of the i-th neuron (Hopfield and Tank 1985):

$$\frac{dU_i}{dt} = \Leftrightarrow \frac{U_i}{\tau} \Leftrightarrow \frac{\partial E}{\partial V_i} \tag{21}$$

(...) . Wilson and Pawley strongly criticised the Hopfield and Tank neural network through the Travelling Salesman Problem. Wilson and Pawley did not know what causes the problem. The use of the decay term $(-U_i/\tau)$ increases the computational energy function E under some conditions instead of decreasing it." (quote from pp. 6 and 7 in [11]).

---

binary neurons and interpreting the V_i's as their *average* values, we may write $E_{HL}(\vec{V}) = E(\vec{V}) - TS$, where $E(\vec{V})$ is the average energy function, $T = 1/\beta$ and S the entropy. This means, that the energy $E_{HL}(\vec{V})$ corresponds to the 'free energy' of the stochastic Hopfield network.

So Takefuji *suggests* (but does not prove) that the problems which Wilson and Pawley encountered, are caused by the decay term U_i/τ (in our formulations $\tau = 1$ holds). We think this suggestion is not correct, for two reasons. First, Takefuji does not add the Hopfield term $E_H(\vec{V})$ to the energy function in his analysis, but he does take up, like some others, the decay-term U_i in equation (21). Then he concludes, that this decay-term is responsible for incrementing the cost function $E(\vec{V})$ under some conditions. This conclusion is correct, but it is not a 'harmful' problem! On the contrary, it appears to correspond to the phenomenon we have described in this subsection, namely to the (small) displacement of solutions. We will prove this in the following subsection.

Second, Wilson and Pawley used the penalty method when they analyzed the Travelling Salesman Problem. We have mentioned the disadvantage of this method that it searches for a solution which is a compromise between optimality and feasibility (section 1.2): this should be considered as the main reason for the poor results of their approach.

2.2 The Decay Term is *not* Harmful

It is well known, that equation (21) continuously decreases $E_{HL} = E(\vec{V}) + E_H(\vec{V})$, because the time derivative $\dot{E} + \dot{E}_H \leq 0$. If, moreover, $E(\vec{V})$ is bounded below, the system is stable[2].

Takefuji argues in the following way that, under some conditions, the cost function $E(\vec{V})$ alone may increase: using equation (21) with $\tau = 1$, one finds:

$$\begin{aligned} \dot{E} &= \sum_i \frac{\partial E}{\partial V_i}\dot{V}_i = \sum_i (\Leftrightarrow\dot{U}_i \Leftrightarrow U_i)\dot{V}_i \\ &= \Leftrightarrow\sum_i (\dot{U}_i^2 + U_i\dot{U}_i)\frac{dV_i}{dU_i}. \end{aligned} \tag{22}$$

Because $\frac{dV_i}{dU_i} > 0$ (owing to the fact that $V_i = g(U_i)$ is monotone increasing), a necessary condition for an increase of $E(\vec{V})$ is: there should be at least one i such, that

$$\dot{U}_i^2 + U_i\dot{U}_i < 0, \tag{23}$$

which is equivalent to

$$\Leftrightarrow U_i < \dot{U}_i < 0 \quad \text{or} \quad 0 < \dot{U}_i < \Leftrightarrow U_i. \tag{24}$$

These two conditions correspond precisely to a displacement of a solution toward the interior of the state space. We shall prove that the first condition results in a displaced minimum with a *lower* value of V_i (the second corresponds to a displacement with a *higher* value). The left inequality of $-U_i < \dot{U}_i < 0$ implies that

$$\Leftrightarrow U_i \Leftrightarrow \dot{U}_i < 0. \tag{25}$$

[2]The first proof of this fact relates to the pure Hopfield model and can be found in [6]

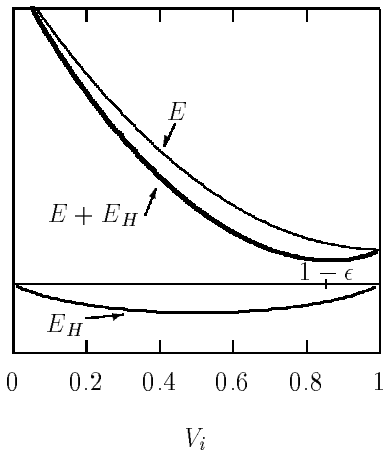

Figure 2: E, E_H and $E + E_H$ as function of V_i

Using (21) (with $\tau = 1$), one finds:

$$\frac{\partial E}{\partial V_i} = \Leftrightarrow U_i \Leftrightarrow \dot{U}_i < 0, \tag{26}$$

so E (as function of V_i) is decreasing.

The right inequality of $-U_i < \dot{U}_i < 0$ implies that $-\dot{U}_i > 0$. Again using (21) and $U_i = \frac{\partial E_H}{\partial V_i}$, one finds:

$$\frac{\partial E}{\partial V_i} + \frac{\partial E_H}{\partial V_i} = \frac{\partial E}{\partial V_i} + U_i = \Leftrightarrow \dot{U}_i > 0, \tag{27}$$

so the sum of E and E_H is increasing. The inequalities (26) and (27) together imply

$$\frac{\partial E_H}{\partial V_i} > 0, \tag{28}$$

so E_H as function of V_i is increasing. Therefore, $V_i > 0.5$. We have put this altogether in figure 2 (for the case that E has a minimum for $V_i = 1$). As one can verify, conditions (26) to (28) imply a displacement of the minimum to the interior, caused by the contribution of $E_H(V_i)$.

It is easy to prove, that the converse also holds: a displacement of a solution to a smaller value of V_i, caused by the Hopfield term, implies $-U_i < \dot{U}_i < 0$. This completes the proof.

2.3 A Lyapunov Function

Now, we return to consider *constrained* optimization using the Hopfield-Lagrange model. Consider the energy function (14) with the corresponding

set of differential equations (15) and (16). The question arises under which circumstances the set of differential equations will converge. Like Platt and Barr, we take physics as a source of inspiration. First, the equations (15) and (16) are taken together to form one second-order equation:

$$\ddot{U}_i = \Leftrightarrow \sum_j A_{ij} \frac{dV_j}{dU_j} \dot{U}_j \Leftrightarrow \dot{U}_i \Leftrightarrow \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i}, \tag{29}$$

where

$$A_{ij} = \frac{\partial^2 E}{\partial V_i \partial V_j} + \sum_\alpha \lambda_\alpha \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j}. \tag{30}$$

Equation (29) is the equation for a damped mass system. Now, we can propose a Lyapunov function consisting of the sum of kinetic and potential energy[3]:

$$E_{ph} = \sum_i \frac{1}{2} \left(\frac{dU_i}{dt} \right)^2 + \sum_{i,\alpha} \int C_\alpha \frac{\partial C_\alpha}{\partial V_i} dU_i. \tag{31}$$

We shall verify by taking the time derivative of E_{ph} that (31) is a Lyapunov function:

$$\begin{aligned}
\dot{E}_{ph} &= \sum_i \dot{U}_i \ddot{U}_i + \sum_i \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i} \dot{U}_i \\
&= \sum_i \dot{U}_i \left(- \sum_j A_{ij} \frac{dV_j}{dU_j} \dot{U}_j - \dot{U}_i - \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i} \right) \\
&\qquad + \sum_i \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i} \dot{U}_i \\
&= - \sum_{i,j} \dot{U}_i A_{ij} \frac{dV_j}{dU_j} \dot{U}_j - \sum_i \dot{U}_i^2 \\
&= - \sum_{i,j} \dot{U}_i B_{ij} \dot{U}_j = - \vec{\dot{U}}^T B \vec{\dot{U}},
\end{aligned} \tag{32}$$

where

$$B_{ij} = A_{ij} \frac{dV_j}{dU_j} + \delta_{ij}. \tag{33}$$

Here, δ_{ij} is the Kronecker delta. The derivation of equation (32) reveals why the gradient ascent (or *sign flip*) is needed in (16): only under these circumstances do the two terms $\sum_i \sum_\alpha C_\alpha \frac{\partial C_\alpha}{\partial V_i} \dot{U}_i$ cancel each other.

Now, we are able to determine conditions for which the Hopfield-Lagrange model will converge. These conditions depend on the properties of the matrix B[4]:

[3] This Lyapunov function is a generalization of the one Platt and Barr introduced [10]. They used the equation (13). Here, this term cannot be used because of the non-linear relationship $V = g(U)$.

[4] The matrix A equals the one that Platt and Barr found.

- If B is a positive definite matrix, we may conclude $\dot{E}_{ph} < 0$, if $\dot{U}_i \neq 0$. If, secondly, E_{ph} is bounded below, the system will converge until $\forall i : \dot{U}_i = 0$. Analysing equation (15), we see that, generally, this will occur when the values of λ_α have become constant, so when $\forall \alpha : \dot{\lambda}_\alpha = 0$. This can only be true if all constraints have been satisfied (see equation (16)).

- If A is positive definite then B will be positive definite too.

In practice it can be a tough job to analyze B, because its elements B_{ij} change dynamically during the update of the differential equations (15) and (16).

Concluding, we notice that monotony of a Lyapunov function provides a *sufficient*, not a *necessary* condition for stability [9]. Therefore, if we are not able to prove convergence, we can still experiment and hope for the best.

2.4 Degeneration to a Penalty Model

In this subsection we prove that, under some conditions, the multipliers of the Hopfield-Lagrange model are not uniquely determined. These conditions appear to hold for a type of frequently used, quadratic constraints. In case the constraints have the quadratic form, we shall prove stability of the Hopfield-Lagrange model. The stability often coincides with a degeneration to a penalty model, yielding large and positive multipliers.

Like in the previous subsection, we start considering the general model (equations (14), (15) and (16)). Suppose, the following condition holds:

$$\forall \alpha, \forall i : C_\alpha = 0 \Rightarrow \frac{\partial C_\alpha}{\partial V_i} = 0. \tag{34}$$

Then it follows from (15), that the values of the multipliers are arbitrary, i.e. *not* uniquely determined[5].

In the literature (e.g. in [3, 4, 11, 12, 13]) and in section 3 of this article, frequently, quadratic constraints can be found of the following form:

$$C_\alpha(\vec{V}) = \frac{1}{2}(\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 = 0, \quad \alpha = 1 \cdots m, \tag{35}$$

where every 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 S_α of $\{1, 2, \ldots, n\}$. Therefore, we note, that

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

[5] In fact, the conditions for the 'Lagrange Multiplier Theorem' ([1], p. 85) do not hold: from (34) it follows, that all $m \times m$ submatrices of the Jacobian, associated with the constraints, are *singular*. The theorem, which guarantees unique numbers λ_α, requires the existence of some *non-singular* submatrix.

Now, it follows that in any point of the state space where the quadratic constraints (35) are fulfilled, the condition (34) holds. Consequently, multipliers, associated with quadratic constraints (35), are not uniquely determined in equilibrium points.

The question arises, how the Hopfield-Lagrange model deals with this nondeterminacy. To answer that question, we consider the energy function (14) and the corresponding equations (15) and (16), using the mentioned quadratic constraints:

$$E_{HL} = E(\vec{V}) + \sum_{\alpha} \frac{\lambda_\alpha}{2} (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 + E_H(\vec{V}), \tag{36}$$

$$\frac{dU_i}{dt} = \Leftrightarrow \frac{\partial E}{\partial V_i} \Leftrightarrow \sum_{\alpha : i \in S_\alpha} \lambda_\alpha (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha) \Leftrightarrow U_i, \tag{37}$$

$$\frac{d\lambda_\alpha}{dt} = \frac{1}{2} (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2. \tag{38}$$

In order to prove eventual stability of this model, we make the following, crucial observations:

First, from equation (38) the important fact follows that, as long as a constraint is not fulfilled, the corresponding multiplier increases:

$$\frac{d\lambda_\alpha}{dt} > 0. \tag{39}$$

Second, if at a certain moment all constraint are fulfilled, the updating equations (37) and (38) are simply reduced to the following:

$$\frac{dU_i}{dt} = \Leftrightarrow \frac{\partial E}{\partial V_i} \Leftrightarrow U_i. \tag{40}$$

Under the general condition about boundedness of $E(\vec{V})$, this system is stable, as we pointed out in the beginning of section 2.2. This implies that instability of the system can *only* be caused by violation of one or more of the quadratic constraints (35). Now, consider the total energy E_{HL} (36). Normally, the system is not in equilibrium in the initial state, so multipliers will start growing when the updating of (37) and (38) begins. We also suppose the system is not stable initially (if it would be, the set of differential equations would converge rapidly). As just stated, the instability must be caused by violation of one or more constraints, so the values of the corresponding multipliers keep increasing and, if the system still remains unstable, they will become large and positive. Now, we conclude from equation (36) that the contribution of

$$\sum_{\alpha} \frac{\lambda_\alpha}{2} (\sum_{i_\alpha} V_{i_\alpha} \Leftrightarrow n_\alpha)^2 \tag{41}$$

to E_{HL} of all, still violated, constraints consists of convex quadratic forms. These forms correspond to ever deeper 'holes' in the 'energy landscape' of E_{HL}. Remembering the discussion of section 1 about the 'penalty method', we see,

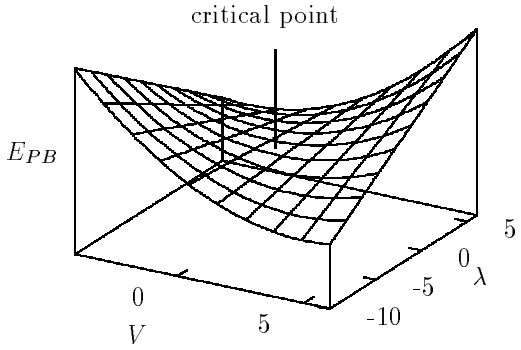

Figure 3: The energy landscape of $V^2 + \lambda(V - 1)$

that the quadratic constraints (35), multiplied by the positive Lagrange multipliers, have become 'penalty terms'. This is caused by the fact that, in the long run, they possess property (5) (in addition to property (10))! Eventually, the contribution (41) to E_{HL} will dominate over that of $E(\vec{V})$ and the system will settle down in one of the arisen 'energy holes' (whose location, as we know, may also have been influenced somewhat by the Hopfield term E_H). This completes the proof.

2.5 The Effect of Hard Constraints

To see how the theory of this section works in practice, we analyze the following toy problem:

$$\begin{aligned} &\text{minimize } f(\vec{V}) = V^2, \\ &\text{subject to: } V \Leftrightarrow 1 = 0. \end{aligned} \tag{42}$$

This simple problem can be solved manually. Applying a Lagrange multiplier, we find that

$$E_{PB} = V^2 + \lambda(V \Leftrightarrow 1) \tag{43}$$

has a critical point for $(V, \lambda) = (1, -2)$. Inspection of figure 3 shows in a different way why a *gradient ascent* of λ (or *sign flip*) is needed (together with a normal gradient descent of V) to find the critical point.

We also solve the problem using the Hopfield-Lagrange model: the energy function becomes

$$E_{HL} = V^2 + \lambda(V \Leftrightarrow 1) + \frac{1}{\beta}[(1 \Leftrightarrow V)\ln(1 \Leftrightarrow V) + V \ln V]$$

and we are looking for the critical point of E_{HL}, which we should find by resolving

$$\dot{U} = \Leftrightarrow \frac{\partial E}{\partial V} = \Leftrightarrow(2V + \lambda) \Leftrightarrow U, \tag{44}$$

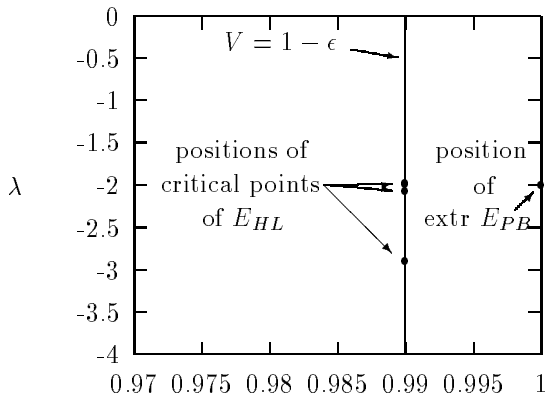

Figure 4: Positions of some critical points of E_{HL} (with a relaxed constraint) and of E_{PB} of the toy problem

$$\dot{\lambda} = +\frac{\partial E}{\partial \lambda} = V \Leftrightarrow 1. \tag{45}$$

Now, V is bounded to the interval $[0, 1]$. During updating E_{ph} is non-increasing because

$$\dot{E}_{ph} = \Leftrightarrow 2\dot{U}^2 . dV/dU \Leftrightarrow \dot{U}^2 \leq 0. \tag{46}$$

Consequently, E_{ph} is monotone decreasing until $\dot{U} = 0$ and therefore until $\dot{\lambda} = 0$, which implies $V = 1$ and $U = \infty$. Inspection of (44) reveals that, in equilibrium, λ should equal $-\infty$. So the critical point of E_{HL} appears if $(V, \lambda) = (1, -\infty)$ and we have run up against an unexpected difficulty. We have lost the pretty feature of the continuous Hopfield model of finding solutions corresponding to *finite* values of U. The reason is obvious: the 'hard' constraint $V - 1$ restricts the solution space to $V = 1$ with corresponding U-value equal to ∞. There exists a simple solution 'in the spirit' of the Hopfield model. If we relax the constraint to

$$V \Leftrightarrow 1 = \epsilon \tag{47}$$

then the new E_{HL} becomes

$$V^2 + \lambda(V \Leftrightarrow 1 + \epsilon) + \frac{1}{\beta}[(1 \Leftrightarrow V)\ln(1 \Leftrightarrow V) + V \ln V],$$

which has a critical point if $(V, \lambda) = (1 - \epsilon, -2 + \Delta\lambda)$, where

$$\Delta\lambda = 2\epsilon + \frac{1}{\beta}\ln(\frac{\epsilon}{1 \Leftrightarrow \epsilon}). \tag{48}$$

We see that the critical point is situated in the neighbourhood of the original value if the error $\Delta\lambda$ (which is determined by ϵ and β) is small. As in the Hopfield model, it can be kept small if we choose large values of β. To determine the sensitivity of the parameters, we present some results: we may conclude

$\epsilon = 0.001$		$\epsilon = 0.01$		$\epsilon = 0.1$	
β	$\Delta\lambda$	β	$\Delta\lambda$	β	$\Delta\lambda$
5000	$+0.0006$	5000	$+0.019$	5000	$+0.199$
500	$\Leftrightarrow 0.01$	500	$+0.010$	500	$+0.196$
50	$\Leftrightarrow 0.136$	50	$\Leftrightarrow 0.07$	50	$+0.156$
5	$\Leftrightarrow 1.38$	5	$\Leftrightarrow 0.90$	5	$\Leftrightarrow 0.239$
1	$\Leftrightarrow 6.9$	1	$\Leftrightarrow 4.58$	1	$\Leftrightarrow 1.997$

Table 1: The error $\Delta\lambda$ as a function of ϵ and β

from the calculations, as given in table 1, that sufficient high values of β indeed guarantee a small error $\Delta\lambda$.
In figure 4, some results have been put together. The position $(1, -2)$ of the constrained minimum is shown, together with some positions of the extrema of E_{HL} for various values of β and $\epsilon = 0.01$.
The described difficulty does not always take place. E.g., if the constraint has the form

$$V_1 + V_2 \Leftrightarrow 1 = 0, \tag{49}$$

then there may exist solutions of the form

$$V_1 = \epsilon \quad \text{and} \quad V_2 = 1 \Leftrightarrow \epsilon, \tag{50}$$

where V_1, V_2 correspond to finite values U_1, U_2.

3 Practical Results

Using the theory of section 1 and 2, we formulate a procedure to use the Hopfield-Lagrange model in constrained optimization problems, taking formula (4) as starting point. First, convert the constrained minimization problem (4) into the unconstrained extremization one using equations (14) to (16). Second, set up a Lyapunov function as given by (31), and try to prove convergence. If you are successful, you are lucky. If not, convergence is still possible, but it requires experimentation. Third, try to solve the problem using equations (15) and (16). Special attention is needed for the initialisation of the state variables V_i and λ_α, for the choice of parameter values β of the sigmoid function, for ϵ (if it applies) and for the numerical method (normally the Euler method is used, so you need to choose Δt and the maximum number of iterations).

3.1 Simple Optimization Problems

We started doing some simple experiments. We used various quadratic cost functions $E(\vec{V})$ with linear constraints:

$$E(\vec{V}) = \frac{1}{2} \sum_{i=1}^{n} d_i (V_i \Leftrightarrow e_i)^2, \tag{51}$$

$$\text{subject to}: \quad a_i^\alpha V_i \Leftrightarrow b_i^\alpha = 0, \ \alpha = 1, .., m, \tag{52}$$

where d_i was chosen positive. Every time, we chose the minimization function with a minimum belonging to the state space $[0,1]^n$. The corresponding Lyapunov function is monotone decreasing: because

$$\frac{\partial^2 E}{\partial V_i \partial V_j} = d_i \delta_{ij} \quad \text{and} \quad \frac{\partial^2 C_\alpha}{\partial V_i \partial V_j} = 0, \tag{53}$$

the time derivative of E_{ph} can be written as

$$\dot{E}_{ph} = \Leftrightarrow \sum_{i=1}^{n} (d_i . dV_i/dU_i + 1) \dot{U}_i^2 \leq 0. \tag{54}$$

Therefore, one may expect convergence for all problem instances. We started with our 'toy problem' applying equations (44) and (45) and choosing random initialisations of V_i.

In all cases, the system converged (with $E(\vec{V})$ monotone increasing). After 10^7 iterations, when U and λ were still growing and shrinking (to ∞ and -∞ respectively), we cut off the calculations. Using $\Delta t = 0.0001$ and $\beta = 50$ we found as 'final' values $V = 0.999959$ and $\lambda = -2.404412$.

Thereupon, we relaxed the constraint to $V - 1 = \epsilon$. Resolving the corresponding set of differential equations, choosing $\epsilon = 0.01$ and leaving the other constants unchanged, we found proper convergence to $V = 0.990000$ and $\lambda = -2.163805$: the first value is correct and the second one equals almost the theoretical value -2.07 (see table 1).

To investigate scalability, we extended the number of neurons and constraints. In all cases we got proper convergence. An example: minimize

$$V_1^2 + (V_2 \Leftrightarrow 1)^2 + V_3^2 + (V_4 \Leftrightarrow 1)^2 + \cdots + (V_{50} \Leftrightarrow 1)^2,$$

subject to:

$$\begin{cases} V_1 + V_2 + \cdots + V_{10} = 5 \\ V_6 + V_7 + \cdots + V_{15} = 5 \\ V_{11} + V_{12} + \cdots + V_{20} = 5 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ V_{41} + V_{42} + \cdots + V_{50} = 5 \end{cases}$$

After 10^6 iterations with $\Delta t = 0.0001$ and $\beta = 50$ we found:

$$\forall i : i \in \{1, 3, 5, \cdots, 49\} : V_i = 0.056360 \tag{55}$$

$$\forall i : i \in \{2, 4, 6, \cdots, 50\} : V_i = 0.943640, \tag{56}$$

so, the constraints have been fulfilled 'exactly'. We also observe the expected effect of the Hopfield term. The values of the 9 multipliers λ_α all equal 0.000000, corresponding to the theoretical ones.

We repeated the experiment, now chosing $\beta = 100$. We found:

$$\forall i : i \in \{1, 3, 5, \cdots, 49\} : V_i = 0.033593 \tag{57}$$

$$\forall i : i \in \{2, 4, 6, \cdots, 50\} : V_i = 0.966407. \tag{58}$$

The influence of the Hopfield term has diminished.

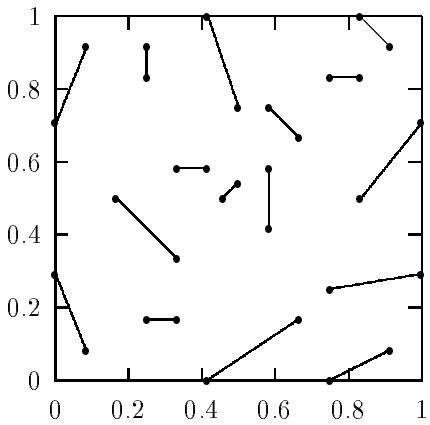

Figure 5: A solution of a Weighted Matching Problem for $n = 32$

3.2 Weighted Matching

Thereupon, we tried to solve *combinatorial* problems, starting with the Weighted Matching Problem: given n points with a known 'distance' d_{ij} between each ij-pair, the problem is to link the points pairwise, with each point linked to exactly one other point, so as to minimize the total length of the links (for more details see [4]).
We experimented with several formulations of the constraints and succeeded using the following one:

$$\text{minimize } E(\vec{V}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} V_{ij}, \tag{59}$$

subject to:

$$C_{1,i} = \frac{1}{2}(\sum_{j=1}^{i-1} V_{ji} + \sum_{j=i+1}^{n} V_{ij} \Leftrightarrow 1)^2 = 0, \tag{60}$$

$$C_{2,ij} = \frac{1}{2} V_{ij}(1 \Leftrightarrow V_{ij}) = 0. \tag{61}$$

The constraints (61) describe the requirement that every V_{ij} must equal 0 or 1. As corresponding multipliers we used ν_{ij}, where for the indices ij

$$\forall i, j : 1 \leq i < j \leq n \tag{62}$$

holds. The quadratic constraints (60) describe, in combination with (61), that every point should be linked to exactly one other. For these constraints, we used as multipliers λ_i.

The set of differential equations is given by

$$\begin{aligned}
\dot{U}_{ij} &= \Leftrightarrow d_{ij} \Leftrightarrow \lambda_i (\sum_{k=1}^{i-1} V_{ki} + \sum_{k=i+1}^{n} V_{ik} \Leftrightarrow 1) \Leftrightarrow \\
&\quad \lambda_j (\sum_{k=1}^{j-1} V_{kj} + \sum_{k=j+1}^{n} V_{jk} \Leftrightarrow 1) \Leftrightarrow \\
&\quad \nu_{ij} (\frac{1}{2} \Leftrightarrow V_{ij}) \Leftrightarrow U_{ij}, \qquad (63) \\
\dot{\lambda}_i &= \frac{1}{2} (\sum_{j=1}^{i-1} V_{ji} + \sum_{j=i+1}^{n} V_{ij} \Leftrightarrow 1)^2, \qquad (64) \\
\dot{\nu}_{ij} &= \frac{1}{2} V_{ij} (1 \Leftrightarrow V_{ij}). \qquad (65)
\end{aligned}$$

We tried to analyze matrix B. Since

$$\frac{\partial^2 E}{\partial V_{ij} \partial V_{kl}} = 0, \quad \frac{\partial^2 C_{1,m}}{\partial V_{ij} \partial V_{kl}} = (\delta_{mi} + \delta_{mj}).(\delta_{mk} + \delta_{ml})$$

$$\text{and} \quad \frac{\partial^2 C_{2,mn}}{\partial V_{ij} \partial V_{kl}} = \Leftrightarrow \delta_{mi}.\delta_{mk}.\delta_{nj}.\delta_{nl}, \qquad (66)$$

all elements B_{ijkl} are known (Note: $\forall i,j : 1 \leq i < j \leq n$ and $\forall k,l : 1 \leq k < l \leq n$). To get a firmer idea about the matrix B, we show it in case of 4 nodes. Enumerating rows and columns in the order (1,2), (1,3), (1,4), (2,3), (2,4), (3,4), we found:

$B_{WM} =$

$$\begin{pmatrix}
\Lambda_{12} & \lambda_1\Phi_{13} & \lambda_1\Phi_{14} & \lambda_2\Phi_{23} & \lambda_2\Phi_{24} & 0 \\
\lambda_1\Phi_{12} & \Lambda_{13} & \lambda_1\Phi_{14} & \lambda_3\Phi_{23} & 0 & \lambda_3\Phi_{34} \\
\lambda_1\Phi_{12} & \lambda_1\Phi_{13} & \Lambda_{14} & 0 & \lambda_4\Phi_{24} & \lambda_4\Phi_{34} \\
\lambda_2\Phi_{12} & \lambda_3\Phi_{13} & 0 & \Lambda_{23} & \lambda_2\Phi_{24} & \lambda_3\Phi_{34} \\
\lambda_2\Phi_{12} & 0 & \lambda_4\Phi_{14} & \lambda_2\Phi_{23} & \Lambda_{24} & \lambda_4\Phi_{34} \\
0 & \lambda_3\Phi_{13} & \lambda_4\Phi_{14} & \lambda_3\Phi_{23} & \lambda_4\Phi_{24} & \Lambda_{34}
\end{pmatrix}$$

where

$$\Lambda_{ij} = 1 + (\Leftrightarrow \nu_{ij} + \lambda_i + \lambda_j) \frac{dV_{ij}}{dU_{ij}} \qquad (67)$$

and

$$\Phi_{ij} = \frac{dV_{ij}}{dU_{ij}}. \qquad (68)$$

In general, we can not prove convergence because the properties of the matrix B change dynamically. Initially, we can set all multipliers equal to 0. At the end, all Φ_{ij} should be 0. So, at these moments we have a positive definite matrix B. During the updating process the situation is much less clear. Therefore, we decided to perform an experiment. The results appeared to be very good: in all cases, with both small and large problem instances, we got proper convergence. E.g., we got the following result using 32 points (the corresponding system has 1024 differential equations and 528 multipliers!). After 40 000 iterations with $\beta = 500$ and $\Delta t = 0.001$ the values of λ_i lay in the interval $[0.14; 0.83]$, while those of ν_{ij} were mostly of order 10^{-4} and sometimes of order 10^{-1}. The values of V_{ij} equalled 0.0000 or lay in the interval $[0.997; 1.000]$, which is interpreted as equal to 1.
We have repeated the experiment and found a solution of almost the same quality with 13 (of the 16) links equal to the ones that are shown in figure 5.
Trying other formulations of the constraints we e.g. modified (60) to a simple linear one

$$\sum_{k=1}^{i-1} V_{ki} + \sum_{k=i+1}^{n} V_{ik} \Leftrightarrow 1 = 0, \tag{69}$$

but now we did not find convergence. This proves experimentally the power of quadratic constraints being completely in conformity with the theory of subsection 2.4. However, we should be aware of the other side of the coin: quadratic constraints do not yield unique values of the multipliers and owe their success to their behaviour as penalty terms. So, we need more experimentation before coming to a definite judgement about them.

3.3 Crossbar Switch Scheduling and Travelling Salesman

3.3.1 General Formulation and Analysis

To see whether the Hopfield-Lagrange model is useful for solving more difficult combinatorial problems, we have tried to solve the benchmark Travelling Salesman Problem (TSP). The Crossbar Switch Scheduling Problem can sometimes be regarded as a special case of this problem.
We shall start with a formulation of the TSP which corresponds exactly to the one Hopfield and Tank [7] used:

$$\text{minimize } E(\vec{V}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} V_{ij} d_{ik} V_{kj+1}, \tag{70}$$

subject to:

$$C_1 \quad = \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k>j}^{n} V_{ij} V_{ik} = 0, \tag{71}$$

$$C_2 = \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{k>i}^{n} V_{ij}V_{kj} = 0, \tag{72}$$

$$C_3 = \frac{1}{2}(\sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij} \Leftrightarrow n)^2 = 0, \tag{73}$$

where V_{ij} means that city i is visited in the $j-$th position (indices should be taken 'modulo n') and where d_{ij} represents the distance between city i and city j ($d_{ij} = d_{ji}$).
Applying the Hopfield-Lagrange model to the TSP, we search for the extrema of

$$E_{HL}(\vec{V}, \vec{\lambda}) = E(\vec{V}) + \sum_{\alpha=1}^{3} \lambda_\alpha . C_\alpha(\vec{V}) + E_H(\vec{V}). \tag{74}$$

It is not very difficult to verify, that the used formulation of the constraints for the TSP have the encountered property of quadratic constraints: using positive penalty weights c_α both condition (5) and condition (10) hold for them. Therefore, one may expect the Hopfield-Lagrange model to behave as a penalty method, provided that all multipliers become positive.
Again, we tried to analyze the stability properties by, as usual, inspecting the matrix B (this time denoted by B_{TSP1}). Since

$$\frac{\partial^2 E}{\partial V_{ij}\partial V_{kl}} = (\delta_{lj-1} + \delta_{lj+1})d_{ik}, \quad \frac{\partial^2 C_1}{\partial V_{ij}\partial V_{kl}} = \delta_{ik}(1 \Leftrightarrow \delta_{jl}),$$

$$\frac{\partial^2 C_2}{\partial V_{ij}\partial V_{kl}} = \delta_{jl}(1 \Leftrightarrow \delta_{ik}) \quad \text{and} \quad \frac{\partial^2 C_3}{\partial V_{ij}\partial V_{kl}} = 1, \tag{75}$$

it is known. To get a better idea of this matrix, we show it for $n = 2$: $B_{TSP1} =$

$$\begin{pmatrix} 1+\lambda_3\Phi_{11} & (\lambda_1+\lambda_3)\Phi_{12} & (\lambda_2+\lambda_3)\Phi_{21} & (d_{12}+\lambda_3)\Phi_{22} \\ (\lambda_1+\lambda_3)\Phi_{11} & 1+\lambda_3\Phi_{12} & (d_{12}+\lambda_3)\Phi_{21} & (\lambda_2+\lambda_3)\Phi_{22} \\ (\lambda_2+\lambda_3)\Phi_{11} & (d_{21}+\lambda_3)\Phi_{12} & 1+\lambda_3\Phi_{21} & (\lambda_1+\lambda_3)\Phi_{22} \\ (d_{21}+\lambda_3)\Phi_{11} & (\lambda_2+\lambda_3)\Phi_{12} & (\lambda_1+\lambda_3)\Phi_{21} & 1+\lambda_3\Phi_{22} \end{pmatrix}$$

The expression Φ_{ij} is already been defined in (68).
If $\forall i : 1 \leq i, j \leq n : d_{ij} = 0$, then $E(\vec{V}) = 0$ and the TSP becomes a pure combinatorial one, that coincides with the Crossbar Switch Scheduling Problem(CSSP). The goal of traffic-scheduling of an $(n \times n)$ Crossbar Switch is to maximize the throughput of packets. The constraints are the following: at any time no two inputs may be connected to the same output and also, no input may be connected to more than one output [6]. If V_{ij} represents whether input i is connected to output j, and, if n connections should be made, the constraints of the CSSP equal the ones of TSP ((71), (72) and (73)).

[6] The problem coincides with the n-Queen problem

3.3.2 The Crossbar Switch Scheduling Problem

Since the Crossbar Switch Scheduling Problem is a purely combinatorial one, and the formulation of the constraints is such, that condition (5) holds, it is expected that this problem can be solved using the penalty method (section 1.3). The problem can be stated as:

$$\text{minimize } E_P = \sum_{\alpha=1}^{3} c_\alpha . C_\alpha(\vec{V}) + E_H(\vec{V}). \tag{76}$$

Using random initialisations of V_i and various values of the c_α's, we found convergence, provided that Δt was chosen small enough. E.g., for $n = 25$, $\beta = 500$, $\Delta t = 0.0001$ and all penalty weights $c_\alpha = 1$, we found convergence to 1 of the approximately 1.55×10^{25} solutions.
Thereupon, we applied the Hopfield-Lagrange model to the CSSP. Now, it is desired to find a critical point of

$$E_{HL} = \sum_{\alpha=1}^{3} \lambda_\alpha . C_\alpha(\vec{V}) + E_H(\vec{V}). \tag{77}$$

Again, we found convergence. E.g., for $n = 25$, $\beta = 500$ and $\Delta t = 0.0001$ we found, after 2000 iterations, $\lambda_1 = 0.655935$, $\lambda_2 = 0.649828$, a still growing multiplier $\lambda_3 = 0.690099$ and an almost feasible solution. The increase of λ_3 can easily be explained by the theory of section 2.5. Note, that all multipliers have become positive, making the contribution $\sum_\alpha \lambda_\alpha . C_\alpha$ behave as a sum of penalty terms. It is difficult to analyze general stability using the matrix B_{TSP_1}, because once again the values of the matrix elements change dynamically. However, in any point of the state space for which the constraints are satisfied, stability can easily be proven.

3.3.3 The Travelling Salesman Problem (1)

Using the Hopfield-Lagrange model, the TSP can be attacked by searching the extrema of (74). The corresponding set of differential equations equals

$$\begin{aligned}
\dot{U}_{ij} &= \Leftrightarrow d_{ij} \sum_k (V_{kj+1} + V_{kj-1}) \Leftrightarrow \lambda_1 \sum_{k=1, k\neq i}^{n} V_{ik} \Leftrightarrow \\
&\quad \lambda_2 \sum_{k=1, k\neq j}^{n} V_{kj} \Leftrightarrow \lambda_3 (\sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij} \Leftrightarrow n) \Leftrightarrow U_{ij}, \\
\dot{\lambda}_1 &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k>j}^{n} V_{ij} V_{ik}, \\
\dot{\lambda}_2 &= \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{k>i}^{n} V_{ij} V_{kj},
\end{aligned}$$

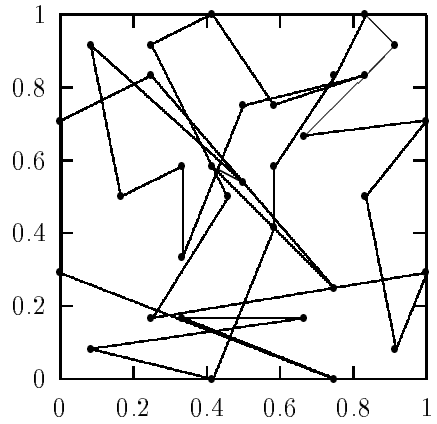

Figure 6: A solution of a TSP (1) for $n = 32$

$$\dot{\lambda}_3 \quad = \quad \frac{1}{2}(\sum_{i=1}^{n}\sum_{j=1}^{n} V_{ij} \Leftrightarrow n)^2. \tag{78}$$

We experimented with this set and, just as in [12], we found proper convergence to nearly feasible solutions. Unfortunately, the quality of the solutions was very poor. Even problem instances of 4 cities did not yield optimal solutions every time. Trying instances with 32 cities yielded solutions like the one shown in figure 6, i.e., solutions far from optimal.

3.3.4 The Travelling Salesman Problem (2)

Inspired by the success with Weighted Matching Problem, we tried to solve the TSP using a similar type of constraints, that is, we tried quadratic constraints with *one multiplier for every single constraint*. We expected to find better solutions, because in this approach many more multipliers are used, which should make the system more 'flexible'. We search an extremum of $E_{HL}(\vec{V}, \vec{\lambda}) =$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} V_{ij} d_{ik} V_{kj+1} + \sum_{i=1}^{n} \frac{\lambda_i}{2}(\sum_{k=1}^{n} V_{ik} \Leftrightarrow 1)^2 +$$
$$\sum_{j=1}^{n} \frac{\mu_j}{2}(\sum_{k=1}^{n} V_{kj} \Leftrightarrow 1)^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\nu_{ij}}{2} V_{ij}(1 \Leftrightarrow V_{ij}), \tag{79}$$

The corresponding differential equations can be easily derived by differentiation. Again, the experiments showed proper convergence. For very small problem instances we found optimal solutions. Large problem instances also yielded feasible, but unfortunately, not optimal solutions. An example of a solution is given in figure 7, where 32 cities were used, $\Delta t = 0.001$ and the applied number of iterations was 100 000. The encountered values of V_{ij} were either 0.0000 or lay

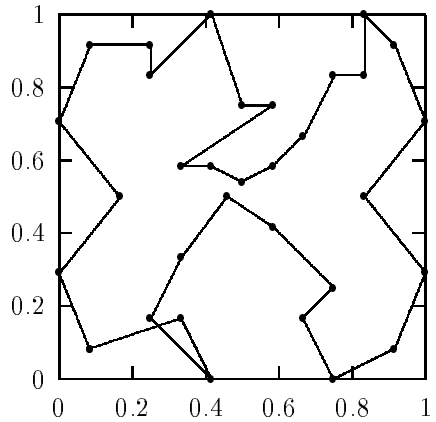

Figure 7: A solution of a TSP (2) for $n = 32$

in the interval [0.9988;1.0000]. The 1088 multipliers were still growing (although very slowly) in order to realize exact fulfilment of the constraints. The quality of the solution is certainly better than the one we found in the previous subsection, although still far from optimal.

4 Conclusions and Outlook

In this report, the Hopfield-Lagrange model has been analyzed. The model is defined by the energy function (14) and the corresponding set of differential equations (15) and (16).
In section 2, the theoretical framework of the model has been shown. We succeeded in giving a simple mathematical explanation of the Hopfield term E_H as well as in explaining why the corresponding decay term U_i is not harmful. The proposed Lyapunov function yields, at least in theory, a strong tool to analyze the properties of convergence of the system. Some type of frequently used, quadratic constraints appear to guarantee stability of the Hopfield-Lagrange model. However, this goes hand in hand with a degeneration to a penalty model with, eventually, positive values for the multipliers (penalty weights). In contrast to the traditional penalty method these values are calculated by the model itself, which is an advantage. It has also been shown, that hard constraints may imply infinite values of multipliers. Sometimes, this can be resolved by relaxing the constraints.
The experimental results do not falsify the theoretical ones, although the quality of the solutions is not always very good. The investigated *quadratic optimization* problems, subject to linear constraints, yielded the correct solutions with the minima somewhat displaced by the Hopfield term, as the theory predicted. Also, the correct values of the multipliers were found. Scalability did not occur.

Purely combinatorial problems, like the Crossbar Switch Scheduling Problem, can also be solved using the Hopfield-Lagrange model. We applied quadratic constraints like many researchers and always got *almost correct* solutions, corresponding to the theory about hard constraints. Equally, no scalability difficulties were found.
For *combinatorial optimization* problems the situation is more complicated. Often, the concrete, corresponding Lyapunov function can not be analyzed sufficiently with mathematical tools. A guarantee for stability could only be given in case of quadratic constraints. Therefore, we were forced to rely mainly on experiments. Until now, we only found feasible solutions when a quadratic formulation of the constraints was used. For relatively simple combinatorial optimization problems (like the Weighted Matching Problem) and for small scaled difficult ones (like the Travelling Salesman Problem), the approach yielded solutions of high (sometimes optimal) quality, provided that a lot of constraints (one multiplier for every single constraint) were used. Unfortunately, difficult combinatorial optimization problems again demonstrated the scalability difficulty: large scaled instances gave solutions far from optimal. Still, the models using a lot of multipliers did much better than the ones using only a few.
Further research is necessary to explain why the used linear constraints did not work for these problems. Also, it is interesting to investigate other mappings trying to find feasible solutions of better quality. In all research about stability, the theory about the derived Lyapunov function can be useful, but it demands a thorough analysis of the properties of matrix B (section 2.3). In this context, an analysis of the (deformation of the) energy landscapes is appropriate too, which may be related to the theory of statistical physics. Still another way to get more insight in the behaviour of the model may be to take up other combinatorial optimization problems. Only after having completed this type of research can a definite judgement about the Hopfield-Lagrange model be given.

5 Acknowledgements

We thank Marc Leipoldt for reading this article and suggesting some changes to the English.

References

[1] A. Benavie. *Mathematical Techniques for Economic Analysis.* Prentice-Hall, Inc.

[2] A.R. Bizzarri. Convergence properties of a modified hopfield-tank model. *Biological Cybernetics*, 64:293–300, 1991.

[3] 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.

[4] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, The advanced Book Program, 1991.

[5] 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.

[6] 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.

[7] J.J. Hopfield and D.W. Tank. "neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1986.

[8] A. Joppe, H.R.A. Cardon, and J.C. Bioch. A neural network for solving the travelling salesman problem on the basis of city adjacency in the tour. In *Proceedings of the International Joint Conference on Neural Networks*, pages 961–964, San Diego, june 1990.

[9] B. Kosko. *Neural Networks and Fuzzy Systems, a dynamical systems approach to machine intelligence*. Prentice-Hall International Editions, 1992.

[10] J.C. Platt and A.H. Barr. Constrained differential optimization. *Proceedings of the IEEE 1987 NIPS Conference*, 1988.

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

[12] E. Wacholder, J. Han, and R.C. Mann. A neural network algorithm for the traveling salesman problem. *Biological Cybernetics*, 61:11–19, 1989.

[13] 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.