April 1991

LU TP 91-9

Complex Scheduling with Potts Neural Networks

Lars Gislén, Carsten Peterson and Bo Söderberg

Department of Theoretical Physics, University of Lund
Sölvegatan 14A, S-22362 Lund, Sweden

Neural Computation **4**, 805 (1992)

Abstract:

In a recent paper (Gislén, Peterson and Söderberg 1989) a convenient encoding and an efficient mean field algorithm for solving scheduling problems using a Potts neural network was developed and numerically explored on simplified and synthetic problems.

In this work the approach is extended to realistic applications both with respect to problem complexity and size. This extension requires among other things the interaction of Potts neurons with different number of components. We analyze the corresponding linearized mean field equations with respect to estimating the phase transition temperature. Also a brief comparison with the linear programming approach is given.

Testbeds consisting of generated problems within the Swedish high school system are solved efficiently with high quality solutions as results.

1 Introduction

The neural network (NN) approach has shown great promise for producing good solutions to difficult optimization problems (Peterson 1990). The NN approach in this problem domain is very attractive from many perspectives. The mappings onto NN are often extremely straight-forward to do implying very modest software development. Parallel implementations are natural. Furthermore, by using the mean field theory (MFT) approximation a set of deterministic equations replace the CPU demanding stochastic updating procedures often used to avoid getting stuck in local minima. Most activities have focused on "artificial" applications like the graph partition and traveling salesman problems. Recently more nested and realistic problems like scheduling were approached with encouraging results (Gislén, Peterson and Söderberg 1989, hereafter referred to as GPS 1989) . The key strategies for obtaining good and parameter insensitive solutions with the NN approach for optimization problems in general are:

- Map the problem onto multi-state (Potts) neurons (Peterson and Söderberg 1989) rather than the binary neurons used in the original work by Tank and Hopfield (Hopfield and Tank 1985).
- Establish the approximate position of the phase transition temperature T_c in advance by estimating the eigenvalue spectrum of the equations linearized in the vicinity of the trivial fix-point (Peterson and Söderberg 1989).

In this paper we further develop the approach of (GPS 1989) to solve scheduling problems. Whereas the testbeds in that work were of a "synthetic" and simplified nature we here aim at solving a realistic problem: scheduling a Swedish high school. This requires further development of our approach in a number of directions. The paper is organized as follows. In the remainder of this section we review the results of (GPS 1989). A brief description of the structure of the Swedish high school curriculum and scheduling rules is also given (details can be found in Appendix B). Based on this testbed we give a list of requirements that the NN approach needs to fulfill. In Section 2 the NN approach of (GPS 1989) is further developed to meet these requirements. Section 3 contains prescriptions for how to deal with the soft constraints occurring in the problem, and in Section 4 we discuss the self-coupling terms needed to improve the dynamics. Issues related to the mean field dynamics are dealt with in Section 5, in particular phase transition properties of interacting Potts neurons with different number of components (details of this discussion can be found in Appendix A). A realistic problem is solved by the NN approach in Section 6. In Section 7 we briefly discuss other approaches for this kind of problem like linear programming. Finally in Section 8 a brief summary and outlook can be found.

1.1 Neural Networks and Scheduling - A Synthetic Example

In (GPS 1989) a simplified scheduling problem with the appropriate basic structure, where N_p teachers give N_q classes in N_x class rooms at N_t time slots, was mapped onto a Potts neural network. In this problem one wants solutions where all N_p teachers give a lecture to each of the N_q classes, using the available space-time slots with no conflicts in space (class rooms) or time. These are the **hard constraints** that have to be satisfied. In addition also **soft constraints** like continuity in class-rooms etc were considered. The basic entities of this problem can be represented by four sets consisting of N_p,N_q,N_x and N_t elements respectively.

There is a very transparent way to describe this problem that naturally lends itself to the Potts neural encoding of (Peterson and Söderberg 1989), where **events**, defined by teacher-class pairs (p, q), are mapped onto **space-time slots** (x, t). Multi-state (or Potts) neurons $S_{pq;xt}$ are defined to be 1 if the event (p, q) takes place in the space-time slot (x, t) and 0 otherwise.

The hard constraints in this picture are as follows:

1. An event (p, q) should occupy precisely one space-time slot (x, t).
2. Different events (p_1, q_1) and (p_2, q_2) should not occupy the same space-time slot (x, t).
3. A teacher p should have at most one class at a time.
4. A class q should have at most one teacher at a time.

A schedule fulfilling all the hard constraints is said to be **legal**. The first constraint can be imbedded in the neural network in terms of the Potts normalization condition

$$\sum_{x,t} S_{pq;xt} = 1 \tag{1}$$

for each event (p, q). In other words we have N_pN_q neurons, each of which has N_xN_t possible states . The other three constraints are implemented using energy penalty terms as follows:

$$E_{XT} = \frac{1}{2}\sum_{x,t}\sum_{p_1,q_1}\sum_{p_2,q_2} S_{p_1q_1;xt}S_{p_2q_2;xt} = \frac{1}{2}\sum_{x,t}[\sum_{pq} S_{pq;xt}]^2 \tag{2}$$

$$E_{PT} = \frac{1}{2}\sum_{p,t}\sum_{q_1,x_1}\sum_{q_2,x_2} S_{pq_1;x_1t}S_{pq_2;x_2t} = \frac{1}{2}\sum_{pt}[\sum_{qx} S_{pq;xt}]^2 \tag{3}$$

$$E_{QT} = \frac{1}{2} \sum_{q,t} \sum_{p_1,x_1} \sum_{p_2,x_2} S_{p_1q;x_1t} S_{p_2q;x_2t} = \frac{1}{2} \sum_{qt} [\sum_{px} S_{pq;xt}]^2 \tag{4}$$

This way of inplementing the constraints is by no means unique. In particular one could add a family of terms with no impact on the energy value that merely have the effect of adding a fixed constant to the energy. These extra terms turn out to have a strong impact on the mean field dynamics (see below).

In the next step **mean field** variables $V_{pq;xt} = <S_{pq;xt}>_T$ are introduced. The corresponding mean field equations at the **temperature** T, are given in terms of the **local fields** $U_{pq;xt}$,

$$U_{pq;xt} = -\frac{1}{T} \cdot \frac{\partial E}{\partial V_{pq;xt}} \tag{5}$$

as

$$V_{pq;xt} = \frac{e^{U_{pq;xt}}}{\sum_{x't'} e^{U_{pq;x't'}}}. \tag{6}$$

Eqs. (6) are then iterated with annealing, i.e. starting at a high temperature, and successively lowering it in the course of the process. The critical temperature T_c, which sets the scale of T, is estimated by expanding eqs. (6) around the trivial fix-point

$$V^{(0)}_{pq;xt} = \frac{1}{N_x N_t} \tag{7}$$

At sufficiently low temperatures the dynamics effectively becomes discrete, turning into a modified version of local optimization. In this regime it turns out to be advantageous to use **autobiased local optimization** by minimizing a modified energy

$$E_{eff} = E - E_\beta \tag{8}$$

where E_β adds a diagonal part to the energy expression. It amounts to a bias with respect to the present neuron state. These autobias terms in the energy correspond to the coupling of neurons to themselves, such that the continuity in (computer) time of the neurons is either rewarded or penalized, depending on the values of the connection strengths[1]. One effect of the autobias terms is that they affect the possibility of escaping from local minima.

If these terms are small enough as compared to the energy quantization scale (we will refer to this case as **low autobias**), we obtain a low temperature limiting behaviour similar to the case without autobias. The behaviour at a non-zero temperature will be different, however. Within the low-autobias region, one can choose the parameters such, that unwanted local minima become flattened out, while at the same time

[1]Autobias is not to be confused with what is normally called bias, and which corresponds to a connection to a permanently fixed, external neuron.

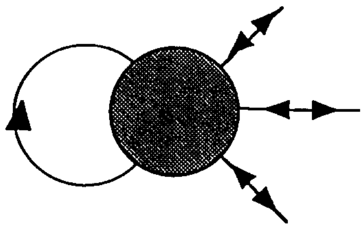

Figure 1: Schematic view of a neuron with a self-coupling corresponding to a diagonal term in the energy.

keeping the difference in critical temperatures between the different modes reasonably low.

The performance of this algorithm was investigated in (GPS 1989) for a variety of problem sizes and for different levels of difficulty, measured by the ratio between the number of events and the number of available space-time slots. It was found that the algorithm consistently found legal solutions with very modest convergence times for problem sizes $(N_p, N_q) = (5,5)$,...,$(12,12)$. With convergence time we here mean the total number of sweeps needed to obtain a legal solution no matter how many trials it takes. Also when introducing soft constraints like the preference of having subsequent lessons in the same room, very good solutions were obtained.

1.2 The Swedish High School System

We have chosen to use schedules inspired by the Swedish high school system as testbeds for two reasons. One reason is that we have easy access to data and the other and more important one is that this is a more complicated and nested problem than e.g. the corresponding US system and hence constitutes more of a challenge to the algorithm. The Swedish high school system we use is described in some detail in Appendix B. Let us here just sketch the main structure in order to see which extensions of the formalism of (GPS 1989) that are needed. To illuminate this structure it is instructive to compare with the more widely known US system.

In the **US high school system** the students are free to pick subjects for each semester subject to the constraint that the curriculum is going to be fulfilled in the end. The number of subjects chosen has to coincide with the number of hours per day. The schedule looks the same for students and teachers each day - one has a **day periodicity**. It also implies that "classes" are never formed like in an elementary school, where a set of equal-grade students continously have lessons together.

The **Swedish high school system** is very different. Basically the curriculum is implemented in the same way as in an elementary school in the sense that "classes" are formed that remain stable for all three years. Moreover the schedules look different from day to day - one has a **week periodicity**. Most subjects are compulsory, but not all. For optional subjects (in particular foreign languages) the proper classes are divided into option groups which subsequently recombine to form temporary classes. To get a feeling for the complexity of the problem we refer the reader to tables 2-5.

1.3 Needed Extensions of Formalism

The synthetic scheduling problem of (GPS 1989) contains several simplifications as compared to realistic problems and in particular when it comes the Swedish high school system. Here we give a list of items that an extended formalism will need to handle.

1. One week periodicity (occasionally extended to two- or four-week periodicity).
2. In (GPS 1989) each teacher has a class exactly once. In our case he has to give lessons in certain subjects a few hours a week, appropriately spread out.
3. In (GPS 1989) it was assumed that all class rooms were available for all subjects. This is not the case in reality. Many subjects require special purpose rooms.
4. Many subjects are taught for two hours in a row (double hours).
5. Group formation. For some optional subjects the classes are broken up into option groups temporarily forming new classes.
6. Certain preferences have to be taken into account, to meet e.g. special desires from teachers.

2 Neural Encoding

2.1 Factorization into x- and t-neurons

Prior to extending the formalism to cover points 1-6 there is an important simplification that can be made with the formalism used in the synthetic problem of (GPS 1989). It turns out that with the encoding $S_{pq;xt}$ the MFT-equations give rise to two phase transitions; one in x and one in t. In other words the system naturally factorizes into two parts. It is therefore economical to implement this factorization already at the encoding level. This is done by replacing $S_{pq;xt}$ by x-neurons $S^{(X)}_{pq;x}$ and t-neurons $S^{(T)}_{pq;t}$

$$S_{pq;xt} = S^{(X)}_{pq;x} S^{(T)}_{pq;t} \tag{9}$$

with separate Potts conditions replacing eq. (1)

$$\sum_x S^{(X)}_{pq;x} = 1 \tag{10}$$

and

$$\sum_t S^{(T)}_{pq;t} = 1 \tag{11}$$

respectively. This means that the number of degrees of freedom reduces from $N_pN_qN_xN_t$ to $N_pN_q(N_x + N_t)$. Redoing the problem of (GPS 1989) with this factorized formalism we find that the quality of the solutions does not deteriorate with this more economical way of encoding the problem. Needless to say the sequential execution time goes down. In what follows this factorized encoding will be used.

2.2 Extending the Formalism

When taking into account the points 1-4 above in our formalism one cannot keep p and q as the independent quantities. Rather we define an independent variable i (event index) to which p and q are attributes, $p(i)$ and $q(i)$. The values of these exist in a look-up table. This table contains all the necessary information to process each event i. The time index t we subdivide into weekdays (d) and daily hours (h). The Potts conditions (eqs. (10,11)) now read

$$\sum_x S^{(X)}_{i;x} = 1 \tag{12}$$

and

$$\sum_t S^{(T)}_{i;t} = 1 \tag{13}$$

The interpretation of $S^{(X[T])}_{i;x[t]}$ is of course the same as before - it is *one* if event i takes place in room x (at time t) and *zeroe* otherwise. The Potts neurons will have different number of components. We describe this by the matrices $C^{(T)}_{i;t}$ and $C^{(X)}_{i;x}$ defined as

$$C^{(T)}_{i;t} = \begin{cases} 1 & \text{if } t \text{ allowed for t-neuron } i \\ 0 & \text{if } t \text{ not allowed} \end{cases}$$

and

$$C^{(T)}_{i;x} = \begin{cases} 1 & \text{if } x \text{ allowed for x-neuron } i \\ 0 & \text{if } x \text{ not allowed} \end{cases}$$

To facilitate the handling of double hours we introduce effective t-neurons $\tilde{S}^{(T)}_{i;t}$ defined as

$$\tilde{S}^{(T)}_{i;t} = \sum_{k=0}^{g_i - 1} S^{(T)}_{i;t-k} \tag{14}$$

where the multiplicity g_i is *1* for single hours and *2* for double hours. With this notation the collision energies of eqs. (2,3,4) read[2]:

$$\begin{aligned} E_{XT} &= \frac{1}{2}\sum_{x,t}\sum_{i\neq i'} S^{(X)}_{i;x} S^{(X)}_{i';x} \tilde{S}^{(T)}_{i;t} \tilde{S}^{(T)}_{i';t} && (15) \\ E_{PT} &= \frac{1}{2}\sum_{p,t}\sum_{i\neq i'} \delta_{p(i),p(i')} \tilde{S}^{(T)}_{i;t} \tilde{S}^{(T)}_{i';t} && (16) \\ E_{QT} &= \frac{1}{2}\sum_{q,t}\sum_{i\neq i'} \delta_{q(i),q(i')} \tilde{S}^{(T)}_{i;t} \tilde{S}^{(T)}_{i';t} && (17) \end{aligned}$$

2.3 Periodicity

Most subjects have week periodicity. However, a few subjects with few hours per semester might occur with two- or four-week periodicity. This feature is easily implemented in our formalism.

2.4 Group Formation

So far the classes have been the fundamental objects. In order to account for point 5 above we need a formalism that allows for the breaking up of these primordial classes and subsequent recombinations into **quasi**-classes. This is a very common practice with e.g. foreign languages (see Appendix B) where the students have many choices. In order to be economical with resources, option groups with a particular choice from different primordial classes should form a temporary quasi-class (see fig. 2).

This complication can be accounted for by a minor modification of the E_{PT} collision term (eq. (17)) concerning the overlap between classes $q(i)$ and $q(i')$. We extend the possible class values $q(i)$ to include also quasi-classes. The Kronecker δ in eq. (17) ensures that only events i and i' with identical primordial classes are summed over. In the case of group formation into quasi-classes one might have contributing pairs of events where the primordial classes are different. Hence one should replace δ with

[2] Throughout this paper the notations like $i \neq i'$ means either a double sum or a single sum over i with fixed i' depending upon context.

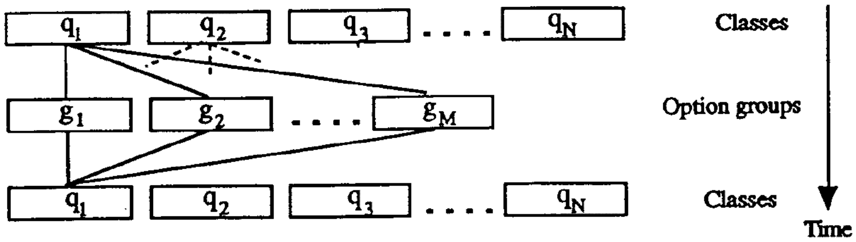

Figure 2: Formation of option groups and recombination into primordial classes

a more structured overlap matrix Γ which is given by a look-up table.

$$\delta_{q(i),q(i')} \rightarrow \Gamma_{q(i),q(i')} \tag{18}$$

2.5 Relative Clamping

In neural network vocabulary the phrase clamping is normally meant to imply that the equations of motion are settled with a few units not subject to the dynamics - they are clamped. Such phenomena will occur in our applications, in particular when it comes to revisions of existing schedules. What we are concerned about here is another kind of clamping when one wants a cluster of events to stick together in time - relative clamping. In our formalism this amounts to make the notational replacement

$$i \rightarrow (j, k) \tag{19}$$

where j denotes the cluster and k an event within the cluster. In this case one has a common t-neuron for the cluster ($S^{(T)}_{j;t}$) but distinct x-neurons ($S^{(X)}_{jk;x}$) for the individual events.

2.6 Null Attributes

Some activities lack teachers (e.g. lunch) and others involve no classes (e.g. teacher's conferences). It is convenient to include those events in our general formalism by letting $p(i) = 0$ or $q(i) = 0$. Such events of course do not contribute to E_{PT} and E_{QT} respectively.

2.7 Final Expressions of Collision Penalty Terms

We are now ready to write down the generalization of the collision terms of eqs. (15,16,17) that ensure that all the hard constraints are satisfied; the solutions are legal. In the next section we will then include penalty terms corresponding to the soft constraints.

$$\begin{aligned}
E_{XT} &= \frac{1}{2}\sum_{jk\neq j'k'}\sum_{x} S^{(X)}_{jk;x}S^{(X)}_{j'k';x}\sum_{t}\tilde{S}^{(T)}_{j;t}\tilde{S}^{(T)}_{j';t} \\
&= \frac{1}{2}\sum_{j\neq j'}\sum_{kk'}\sum_{x} S^{(X)}_{jk;x}S^{(X)}_{j'k';x}\sum_{t}\tilde{S}^{(T)}_{j;t}\tilde{S}^{(T)}_{j';t} + \frac{1}{2}\sum_{j} g_j \sum_{k\neq k'}\sum_{x} S^{(X)}_{jk;x}S^{(X)}_{jk';x} && (20) \\
E_{PT} &= \frac{1}{2}\sum_{pt}[\sum_{jk}\tilde{\delta}_{p,p(j,k)}\tilde{S}^{(T)}_{j;t}][\sum_{j'\neq j}\sum_{k'}\tilde{\delta}_{p,p(j',k')}\tilde{S}^{(T)}_{j';t}] \\
&= \frac{1}{2}\sum_{j\neq j'}\sum_{kk'}\tilde{\delta}_{p(j,k),p(j',k')}\sum_{t}\tilde{S}^{(T)}_{j;t}\tilde{S}^{(T)}_{j';t} && (21) \\
E_{QT} &= \frac{1}{2}\sum_{qt}[\sum_{jk}\delta_{q,q(j,k)}\tilde{S}^{(T)}_{j;t}][\sum_{j'\neq j}\sum_{k'}\Gamma_{q,q(j',k')}\tilde{S}^{(T)}_{j';t}] \\
&= \frac{1}{2}\sum_{j\neq j'}\sum_{kk'}\Gamma_{q(j,k),q(j',k')}\sum_{t}\tilde{S}^{(T)}_{j;t}\tilde{S}^{(T)}_{j';t} && (22)
\end{aligned}$$

In order to restrict the sum over p in eq. (21) to $p(j,k) \neq 0$ events we have introduced $\tilde{\delta}_{j,k}$ according to

$$\tilde{\delta}_{jk} = \delta_{j,k}(1 - \delta_{j,0}\delta_{k,0}) \qquad (23)$$

3 Soft Constraints

There are basically four kinds of soft constraints we encounter when scheduling Swedish high schools.

1. The different lessons for a class in a particular subject should be spread over the week such that they don't appear on the same day.
2. The lunch "lesson" has to appear within 3 hours around noon.
3. The schedules should have as few "holes" as possible; lessons should be glued together.
4. Teachers could have various individual preferences.

The second point is easily taken into account in the Potts condition for the relevant $p(j,k) = 0$ or $q(j,k) = 0$ events. Hence it is not formally a soft constraint. The individual preferences (point 4) will be omitted in our treatment due to lack of data. Here we will focus on **spreading** and **gluing**.

3.1 Spreading

First we assume that we have access to a subject attribute $s(j,k)$. We then introduce a penalty term E_{QSD} that spreads the lessons for a class in a particular subject over the different week-days.

$$
\begin{aligned}
E_{QSD} &= \frac{1}{2}\sum_{qsd}[\sum_{jk}\sum_{h}\delta_{q,q(j,k)}\delta_{s,s(j,k)}\tilde{S}^{(T)}_{jk;dh}]^2 \\
&= \frac{1}{2}\sum_{d}\sum_{j'\neq j}\sum_{kh}\sum_{k'h'}\delta_{q(j,k),q(j',k')}\delta_{s(j,k),s(j',k')}\tilde{S}^{(T)}_{j;dh}\tilde{S}^{(T)}_{j';dh} \\
&= \frac{1}{2}\sum_{j\neq j'}\sum_{kk'}\delta_{q(j,k),q(j',k')}\delta_{s(j,k),s(j',k')}\sum_{d}S^{(D)}_{j;d}S^{(D)}_{j';d} \qquad (24)
\end{aligned}
$$

In eq. (24) we have introduced an effective "day-neuron" according to

$$
S^{(D)}_{j;d} = \sum_{h}\tilde{S}^{(T)}_{j;dh} = g_j S^{(T)}_{j;dh} \qquad (25)
$$

3.2 Gluing

In order to avoid "holes" in time for a class we need a penalty term that rewards situations where (d,h)-events are glued to $(d,h-1)$-events. The following reward energy serves that purpose:

$$
\begin{aligned}
E_{QDH} &= -\delta\sum_{q}\sum_{d,h}[\sum_{j,k}\delta_{q,q(j,k)}\tilde{S}^{(T)}_{j;dh}][\sum_{j'\neq j,k'}\Gamma_{q,q(j',k')}\tilde{S}^{(T)}_{j';d(h-1)}] \\
&= -\delta\sum_{j'\neq j}\sum_{k,k'}\Gamma_{q(j,k),q(j',k')}\sum_{d,h>1}\tilde{S}^{(T)}_{j;dh}\tilde{S}^{(T)}_{j';d(h-1)} \qquad (26)
\end{aligned}
$$

In eq. (26) the parameter δ governs the relative strength of this reward relative to the collision energies (eqs. (20,21,22)).

4 Diagonal Terms

It is clear from eqs. (12,13) and the fact that $S_{i;x[t]}^{(X[T])}$ are either 0 or 1, that any partial sum X of neuronic components must also be 0 or 1, such that

$$X^2 = X \tag{27}$$

We can use the particular combinations $\sum_t S_{j;t}^{(T)^2} \equiv 1$ and $\sum_x S_{jk;x}^{(X)^2} \equiv 1$ to add trivial-valued auxiliary terms to the energy:

$$E_\beta = -\frac{\beta^{(T)}}{2}\sum_j\sum_t [S_{j;t}^{(T)}]^2 - \frac{\beta^{(X)}}{2}\sum_{jk}\sum_x [S_{jk;x}^{(X)}]^2 \tag{28}$$

These are the only non-trivial terms of this kind, which respect the obvious permutation symmetries of the problem. The effect of these extra terms on the energy is merely that of adding a fixed constant, but they will turn out to be important for the mean field dynamics. These diagonal terms correspond to self-coupling interactions (see fig. 1). Finding legal solutions with no collisions corresponds to minimizing the **hard energy**

$$E_{hard} = E_{XT} + E_{PT} + E_{QT} \tag{29}$$

Legal solutions are given by $E_{hard} = 0$. The **soft energy** is given by

$$E_{soft} = E_{QSD} + E_{QDH} \tag{30}$$

The total energy E to be minimized is then a sum of the hard and soft pieces:

$$E = E_{hard} + E_{soft} \tag{31}$$

5 Mean Field Dynamics

Now we introduce the continuous **mean field** variables $V_{jk;x}^{(X)}$ and $V_{jk;t}^{(T)}$, corresponding to $S_{jk;x}^{(X)}$ and $S_{jk;t}^{(T)}$ respectively. These have the interpretation of being probabilities that the corresponding events occur at given x- and t-values with normalizations given by eqs. (12, 13) replacing S by V. Substituting V for S in the energy expressions, the mean field equations at a finite **temperature** T, are given in terms of the **local fields** $U_{jk;x}^{(X)}$ and $U_{jk;t}^{(T)}$

$$U_{jk;x}^{(X)} = -\frac{1}{T}\cdot\frac{\partial E}{\partial V_{jk;x}^{(X)}} \tag{32}$$

$$U_{j;t}^{(T)} = -\frac{1}{T}\cdot\frac{\partial E}{\partial V_{j;t}^{(T)}} \tag{33}$$

as

$$V_{jk;x}^{(X)} = \frac{e^{U_{jk;x}^{(X)}}}{\sum_{x'} e^{U_{jk;x'}^{(X)}}}. \tag{34}$$

$$V_{j;t}^{(T)} = \frac{e^{U_{j;t}^{(T)}}}{\sum_{t'} e^{U_{j;t'}^{(T)}}}. \tag{35}$$

where it is understood that only allowed states are considered.

The natural mean field dynamics for this system consists of iterating eqs. (32-35). There are two main options for how to do this:

- **Synchronous updating:** A sweep consists of first computing all the local fields, and then updating all the neurons.
- **Serial updating:** Here, for each neuron $S_{jk;x[t]}^{(X[T])}$, the local field is computed immediately before the corresponding neuron state is updated. Thus, in this case, the sweep order might matter slightly for the outcome of the dynamics. We have consistently used ordered sweeps.

For both methods, in addition, the performance depends on the values of the β-parameters and T. The next problem is to understand their effect, and to give them suitable values.

5.1 Choice of parameters

In principle one could run the algorithm at a fixed temperature if a suitable value for the the latter is known in advance. Empirically, however, a more efficient way of obtaining good solutions turns out to be **annealing**, i.e. starting at a high temperature, and successively lowering it. One can get a good understanding of the parameter-dependence by considering the two limits:

- **High T.** At a sufficiently high temperature the system is attracted to a trivial fix-point. The behaviour of the system in this phase can be understood in terms of linearized dynamics in the vicinity of the fix-point.
- **Low T.** When the temperature tends to zero, the system enters another well-understood phase: the dynamics effectively becomes discrete, turning into a modified version of **local optimization**.

At intermediate temperatures, the dynamics is more complex – it is here that the actual computing is done. Loosely speaking, the smooth energy-landscape appearing at high T is gradually gaining structure, and the neurons are gradually forced into a decision. When $T \to 0$ the original discrete landscape is recovered, and the neurons get stuck in a firm decision, representing a local minimum of the energy.

Prior to investigating the two extreme T-regions we want to define **order parameters** that monitor the transitions from one phase to the other. As in (GPS 1989) we find it convenient to introduce t- and x-**saturations** for this purpose.

$$\Sigma_X = \frac{1}{N_X} \sum_{jk;x} [V^{(X)}_{jk;x}]^2 \tag{36}$$

and

$$\Sigma_T = \frac{1}{N_T} \sum_{j;dh} [V^{(T)}_{j;dh}]^2 \tag{37}$$

where N_X and N_T denote the total number of x- and t- neurons respectively (for the t-neurons of course only the clusters are counted). The $T \to 0$ limit is characterized by $\Sigma_X, \Sigma_T \to 1$.

5.2 The High Temperature Phase

As stated above, at high temperatures the system has an attractive trivial fix-point, and the high-T dynamics can be understood by linearizing it in the vicinity of this point.

In (GPS 1989) this fix-point was the totally symmetric one of eq. (7). In this case the situation is somewhat more complicated due to the lesser degree of symmetry.

At high temperatures, however, the trivial fix-point is well approximated by the symmetric point (corrections are of order $1/T$). For the linear fluctuations $v^{(X)}_{j;t}, v^{(T)}_{jk;x}$ around this point, the dynamics (eqs. (34,35)) is then replaced by the linearized equations (see Appendix A)

$$\begin{aligned} v^{(T)}_{j;t} &= \frac{1}{TK^{(T)}_j} \sum_{t'} Q^{(T)j}_{tt'} u^{(T)}_{j;t'} && (38) \\ v^{(X)}_{jk;t} &= \frac{1}{TK^{(X)}_{jk}} \sum_{x'} Q^{(X)jk}_{xx'} u^{(X)}_{jk;x'} && (39) \end{aligned}$$

Here, the Q-matrices are mere projections onto the locally allowed states and K_j (K_{jk}) is the number of allowed states for t-neuron j (x-neuron jk), and $u^{(X)}$ and $u^{(T)}$ are the linear fluctuations in the corresponding local fields,

$$u^{(T)}_{j;t} = \sum_{j',t'} A^{j';t'}_{j;t} v^{(T)}_{j';t'} + \sum_{j'l,x'} C^{j;t}_{j'l;x} v^{(X)}_{j'l;x} \tag{40}$$

$$u^{(X)}_{jk;x} = \sum_{j'l,x'} B^{j'l;x'}_{jk;x} v^{(X)}_{j'l;x'} + \sum_{j',t} C^{j';t}_{jk;x} v^{(T)}_{j';t} \tag{41}$$

where A, B and C are matrices resulting from linearizing eqs. (32,33). Due to the $1/T$ in eqs. (32,33), $v^{(X)}_{jk;x}$ and $v^{(T)}_{j;t}$ will tend to zero under iteration of these equations for high enough T, and the fix-point is stable. This defines the high-T phase. At a certain temperature T_c, the stability will be lost due to an eigenvalue crossing the unit circle. Empirically, for moderate parameter values the corresponding mode is dominantly a t-mode, which is not surprising, since the x-neurons appear in only one of the energy terms, whereas the t-neurons appear in all of them. Thus, to estimate T_c we can disregard the x-modes and we have the following equations for the t-modes:

$$u^{(T)}_{j;t} = \sum_{j',t'} A^{j';t'}_{j;t} v^{(T)}_{j';t'} \tag{42}$$

$$v^{(T)}_{j;t} = \frac{1}{TK^{(T)}_j} \sum_{t'} Q^{(T)j}_{tt'} u^{(T)}_{j;t'} \tag{43}$$

or, with matrix notation in Potts space,

$$\mathbf{v}^{(T)}_j = \frac{1}{TK^{(T)}_j} \mathbf{Q}^{(T)j} \sum_{j'} \mathbf{A}^{j'}_j \mathbf{v}^{(T)}_{j'} \tag{44}$$

For synchronous updating, an update sweep consists of iterating this matrix equation, and since the matrix is proportional to $1/T$, the computation of T_c is reduced to the computation of the eigenvalues of a fix matrix. The case of sequential updating is a little more tricky. A generic discussion of the high-T phase-transition for both types of updating can be found in Appendix A.

The point we want to make here is that, for both methods of updating, the critical temperature can be estimated in advance. It will of course depend on the self-coupling $\beta^{(T)}$ (and in principle also on $\beta^{(X)}$). This dependence will typically be as follows (cf. also fig. 3):

- For large negative $\beta = \beta^{(T)}$, $T_c(\beta)$ has a negative slope ($\approx -1/K^{(T)}_{eff}$, where $K^{(T)}_{eff}$ is an effective Potts-dimension) and the relevant eigenvalue at T_c is -1, which is disastrous for the stability of the dynamics, since it gives rise to oscillating behaviour.
- For sequential updating, there follows a region, still with negative slope, with complex critical eigenvalues (in the synchronous case this is ruled out, since the relevant matrix can be made Hermitian).
- Finally, for large enough β the slope is positive ($\approx 1/K^{(T)}_{eff}$) and the relevant eigenvalue is $+1$. This is the desired case, but it is difficult to obtain with synchronous updating, and we have therefore consistently used sequential updating, where it can be obtained even with a small negative self-coupling.

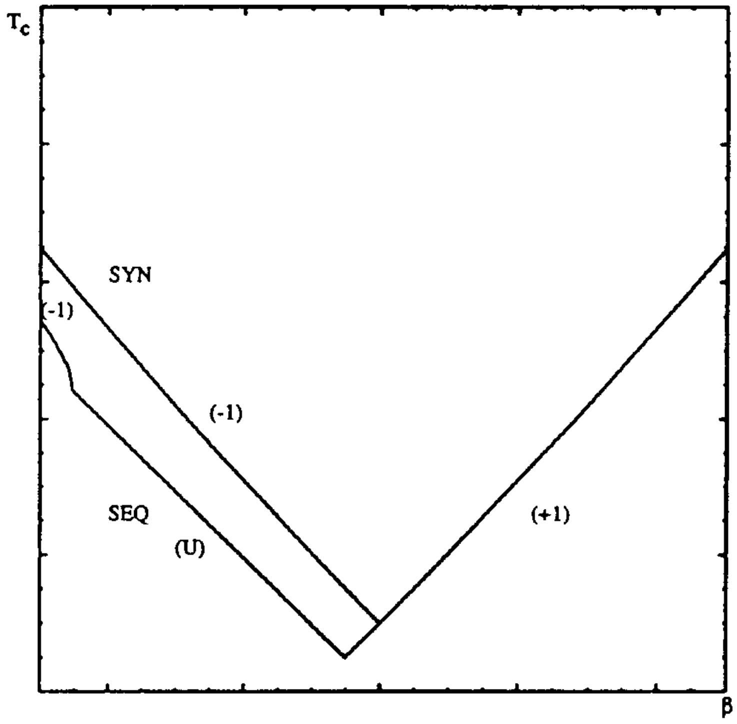

Figure 3: The dependence of T_c on the selfcoupling β for synchronous (SYN) and sequential (SEQ) updating. The type of eigenvalue is indicated by (±1) or (U) (for complex unitary).

5.3 The Low Temperature Limit

When $T \rightarrow 0$, the dynamics effectively turns discrete (winner-takes-all), and is in the absence of self-coupling equivalent to simple **local optimization** of the discrete energy of eq. (31). With a non-zero self-coupling, we obtain as mentioned in the introduction auto-biased local optimization: staying in the present state is rewarded or penalized, depending on the sign of the self-coupling parameter β. If the self-coupling is positive, there is an increased tendency to stay in the present state, leading to an increased stability. With a negative self-coupling, the effect is the opposite: the stability of the present state is decreased. In both cases, if the self-coupling is smaller than the quantization scale of the energy, it has no effect in practice at small T.

6 Solving a Realistic Problem

The entire process of scheduling high school activities in a given geographical area is very complex. With existing manual or semi-manual procedures it is always factorized down to a few independent steps.

- Concentrate certain majors and special subjects (e.g. languages) to certain schools. Assign the students to these schools based on their choices of majors and options (see Appendix B). Form the classes and option groups at the different schools accordingly.
- Given their competence assign teachers to the different classes and groups.
- Given the conditions generated by the two steps above solve the scheduling problem.

It is the last step that is the subject of this work. We have used data from the Tensta high school (see Appendix B) as a test problem for our algorithm. Since a complete set of consistent data has not been available to us we have chosen to generate legal schedules based on available class and teacher data. For each time-slot in a week approximately N_q events are generated in the following way[3]: For each event a teacher, a subject and a classroom category are randomly generated, with the constraint that no collisions occur. Two class-room categories are assumed to exist, small and large, to host entire classes and minor groups respectively. These two categories are generated with suitable relative probability. Similarly single and double hour lessons are generated with equal probability. We have generated 10 different problems in this way. One might object that this is not an entirely real problem and hence not relevant for testing the algorithm. We think the situation is the other way around - our generated problems are very likely far more difficult since there are no preset ties between teachers, classes and class-rooms that makes the problem more structured in terms of lesser "effective" degrees of freedom.

Our algorithm is implemented in the following "black box" manner in the case of sequential updating:

- Choose a problem.
- Set $\delta = 0.2$, $\beta^{(X)} = -0.1$ and $\beta^{(T)} = -0.1$ repectively. The results are fairly insensitive to these choices.
- Determine T_c by iterating the linearized dynamics (eq. (38)).

[3] When consistent lists will become available to us we of course intend to use real data.

slot	*M*	*T*	*W*	*T*	*F*	*M*	*T*	*W*	*T*	*F*	*M*	*T*	*W*	*T*	*F*	*M*	*T*	*W*	*T*	*F*
1		**W**					**W**					**B**					**B**			
2		C					C				**X**	C				**A**	C			
3		E	**F**		G		E	**Y**		G	**D**	E	**F**	**G**	G	**D**	E	**Y**		G
4	*	E	H	G	*	*	E	H	G	*	*	E	H	G	*	*	E	H	G	*
5	I	*	*	*	J	I	*	*	*	J	I	*	*	*	J	I	*	*	*	J
6	K	L	**M**	M	O	K	L	**M**	N	O	K	L		N	O	K	L		N	O
7		**K**		**Q**		**C**	**K**	**P**		**R**	**C**			**Q**	**R**	**C**	**K**		**Q**	
8	S		D	**Q**		S		D		**F**	S		D	**Q**	**F**	S		D	**Q**	
9	S		T			S		T		**D**	S		T		**Q**	S		T		
10	U		V			U		V			U		V			U		V		

Table 1: Four-week schedule for E2b. Different letters stand for different subjects; lunch is denoted by an asterisk. Lessons which differ from week to week are given in boldface.

- Initialize with $V^{(X)}_{jk;x} = C^{(X)}_{jk;x}/K^{(X)}_{jk}(1 + 0.001 \times rand[-1, 1])$ and $V^{(T)}_{j;t} = C^{(T)}_{j;t}/K^{(T)}_{j}(1 + 0.001 \times rand[-1, 1])$ respectively.

- Anneal with $T_n = 0.9 \times T_{n-1}$ until $\Sigma = 0.9$.

- At each T_n perform one update per neuronic component (= one sweep) with sequential updating using eqs. (32-35).

- After $\Sigma = 0.9$ is reached check whether the obtained solutions are legal, i.e. $E_{hard} = 0$ (eq. (29)). If this is not the case the network is initialized with a different seed and is allowed to resettle. This procedure is repeated until a legal solution is found.

We have used this procedure for 10 generated sets of data as described above. In table. 1 we show a typical evolution of E_{hard}, Σ_T, Σ_T and T as a function of N_{sweep}. For the 10 experiments performed the average N_{sweep} needed to reach a legal solution was ≈ 70. We did not observed a single case where more than one annealing was needed to obtain a $E_{hard} = 0$ solution.

Although there is no objective measure of the quality of solution, in the opinion of individuals acquainted with the scheduling process, all the network solutions are comparable to or better than solutions that would be obtained with the semi-manual or manual procedures currently in use. A typical example is shown in table 1, where we note how well the algorithm has glued the lessons together - very few holes. Another prominent feature is how well the algorithm handles the fact that events occur with different periodicity.

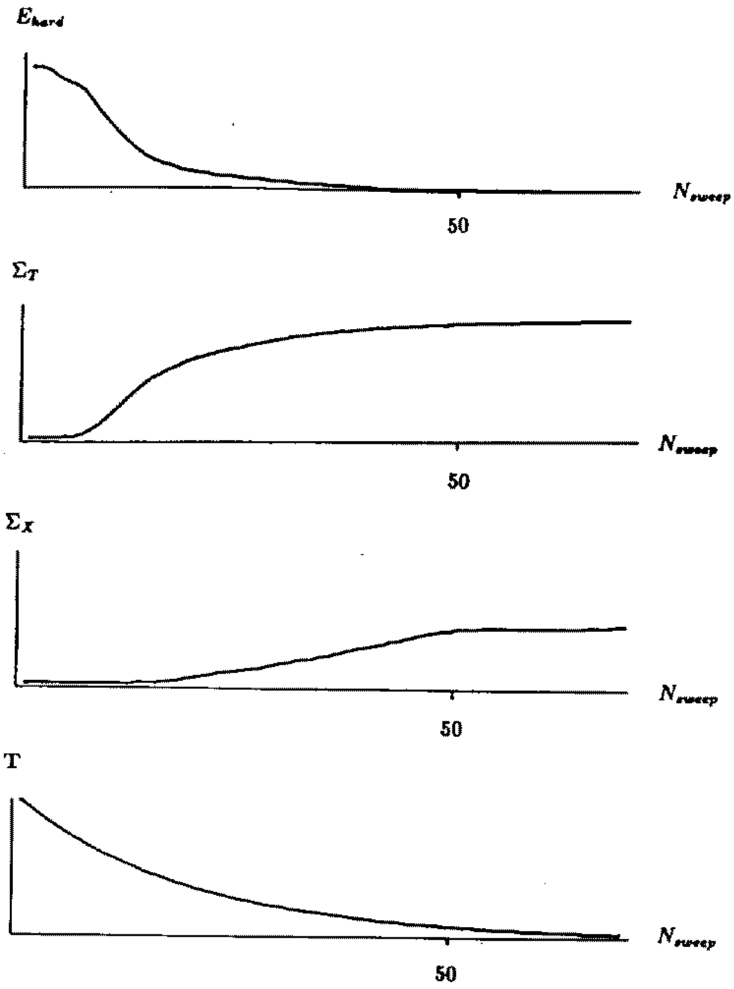

Figure 4: Energy (E_{hard}), saturations (Σ_T, Σ_X) and temperature (T) as functions of N_{sweep} for one run.
is indicated.

The algorithm was implemented in F77 on an APOLLO DN 10000 workstation. It takes approximatively one hour to perform the required 70 sweeps.

7 Comparison with Other Approaches

What other algorithms do exist which have been pursued for the kind of realistic problems dealt with in this paper? An extensive search in the operations reasearch literature have not yielded any result with respect to algorithms that deal with scheduling problems of our size and complexity. This is not to say that there exist no computerized tools for scheduling in the educational sector. However, all those programs known to the authors represent refinements to the manual method with book-keeping capabilities for the manual decisions made. Since these are not problem solvers in the real sense we have not found it meaningful to compare the results from our algorithm with procedures using such packages. For sub-problems with less complexity there exists in the literature an automated approach for classroom allocation using linear programming (Gosselin and Truchon 1986). This is a simpler problem since it does not contain the time-tabling part. We have not had access to the test bed of this simpler problem. It is nevertheless interesting to compare the algorithmic complexity of the neural approach versus the linear programming one for this problem.

Denote the total number of rooms and and time slots with N_X and N_T respectively. In (Gosselin and Truchon 1986) the rooms are categorized, so another important quantity is the number of room categories N_C. The computational load (n_c) with linear programming should then scale as

$$n_c \propto N_C^3 N_X^2 N_T^3 \tag{45}$$

One can also convince oneself that with the Potts encoding and MFT techniques used in this paper the corresponding scaling relation is (with M_X the average number of room alternatives per event)

$$n_c \propto M_X N_X N_T \tag{46}$$

where we have assumed that the number of iterations needed for the MFT equations is constant with problem size. This latter empirical fact seems to be realized whenever the location of T_c is estimated in advance using the linearized MFT equations as is done in this paper and in (Peterson 1990). As can be seen from eqs. (45,46) the MFT neural approach scales better in N_X and N_T for this problem. The same relative advantage for the neural approach of course holds for the problem dealt with in this paper - that is the reason the neural approach is able to solve a problem of this complexity.

8 Summary and Outlook

In this paper we have applied a neural network technique to find good solutions to a difficult scheduling problem with very good performance results both with respect to quality and CPU time consumption. Even with sequential updating of the neurons the algorithm can be efficiently executed on a SIMD computer like CRAY by parallelizing the calculation of the local fields (eq. (33,32)). Speedup factors proportional to N_x (N_t) can then be obtained, which in our case corresponds to 50-60. This amounts to an execution time of approximately *one minute* on a CRAY XMP.

Indeed, such gains were realized when executing the problem of (GPS 1989) on a CRAY XMP.

As compared with previous work (GPS 1989) we have developed the formalism in the following directions:

1. Factorization into x- and t-neurons (eq. (9)).
2. Analysis of the dynamics in the critical T-region also in the case where the neurons have different Potts dimensions.
3. Extension of the formalism to deal with group formation, double hours, week periodicity (sometimes broken into 4-week periodicity), relative clamping and spreading. All of these features are necessary for solving Swedish high school problems.

Also, the realistic problems we are considering constitute a substantially larger problem size than dealt with in (GPS 1989). With the approximately 90 teachers, 50 weekly hours, 45 classes and 60 class-rooms the problems corresponds to approximately 10^{4600} possible choices. In our factorized Potts formulation this corresponds to roughly 10^5 neuronic degrees of freedom.

Empirically we have found that the number of sweeps needed to produce legal solutions increases only very slowly with the problem size N_q, implying that the CPU time consumption scales approximately like N_q^2.

A revision capability is inherent in our formalism which is handy in situations when encountering unplanned events once a schedule exists. Such rescheduling is performed by clamping those events not subject to change and heating up and cooling the remaining dynamical neurons.

One should keep in mind that problems of this kind and size are so complex that even several man-months of human planning will in general not yield solutions that will meet all the requirements in an optimal way. We have not been able to find any algorithm in the literature, that solves a realistic problem with this complexity. Existing commercial software packages do not solve the entire problem. Rather the problem is solved with an interactive user taking step-wise decisions. Linear programming methods have only been applied to the simpler problem of classroom allocation only. These methods scale with problem size and complexity in such a way that it is very hard to deal with the kind of problem dealt with in this work.

In cases where the dimensions of the optimization problem due to geometrical nature can be dimensionally reduced (e.g. the traveling salesman problem) it is advantegous to abandon the Potts probabilistic description to a template (elastic net (Durbin and Willshaw 1987)) formulation for reducing the number of degrees of freedom. No such reduction of number of degrees of freedom is possible in scheduling problems like the one studied here.

The scheduling of a high school is of course only one example of a difficult planning problem; there are many similar problems occuring in the industry and in the public sector. We have chosen this particular application since we feel that the problem is representative enough for this class of problems and also because real data were available to us. In one respect this problem is simple - it has no topological structure as in e.g. airline crew scheduling. We are presently extending our formalism to cover such cases (Gislén, Peterson and Söderberg, in progress).

Acknowledgements:

We would like to thank L. Jörnstad for providing us with data from the Tensta high school.

Appendix A

The High Temperature Fixed Point

For the sake of notational simplicity (and generality) the arguments in this section are carried through with neurons S_{ia} subject to the Potts condition

$$\sum_a S_{ia} = 1 \tag{A1}$$

Consider an energy with a general form of the interactions T_{ij}^{ab}

$$E = \frac{1}{2} \sum_{i \neq j} \sum_{ab} T_{ij}^{ab} S_{ia} S_{jb} \tag{A2}$$

Replacing S_{ia} in eq. (A2) with the corresponding MFT variables V_{ia}, the local fields $U_{ia} = -1/T \; \partial E / \partial V_{ia}$ are given by

$$U_{ia} = \frac{1}{T} [\beta V_{ia} - \sum_{j \neq i} \sum_b T_{ij}^{ab} V_{ib}] \tag{A3}$$

where we have inserted a self-coupling β. Introducing K_i as the number of allowed states for neuron i, and the matrix C_{ia} as in Section 2.2 the MFT equations read

$$V_{ia} = \frac{C_{ia} e^{U_{ia}}}{\sum_b C_{ib} e^{U_{ib}}}. \tag{A4}$$

At high enough temperature T the dynamics expressed by eqs. (A3,A4) will have an attractive trivial fix-point $V_{ia}^{(0)}$, close to the symmetry point

$$V_{ia}^{(s)} = \frac{C_{ia}}{K_i} \tag{A5}$$

where every allowed state is equally probable. At a certain critical temperature T_c the trivial fix-point will become unstable. In order to find the position of this phase-transition, we have to linearize the dynamics in the neighbourhood of the fix-point. In terms of the deviation $v_{ia} = V_{ia} - V_{ia}^{(0)}$, the linearized dynamics takes the form

$$v_{ia} = V_{ia}^{(0)} u_{ia} - V_{ia}^{(0)} \sum_b V_{ib}^{(0)} u_{ib}, \tag{A6}$$

with u_{ia} given by

$$u_{ia} = \frac{1}{T} [\beta v_{ia} - \sum_{j \neq i} \sum_b T_{ij}^{ab} v_{ib}]. \tag{A7}$$

Approximating $V_{ia}^{(0)}$ by $V_{ia}^{(s)}$, eq. (A6) can be written as

$$v_{ia} = \frac{1}{K_i} \sum_b Q_{iab} u_{ib}, \tag{A8}$$

where Q_i is a mere projection on the locally transverse (i.e. $\sum_a v_{ia} = 0$) allowed subspace, with the dimension $\sum_i (K_i - 1)$:

$$Q_{iab} = C_{ia}\delta_{ab} - \frac{1}{K_i} C_{ia} C_{ib}. \tag{A9}$$

Thus, if we define M_{ij}^{ab} as the projection of T_{ij}^{ab} on this subspace,

$$M_{ij}^{ab} = \sum_{cd} Q_{iac} T_{ij}^{cd} Q_{idb} \tag{A10}$$

we obtain for the dynamics in the subspace

$$v_{ia} = \frac{1}{K_i T}[\beta v_{ia} - \sum_{j \neq i} \sum_b M_{ij}^{ab} v_{jb}]. \tag{A11}$$

This can be symmetrized in terms of $\phi_{ia} = \sqrt{K_i} v_{ia}$:

$$\phi_{ia} = \frac{1}{T} \sum_{jb} \tilde{M}_{ij}^{ab} \phi_{jb}. \tag{A12}$$

with

$$\tilde{M}_{ij}^{ab} = \frac{1}{\sqrt{K_i}}[\beta \delta_{ij}\delta_{ab} - M_{ij}^{ab}] \frac{1}{\sqrt{K_j}}$$

Above T_c all the eigenvalues of the corresponding sweep update matrix are less than unity in magnitude, and T_c is obviously the highest T allowing a unitary eigenvalue. It will depend, though, on how the updating is done.

For **synchronous updating**, the sweep update matrix is simply $\tilde{M}/T$, and the only possible eigenvalues are ± 1, since $\tilde{M}$ is real and symmetric. We then obtain, in terms of the extreme eigenvalues λ_{max} and λ_{min} of the matrix $\tilde{M}$,

$$T_c = \max(-\lambda_{min}, \lambda_{max}) \tag{A13}$$

and the corresponding dominant eigenvalue is -1 for β below a certain value, and $+1$ above this value.

For ordered **sequential updating**, things are a little more involved. When computing the new value of a neuron, fresh values are used for the already updated neurons. The sweep update matrix is then somewhat more complicated, but can still be expressed in terms of the diagonal (β) part $\tilde{M}_D$, the upper ($i < j$) part $\tilde{M}_U$ and the lower part $\tilde{M}_L$ of $\tilde{M}$ as

$$(T\mathbf{1} - \tilde{M}_L)^{-1}(\tilde{M}_D + \tilde{M}_U) \tag{A14}$$

and T_c in this case is obtained by examining when this matrix has an eigenvalue of modulus one. In this case the eigenvalues do not have to be ± 1, but these values play a special role also here. Assuming an eigenvector ϕ with a unitary eigenvalue $e^{2i\theta}$, we have

$$(e^{i\theta}T - e^{-i\theta}M_D)\phi = (e^{i\theta}M_L + e^{-i\theta}M_U)\phi, \tag{A15}$$

where the total matrix on the right side is Hermitian. Note that with eigenvalue +1, the result $T_c = \lambda_{max}$ for synchronous updating is reproduced.

For other eigenvalues, we can get some information by taking the scalar product with $\phi^\dagger$. Assuming ϕ to be normalized, we obtain

$$e^{i\theta}T - e^{-i\theta}\beta < 1/K >= \phi^\dagger(e^{i\theta}M_L + e^{-i\theta}M_U)\phi, \tag{A16}$$

where <1/K> appears because of the special form of the diagonal term in eq. (A3).Since the right member of eq. (A6) is obviously real, the imaginary part of the left member must vanish:

$$\sin\theta(T + \beta < 1/K >) = 0 \tag{A17}$$

For eigenvalues $\neq 1$, this implies

$$T = -\beta < 1/K > . \tag{A18}$$

We conclude that eigenvalue 1 is the only possibility for non-negative β.

Empirically, the dominant eigenvalue is -1 for large enough negative β, complex unitary for intermediate β values, and +1 for large enough β values (including positive β).

In the special case where all the K_i are the same, the eigenvalue -1 appears for sequential updating only in the limit $\beta \to \infty$. In this case the β dependence of T_c is also particularly simple (Peterson and Söderberg 1989). It is piecewise linear for both updating methods.

Appendix B

The Swedish High School System

The Swedish high school system differs strongly from that of the US. Prior to entering the high school the students decide upon a certain **major**. These majors could be either theoretical or more trade-oriented. The theoretical majors take 3 years to complete whereas some of the more practically oriented majors take 2 years. Within each major some subjects are optional, in particular foreign languages. Apart from that the students follow each other in classes throughout the high school. As mentioned in the text the schedules are essentially periodic over weeks (and not days as in the US system). All in all there are some 20 different possible majors. However, for a typical high school only 8-10 of these are available. In the particular high school we use as a test-bed the following 8 majors are present:

N	=	Natural Sciences	(3 years)	T	=	Technology	(4 years)
S	=	Social Studies	(3 years)	So	=	Social Studies	(2 years)
FA	=	Fine Arts	(3 years)	HC	=	Health Care	(2 years)
E	=	Economics	(3 years)	M	=	Merchandize &	
C	=	Commercial	(3 years)			Consumer Techn.	(2 years)

The corresponding curriculae are given in tables 2,3 and 4. A few explanations:

All numbers refer to weekly hours. Fractional weekly hours like 2.5 could be implemented by having for example 2 hours in the fall and 3 hours in the spring. Another option is to have alternating weekly schedules. In elementary school it is compulsory for the students to take Swedish and English plus one additional foreign language (B-language). When entering high school this B-language can either be pursued further or be replaced by a new foreign language (C-language). In some majors both B/C-language and an additional C-language is chosen. B/C-langauges could be French, German, Spanish, Russian etc. varying from school to school.

The Test-bed Problem

The realistic problem we have considered was obtained from *Tensta High School* in terms of number of classes for the different majors. More specifically this high school has the classes shown in table 5, which roughly corresponds to 1000 students.

Subject	N			S			FA		
	yr 1	yr 2	yr 3	yr 1	yr 2	yr 3	yr 1	yr 2	yr 3
Swedish	3	3	3.5	3	3	3.5	3	3	3.5
English	3	3	3/0[1]	3[3]	3[3]	3	3	3	3[4]
B/C-language	3/4	3	3/0[1]	3/4[3]	3[3]	3	3/4	3	3[4]
C-language				3[3]	3[3]	3	4	3	3[4]
Linguistics								3	
Greek								6[4]	
History	2	2	2/0[1]	2	5	2.5	2	5	2.5
Religion			2			3			3
Philosophy			2/0[2]			2			2
Psychology			0/2[2]			2			2
Civics	2	2.5		3	2.5	6	3	2.5	6[4]
Social Science				3[3]	3[3]				
Mathematics	6	5	4	5	3	3	5		
Natural Sciences				4	4		4	4	
Physics	2.5	4	4						
Chemistry	3.5	3	2						
Biology		2	3						
History of Music and Arts		2/0[1]			2			2	
Arts/Music	2			2			2		
Special subject		0/3[1]	0/5[1]						
Physical Education	3	2.5	2	3	2.5	2	3	2.5	2
Project			2			2			2

Table 2: Curriculum for Natural Sciences (N), Social Studies (S) and Fine Arts (FA) majors. **(1)** In year 2 or 3 a special subject can be chosen. In that case History of Music and Arts is dropped in year 2. In year 3 the B/C language or English is dropped together with history. **(2)** Choice between Philosophy and Psychology in year 3. **(3)** Two of these subject are chosen. **(4)** Greek is optional in year 3, in which case Civics and two out of three languages are dropped.

Subject	E			C	T			
	yr 1	yr 2	yr 3	yr 1 - yr 3	yr 1	yr 2	yr 3	yr 4
Swedish	3	3.5	3	7	3	3	2.5	
English	3	3	3	6	3	3		
B/C-language	3/4[1]	3[2]	3[2]		3	3/4	3	
C-language	4[1]	3[2]	3[2]					
History	2	2			2		2	
Religion			2				2	
Psychology			2	2				
Civics	3	2.5	2	5	2	2.5		
Mathematics	5	3[2]	3[2]	6	6	5	4	
Physics					2.5	4	4	
Chemistry					3.5	3		
Biology						2		
Technology					5	4.5		
Misc. Technical Subjects							12	30
Industrial Practice					3	3		
Natural Sciences	4[1]							
Business Economy	4	6		14				2-4
Administration			2					
Law			2					
Marketing				14				
Methodology		3						
Ergonomics								2-3
Typing	2	3						
Shorthand		3[2]	3[2]					
Information and Text Processing				16				
Communication				2				
Special Subject				28				
Physical Education	3	2	2	4	3	2	2	
Project			2					

Table 3: Curriculum for Economics (E), Commercial (C) and Technology (T) majors. **(1)** Two of these subjects should be chosen. **(1)** Two of these subjects should be chosen, of which at least one should be B- or C-language. The distribution of weekly hours between the different years may vary locally for the Commercial major.

Subject	So		HC		M	
	yr 1	yr 2	yr 1	yr 2	yr 1	yr 2
Swedish	4	3	4	3	6	6
English	3	3			3	3
B/C-language						
History	3	2				
Religion		2				
Psychology			1	2		
Ergonomics						
Civics	3	3	3	3	2	2
Social Science		5			5	5
Mathematics	3	3				
Natural Sciences	6	6		2		
Administration					2	2
Health Care			2	2		
Child and Youth Care			12	11		
Health Care Practice			7	7		
Arts/Music	2^1	2^1	3	2	1.5	
Typing	4				4	4
Goods handling	4				8	8
Consumer knowledge	4				3	3
Special Subject	3	3	3	3		
Physical Education	3	2	2	2	2	2
Project			2			2

Table 4: Curriculum for Social Studies [2 year] (So), Health Care (HC) and Goods and Merchandize and Consumer Technology (M) majors. **(1)** One of these subjects should be chosen.

Major	year 1	year 2	year 3
N	N1a,N1b	N2a,N2b	N3a,N3b
S	S1	S2	S3
H	H1	H2	H3
E	E1a,E1b	E2a,E2b	E3a,E3b
C	C1a,C1b	C2a,C2b	C3
T	T1	T2	
So	So1a,So1b	So2a,So2b	
HC	HC1a,HC1b	HC2	
M	M		

Table 5: Tensta high school classes in 1990. In cases with more than class per major and year the notation is $a, b, \ldots$

References

[1] C. Peterson, "Parallel Distributed Approaches to Combinatorial Optimization", *Neural Computation* **2**, 261 (1990).

[2] L. Gislén, C. Peterson and B. Söderberg, "Teachers and Classes with Neural Networks", *International Journal of Neural Systems* **1**, 167 (1989).

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

[4] C. Peterson and B. Söderberg, "A New Method for Mapping Optimization Problems onto Neural Networks", *International Journal of Neural Systems* **1**, 3 (1989).

[5] R. Durbin and G. Willshaw, "An Analogue Approach to the Travelling Salesman Problem Using an Elastic Net Method", *Nature* **326**, 689 (1987).

[6] L. Gislén, C. Peterson and B. Söderberg, work in progress.

[7] K. Gosselin and M. Truchon, "Allocation of Classrooms by Linear Programming", *Journal of Operational Research Society* **37**, 561 (1986).