Convergence Properties of the Softassign Quadratic Assignment Algorithm

Anand Rangarajan
Department of Diagnostic Radiology
Yale University
anand@noodle.med.yale.edu

Alan Yuille
Smith-Kettlewell Eye Research Institute
San Francisco, CA
yuille@skivs.ski.org

Eric Mjolsness
Dept. of Computer Science and Engineering
University of California San Diego
emj@cs.ucsd.edu

Abstract

The softassign quadratic assignment algoithm is a discrete time, continuous state, synchronous updating optimizing neural network. While its effectiveness has been shown in the traveling salesman problem, graph matching and graph partitioning in thousands of simulations, there was no associated study of its convergence properties. Here, we construct discrete time Lyapunov functions for the cases of exact and approximate doubly stochastic constraint satisfaction which can be used to show convergence to a fixed point. The combination of good convergence properties and experimental success make the softassign algorithm the technique of choice for neural QAP optimization.

1 Introduction

Discrete time optimizing neural networks are a well honed topic in neural computation. Beginning with the discrete state Hopfield model (Hopfield, 1982), considerable effort has been spent in analyzing the convergence properties of discrete time networks, especially along the dimensions of continuous versus discrete state and synchronous versus sequential update (Hopfield, 1984; Peterson and Soderberg, 1989; Fogelman-Soulie et al., 1989; Marcus and Westervelt, 1989; Blum and Wang, 1992; Waugh and Westervelt, 1993; Koiran, 1994; Wang et al., 1996). Relative to their continuous time counterparts, discrete time optimizing networks can be more easily implemented on digital computers; a temporal step size parameter is needed for implementing continuous time networks. However, this advantage is partially offset by the problem of constraint satisfaction in discrete time networks. Continuous time Lyapunov functions have been shown to exist for quadratic assignment (QAP) optimizing networks (Gee et al., 1993; Gee and Prager, 1994; Wolfe et al., 1994; Yuille and Kosowsky, 1994; Urahama, 1996) but not for their discrete time counterparts. Quadratic assignment networks are important not only because they subsume the traveling salesman problem (TSP) (Peterson and Soderberg, 1989), graph partitioning (Van den Bout and Miller III, 1990; Peterson and Soderberg, 1989), graph isomorphism (Rangarajan et al., 1996; Simic, 1990) and graph matching (Simic, 1990; Gold and Rangarajan, 1996) but because they embody doubly stochastic constraint satisfaction.

The softassign quadratic assignment algorithm is a discrete time, continuous state, synchronous updating neural network. Despite embodying a fairly complex doubly stochastic constraint satisfaction subnetwork, it is in the same lineage as earlier discrete time optimizing neural networks. While its effectiveness has been shown in QAP problems like TSP, graph partitioning and graph matching (Gold and Rangarajan, 1995; Rangarajan et al., 1996; Gold and Rangarajan, 1996) and linear problems like point matching (Rangarajan et al., 1997a) and linear assignment (Kosowsky and Yuille, 1994), the existence of a discrete time Lyapunov function has never been shown, until now.

In this paper, we demonstrate the existence of a discrete time Lyapunov function for the softassign quadratic assignment neural ntwork. We begin in Section 3.1 by first considering the simpler case of exact doubly stochastic constraint satisfaction. This directly leads to a general discrete time Lyapunov function broadly applicable to any choice of the neuronal activation function. In contrast, in Section 3.2 we show that for the case of approximate doubly stochastic constraint satisfaction, a discrete time Lyapunov function can be easily constructed only for the exponential neuronal activation function.

2 The Softassign Quadratic Assignment Algorithm

The quadratic assignment problem (QAP) is stated as follows:

$$\min_M E_{\mathrm{qap}}(M) = -\frac{1}{2}\sum_{aibj} \hat{C}_{ai;bj} M_{ai} M_{bj} + \sum_{ai} A_{ai} M_{ai}$$
$$\text{subject to } \sum_a M_{ai} = 1,\ \sum_i M_{ai} = 1,\ \text{and } M_{ai} \in \{0,1\}. \tag{1}$$

In (1), $\hat{C}$ is the quadratic assignment benefit matrix, A is the linear assignment benefit matrix and M is the desired $N \times N$ permutation matrix.

When the binary constraint is relaxed to a positivity constraint, M becomes doubly stochastic:

$$\min_M E_{\mathrm{dsqap}}(M) = -\frac{1}{2}\sum_{aibj} \hat{C}_{ai;bj} M_{ai} M_{bj} + \sum_{ai} A_{ai} M_{ai}$$
$$\text{subject to } \sum_a M_{ai} = 1,\ \sum_i M_{ai} = 1,\ \text{and } M_{ai} > 0. \tag{2}$$

As it stands, minimizing (2) over the space of doubly stochastic matrices will not yield a permutation matrix. However, (Yuille and Kosowsky, 1994) have shown that if $\hat{C}$ is positive definite (when rewritten as a 2-D matrix), then the minima of (2) will be permutations.

Since $\hat{C}$ is specified by the problem, it is unlikely to be positive definite. One way to fix this is by adding the term $\frac{\gamma}{2}\sum_{ai} M_{ai}(1 - M_{ai})$ to the objective function in (2). Since $\sum_{ai} M_{ai} = N$, this is equivalent to adding a *self-amplification* term (Rangarajan et al., 1996; von der Malsburg, 1990) $-\frac{\gamma}{2}\sum_{ai} M_{ai}^2$ to the QAP objective function. Adding the self-amplification term is equivalent to defining a new benefit matrix $C_{aibj} \stackrel{\mathrm{def}}{=} \hat{C}_{aibj} + \gamma\delta_{ab}\delta_{ij}$. For a given $\hat{C}$, there exists a lower bound for the self-amplification parameter γ

which makes the newly defined benefit matrix C positive definite. Henceforth, we refer to C as the QAP benefit matrix with the understanding that its eigenvalues can be easily shifted by changing the value of γ.

The softassign quadratic assignment algorithm is a discrete-time, synchronous updating dynamical system (Rangarajan et al., 1996). It combines deterministic annealing, self-amplification and the softassign and is based on minimizing the following objective function (Rangarajan et al., 1996; Yuille and Kosowsky, 1994):

$$\begin{aligned} E_{\text{saqap}}(M,\mu,\nu) = &-\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai} M_{bj} + \frac{1}{\beta}\sum_{ai} M_{ai}\log M_{ai} \\ &+ \sum_a \mu_a(\sum_i M_{ai} - 1) + \sum_i \nu_i(\sum_a M_{ai} - 1) \end{aligned} \tag{3}$$

This form of the energy function has two Lagrange parameters μ and ν for constraint satisfaction, an $x\log x$ *entropy* barrier function (Luenberger, 1984) which ensures positivity of $\{M_{ai}\}$ and the deterministic annealing inverse temperature parameter β. An annealing schedule is typically prescribed for the β. The $x \log x$ entropy term is somewhat different from the traditional barrier functions used in nonlinear optimization in that it does not tend to ∞ as we approach the boundary $x = 0$. As we shall see, this does not prevent it from being a useful "barrier" function.

The QAP benefit matrix C is preset based on the chosen problem, for example, graph matching, TSP or graph partitioning and subsequently modified in a restricted manner (as indicated earlier) by self-amplification. Handling the graph partitioning *multiple membership* constraint requires a slight modification in the above objective function. In all problems, we assume that $\{C_{ai;bj}\}$ is symmetric, i.e. $C_{ai;bj} = C_{bj;ai}$. Our notation is summarized in Table 1.

To derive the softassign quadratic assignment algorithm, we first apply the following algebraic transformation (Mjolsness and Garrett, 1990) to the quadratic term:

$$-\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai} M_{bj} = \min_{\sigma} -\sum_{aibj} C_{ai;bj} M_{ai}\sigma_{bj} + \frac{1}{2}\sum_{aibj} C_{ai;bj}\sigma_{ai}\sigma_{bj} \tag{4}$$

Since C is positive definite, the right side of (4) has a unique minimum w.r.t. σ, namely $\sigma_{ai} = M_{ai},\ \forall a, i$. With the algebraic transformation in place, the modified energy function is

$$\begin{aligned} E_{\text{saqap}}(M,\sigma,\mu,\nu) = &-\sum_{aibj} C_{ai;bj} M_{ai}\sigma_{bj} + \frac{1}{2}\sum_{aibj} C_{ai;bj}\sigma_{ai}\sigma_{bj} + \sum_{ai} A_{ai} M_{ai} \\ &+ \sum_a \mu_a(\sum_i M_{ai} - 1) + \sum_i \nu_i(\sum_a M_{ai} - 1) + \frac{1}{\beta}\sum_{ai} M_{ai}\log M_{ai} \end{aligned} \tag{5}$$

Application of the algebraic transformation results in an energy function that is linear in M but quadratic in σ. However, the minimum w.r.t. σ is already known ($\sigma = M$). Upon alternately minimizing (5) w.r.t. σ and M, while keeping the Lagrange parameter vectors μ and ν fixed, we get

$$\begin{aligned} \sigma_{ai} &= M_{ai} \\ M_{ai} &= \exp\left[\beta\left(\sum_{bj} C_{ai;bj}\sigma_{bj} - A_{ai} - \mu_a - \nu_i\right) - 1\right] \end{aligned} \tag{6}$$

Table 1: Variables and constants in the softassign QAP dynamical system

N	QAP problem size
$\{C_{ai;bj}\}$	QAP quadratic benefit matrix
$\{A_{ai}\}$	QAP linear benefit matrix
β	inverse temperature
$\{M_{ai}^{(n)}\}$	match matrix at the nth step
$\{B_{ai}^{(n)}\}$	effective assignment matrix defined as $\{\sum_{bj} C_{ai;bj} M_{bj}^{(n-1)} - A_{ai}\}$
$\{\mu_a^{(n)}\}$	row constraint Lagrange parameter at the nth step
$\{\nu_i^{(n)}\}$	column constraint Lagrange parameter at the nth step
ϵ	row constraint convergence criterion $\lvert \sum_i M_{ai} - 1 \rvert < \epsilon,\ \forall a \in \{1, \ldots, N\}$
$\{\Delta M_{ai}\}$	match matrix difference defined as $\{M_{ai}^{(n+1)} - M_{ai}^{(n)}\}$
δ	overall convergence criterion at each temperature $\sqrt{\sum_{ai} \Delta M_{ai}^2} \leq \delta,\ \forall a, i \in \{1, \ldots, N\}$
$\{\Delta\mu_a\}$	row constraint Lagrange parameter difference defined as $\{\mu_a^{(n+1)} - \mu_a^{(n)}\}$
$\{\Delta\nu_i\}$	column constraint Lagrange parameter difference defined as $\{\nu_i^{(n+1)} - \nu_i^{(n)}\}$
$\{\mu\mathrm{max}_a\}$	upper bound on the absolute value of $\{\mu_a\}$
ΔL	discrete-time Lyapunov energy difference
λ	smallest eigenvalue of $\{C_{ai;bj}\}$ in the subspace of the column constraint

From (6), we identify σ with the value of M at the previous step to get the following discrete-time synchornous updating dynamical system:

$$M_{ai}^{(n+1)} = \exp\left[\beta\left(B_{ai}^{(n+1)} - \mu_a - \nu_i\right) - 1\right] \tag{7}$$

where

$$B_{ai}^{(n+1)} \stackrel{\text{def}}{=} \sum_{bj} C_{ai;bj} M_{bj}^{(n)} - A_{ai}$$

The main intuition as to why the above alternating sequence of updates lowers the overall energy is by analogy to the Expectation–Maximization (EM) algorithm (Dempster et al., 1977; Jordan and Jacobs, 1994). Later on, we rigorously demonstrate the existence of a Lyapunov function for the above sequence of updates.

From (7), we see that the $x \log x$ barrier function leads to an exponential neuronal activation function. We have not yet specified the Lagrange parameters μ and ν in (7). *At each iteration n of the discrete-time dynamical system in (7), we have to satisfy the row and column constraints on M.* These are (exactly or approximately) satisfied by solving for the Lagrange parameters μ and ν. It is possible to specify a separate constraint satisfaction energy function for the Lagrange parameters. At this juncture, we cannot overemphasize the point that when constraint satisfaction is undertaken, the matrix $B^{(n)}$ and time step n are held fixed. Substituting (6) in (5), using the definition of B above and reversing the sign (because solving for the Lagrange parameters is a maximization and not a minimization problem), we get a new energy function to be minimized w.r.t. the Lagrange parameters μ and ν.

$$E_{\text{lag}}^{(n)}(\mu,\nu) = \frac{1}{\beta}\sum_{ai}\exp\left[\beta\left(B_{ai}^{(n)} - \mu_a - \nu_i\right) - 1\right] + \sum_a \mu_a + \sum_i \nu_i \tag{8}$$

Taking derivatives w.r.t. μ and ν, we obtain the following fixed point relations:

$$\begin{aligned}
\frac{\partial E^{(n)}}{\partial \mu_a} = 0 \quad &\Rightarrow \quad 1 - \sum_i \exp\left[\beta\left(B_{ai}^{(n)} - \mu_a - \nu_i\right) - 1\right] = 0 \\
&\Rightarrow \quad \sum_i M_{ai}^{(n)} = 1 \\
\frac{\partial E^{(n)}}{\partial \nu_i} = 0 \quad &\Rightarrow \quad 1 - \sum_a \exp\left[\beta\left(B_{ai}^{(n)} - \mu_a - \nu_i\right) - 1\right] = 0 \\
&\Rightarrow \quad \sum_a M_{ai}^{(n)} = 1
\end{aligned} \tag{9}$$

It is not possible to simultaneously solve in closed form for the two Lagrange parameters μ and ν. However, from (9), an alternating minimization algorithm can be obtained. We first assume initial condition vectors μ^0 and ν^0 (with $B^{(n)}$ held fixed for the moment). Then we specify an alternating sequence of updates $\mu^{(k)}$ and $\nu^{(k)}$. With this specification and from (9), we get:

$$\begin{aligned}
\exp\left(\beta\mu_a^{(k+1)}\right) &= \sum_i \exp\left[\beta\left(B_{ai}^{(n)} - \nu_i^{(k)}\right) - 1\right] \\
\exp\left(\beta\nu_i^{(k)}\right) &= \sum_a \exp\left[\beta\left(B_{ai}^{(n)} - \mu_a^{(k)}\right) - 1\right]
\end{aligned} \tag{10}$$

Corresponding to the alternating sequence of updates of the Lagrange parameters, we may define an odd-even sequence of updates of the matrix M. The odd part fixes the rows of M while the even part fixes the columns:

$$\begin{aligned}
M_{ai}^{(n,2k-1)} &= \exp\left[\beta\left(B_{ai}^{(n)} - \mu_a^{(k)} - \nu_i^{(k-1)}\right) - 1\right] \\
M_{ai}^{(n,2k)} &= \exp\left[\beta\left(B_{ai}^{(n)} - \mu_a^{(k)} - \nu_i^{(k)}\right) - 1\right]
\end{aligned} \tag{11}$$

From (10) and (11, we see that

$$\frac{M_{ai}^{(n,2k-1)}}{M_{ai}^{(n,2k)}} = \exp\left[\beta\left(\nu_i^{(k)} - \nu_i^{(k-1)}\right)\right] = \sum_b M_{bi}^{(n,2k-1)} \tag{12}$$

Similarly, for the row constraint, we get

$$\frac{M_{ai}^{(n,2k)}}{M_{ai}^{(n,2k+1)}} = \exp\left[\beta\left(\mu_a^{(k+1)} - \mu_a^{(k)}\right)\right] = \sum_j M_{aj}^{(n,2k)} \tag{13}$$

From (12) and (13), we may write

$$M_{ai}^{(n,2k)} = \frac{M_{ai}^{(n,2k-1)}}{\sum_a M_{ai}^{(n,2k-1)}}, \text{ and } M_{ai}^{(n,2k+1)} = \frac{M_{ai}^{(n,2k)}}{\sum_i M_{ai}^{(n,2k)}} \tag{14}$$

The alternating sequence of normalizations in (14) is identical to Sinkhorn's theorem (also called Sinkhorn balancing) (Sinkhorn, 1964): "a doubly stochastic matrix can be obtained from any positive square matrix by the simple process of alternating row and column normalizations." Again note the two time indices in $M_{ai}^{(n,2k)}$. With the outer time index n (and consequently $B^{(n)}$) held fixed, we apply Sinkhorn balancing. During Sinkhorn balancing, M changes in order to satisfy the row and column constraints but $B^{(n)}$ does not. Note that the Lagrange parameters are set to some suitable initial values $\mu^{(0)}$ and $\nu^{(0)}$ *each time* we enter the Sinkhorn balancing loop. Having completely specified the softassign QAP dynamical system, we now write down its pseudocode:

The Softassign Quadratic Assignment Algorithm

Initialize β to β_i, $M_{ai}^{(0,0)}$ to $\frac{1}{N} + \xi_{ai}$

Begin A: Deterministic Annealing. Do A until $\beta \geq \beta_f$

Begin B: Relaxation. Do B until all M_{ai} converge.

$B_{ai}^{(n)} \leftarrow \sum_{bj} C_{ai;bj} M_{bj}^{(n-1)} - A_{ai}$

Begin Softassign:

Initialize the Lagrange parameters to $\mu^{(0)}$ and $\nu^{(0)}$

$M_{ai}^{(n,0)} \leftarrow \exp\left[\beta\left(B_{ai}^{(n)} - \mu_a^{(0)} - \nu^{(0)}\right) - 1\right]$

Begin C: Sinkhorn. Do C until all M_{ai} converge

Update $M^{(n,\cdot)}$ by normalizing the columns:

$M_{ai}^{(n,2k)} \leftarrow \frac{M_{ai}^{(n,2k-1)}}{\sum_a M_{ai}^{(n,2k-1)}}$

Update $M^{(n,\cdot)}$ by normalizing the rows:

$M_{ai}^{(n,2k+1)} \leftarrow \frac{M_{ai}^{(n,2k)}}{\sum_i M_{ai}^{(n,2k)}}$

End C

End Softassign

End B

$\beta \leftarrow \beta\beta_r$

End A

Having specified the softassign QAP algorithm, natural questions that arise at this juncture are: i) Does the dynamical system in (7) converge to a fixed point at each setting of β? And if so, (ii) how does the convergence criterion for the Sinkhorn balancing procedure affect the convergence properties of the overall dynamical system at each temperature? Next, we present our answers to these questions.

3 Convergence Properties

Recall from the previous section that minimizing (2) over the space of doubly stochastic matrices yields a permutation matrix provided C is positive definite. Consequently, convergence of the overall algorithm to a permutation matrix also crucially depends on minimizing the energy function in (3). The main issue then is whether the discrete-time dynamical system in (7) along with solving for the Lagrange parameters μ and ν converges to a fixed-point at each value of the barrier function parameter β.

We have not emphasized or motivated the choice of the $x \log x$ entropy barrier function (or the associated exponential neuronal activation function). While the entropy barrier function can be independently motivated from statistical physics considerations (Yuille and Kosowsky, 1994; Rangarajan et al., 1996), it turns out to play a central role in the convergence properties analyzed here. Since the $x \log x$ function is not privileged from a barrier function perspective (Luenberger, 1984), we first develop the analysis of the convergence properties using a general barrier function. As we shall see, the choice of the barrier function and the convergence assumptions made regarding Sinkhorn balancing are connected in a fundamental way. It turns out that the key assumption separating general barrier functions from the entropy barrier function is the convergence of Sinkhorn balancing to a doubly stochastic matrix. If Sinkhorn returns a doubly stochastic matrix, a very general analysis in terms of an unspecified barrier function can be carried out. If Sinkhorn returns a matrix that is merely close to being doubly stochastic, not only is the analysis more complex, but even designing an algorithm equivalent to Sinkhorn for constraint satisfaction is more complex. For instance, if we used a barrier function $\phi(x) = -\log x$ (a very popular choice in interior point methods), an analytical solution for the Lagrange parameters turns out to be very hard to derive (if not impossible). Accordingly, in Section 3.1, we begin by assuming that Sinkhorn always returns a doubly stochastic matrix. Then in Section 3.2, this assumption is relaxed and an analysis when Sinkhorn approximately converges is carried out.

3.1 Exact convergence of Sinkhorn

Examples of barrier functions $\phi(x)$ other than the $x \log x$ function are $\phi(x) = -\log x,\ \frac{1}{x},\ -\frac{1}{x^2},\ x\log x + (1-x)\log(1-x)$ etc. Barrier functions and barrier function control parameters (β) are inseparable (Luenberger, 1984); for any barrier function, an annealing schedule has to be prescribed for β. We now rewrite the QAP energy function (3) for a general barrier function. We also assume exact constraint satisfaction:

$$\begin{aligned} E_{\text{saqap}}(M, \mu^*, \nu^*) = {} & -\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai} M_{bj} + \sum_{ai} A_{ai} M_{ai} \\ & + \sum_a \mu_a^* (\sum_i M_{ai} - 1) + \sum_i \nu_i^* (\sum_a M_{ai} - 1) + \frac{1}{\beta}\sum_{ai} \phi(M_{ai}) \end{aligned} \tag{15}$$

where $\phi(x)$ is some suitable barrier function. For every barrier function $\phi(x)$ the corresponding neuronal activation function is $x = (\phi')^{-1}(y)$ provided the relevant derivatives and inverses exist. That we are assuming exact constraint satisfaction is flagged by the use of μ^* and ν^*.

We repeat the derivation in Section 2 of the discrete-time softassignquadratic assignment algorithm. We begin with the objective function (3). Next, we introduce an algebraic transformation (and a new variable σ) in (4) to get a new objective function shown in (5). This is minimized w.r.t. M and σ to yield the alternating update equation (6). Finally, σ is shown to be the value of M at the previous time step from which we obtain the discrete-time synchronous update equation (7). After replacing the entropy barrier function with the more general barrier function $\phi(x)$ and exactly repeating the aforementioned steps, we obtain the discrete-time synchronous update dynamical system corresponding to a general barrier function ϕ:

$$M_{ai}^{(n+1)} = (\phi')^{-1}\left[\beta(\sum_{bj} C_{aibj} M_{bj}^{(n)} - A_{ai} + \mu_a + \nu_i)\right] \text{ or}$$
$$\frac{1}{\beta}\phi'(M_{ai}^{(n+1)}) = \sum_{bj} C_{aibj} M_{bj}^{(n)} - A_{ai} + \mu_a + \nu_i \tag{16}$$

For most choices of the barrier function $\phi(x)$ other than $x \log x$, $(\phi')^{-1}$ has an unpleasant form making it difficult to solve for the Lagrange parameters μ and ν. This is why we assume exact constraint satisfaction. In (16), we see that the barrier function $\phi(x)$ has lead to the corresponding $(\phi')^{-1}$ neuronal activation function. As mentioned above, instead of solving for the Lagrange parameter vectors μ and ν, we use Sinkhorn's theorem instead to ensure that the row and column constraints are satisfied. A deeper assumption lurks in (16). While the exponential activation update equation for the entropy barrier function shown in (7) trivially satisfies the positivity constraint (on M_{ai}), the same is not true for general barrier functions. The positivity constraint has to be separately checked for each barrier function. In this section, we bypass these potential problems by assuming exact constraint satisfaction—of positivity and the row and column constraints. This assumption of exact constraint satisfaction will be relaxed later when we specialize to the $x \log x$ barrier function.

From our assumption of exact convergence of Sinkhorn, it follows that the Lagrange parameter vectors μ and ν can be dropped from the energy function in (15). This is tantamount to assuming that M is restricted to always being doubly stochastic. (We assume that the positivity constraint is always satisfied.) After dropping the terms involving the Lagrange parameters, we write down the new energy function. (Since this new energy function turns out to be a suitable discrete-time Lyapunov energy function, we modify our notation somewhat:)

$$L(M) = -\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai} M_{bj} + \sum_{ai} A_{ai} M_{ai} + \frac{1}{\beta}\sum_{ai}\phi(M_{ai}) \tag{17}$$

With the energy function (17) and the discrete-time synchronous updating dynamical system (16) equation (at each temperature) corresponding to exact constraint satsifaction in place, we can state the following theorem:

Theorem 1 Given that the barrier function $\phi(x)$ is convex and constraint satisfaction is exact, at each temperature, the energy function specified in (17) is a discrete-time Lyapunov function for the discrete-time synchronous update dynamical system specified in (16) (provided the Lagrange parameters are specified such that the row and column constraints are satisfied).

Proof: We need to show that the change in energy from step n to step $(n+1)$ is greater than zero. (We do not need to make a separate argument for $n = 1$.) The change in energy

$$\Delta L \stackrel{\text{def}}{=} L(M^{(n)}) - L(M^{(n+1)}) = -\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai}^{(n)} M_{bj}^{(n)} + \sum_{ai} A_{ai} M_{ai}^{(n)}$$
$$+\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai}^{(n+1)} M_{bj}^{(n+1)} - \sum_{ai} A_{ai} M_{ai}^{(n+1)} + \frac{1}{\beta}\sum_{ai}\phi(M_{ai}^{(n)}) - \frac{1}{\beta}\sum_{ai}\phi(M_{ai}^{(n+1)}) \tag{18}$$

If the function $\phi(x)$ is convex in $\mathcal{R}^1$, then

$$\phi(y) - \phi(x) \geq \phi^{'}(x)(y - x)$$

Using this, the change in energy is rewritten as

$$\Delta L \geq \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} + \sum_{aibj} C_{aibj} M_{bj}^{(n)}\Delta M_{ai} - \sum_{ai} A_{ai}\Delta M_{ai} - \frac{1}{\beta}\sum_{ai}\Delta M_{ai}\phi'(M_{ai}^{(n+1)}) \tag{19}$$

Substituting (16) in (19), we get

$$\Delta L \geq \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} + \sum_{ai}\mu_a\Delta M_{ai} + \sum_{ai}\nu_i\Delta M_{ai} \tag{20}$$

Since constraint satisfaction is exact, this reduces to

$$\Delta L \geq \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} > 0$$

due to the positive definiteness of C.
$\square$

After examining the proof, it should be clear that global positive definiteness of C is a stronger condition than required for the energy function in (17) to be a discrete-time Lyapunov function. It is sufficient for C to be positive definite in the linear subspace spanned by the row and column constraints.

To summarize, we have shown that a Lyapunov function exists for the fairly general discrete-time dynamical system in (16). The two main assumptions are: i) a convex barrier function $\phi(x)$ and (ii) exact constraint satisfaction.

3.2 Approximate convergence of Sinkhorn

We cannot always assume that Sinkhorn balancing yields a doubly stochastic matrix. In practice, the softassign is stopped after a suitable convergence criterion is met. Without loss of generality, we may only consider the situation when the column constraint is exactly satisfied ($\sum_a M_{ai} = 1$) and the row constraint is merely approximately satisfied; ($|\sum_i M_{ai} - 1| < \epsilon$) where ϵ is a small positive quantity.

In Section 3.2.1, we analyze the convergence properties of the softassign QAP algorithm when Sinkhorn only approximately converges. The analysis is carried out for the entropy barrier function alone. We think it is difficult to analyze the general case. To demonstrate this, we write down the energy difference for a general barrier function and for the case when the column constraint is exactly satisfied and the row constraint approximately satisfied. This is done by substituting (16) in the energy difference formula (18) with the column constraint exactly satisfied:

$$\Delta L = \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} + \sum_{ai}\mu_a \Delta M_{ai}$$
$$+\frac{1}{\beta}\sum_{ai}\left[\phi(M_{ai}^{(n)}) - \phi(M_{ai}^{(n+1)}) + \Delta M_{ai}\phi'(M_{ai}^{(n+1)})\right] \quad (21)$$

The first term and the third term are positive due to the positive definiteness of C and the convexity of ϕ respectively. However, analyzing the properties of the Lagrange parameter vector μ for a general barrier function turn out to be quite intricate and involved. In contrast, bounds on the Lagrange parameters can be easily derived for the entropy barrier function as shown in Appendix A. It may be possible to repeat this analysis for other specific barrier functions. From this point on, we focus almost exclusively on the entropy barrier function.

3.2.1 A Lyapunov function for the entropy barrier function

Theorem 2 At each temperature, the energy function

$$L(M) = -\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai} M_{bj} + \sum_{ai} A_{ai} M_{ai} + \frac{1}{\beta}\sum_{ai} M_{ai}\log M_{ai} \quad (22)$$

is a discrete-time Lyapunov function for the discrete-time synchronous updating dynamical system

$$M_{ai}^{(n+1)} = \exp\left[\beta\left(B_{ai}^{(n+1)} - \mu_a - \nu_i\right) - 1\right] \quad (23)$$

provided the following conditions hold:

(i) The column constraint $\sum_a M_{ai} = 1$ is exactly satisfied.

(ii) The row constraint is approximately satisfied: $|\sum_i M_{ai} - 1| < \epsilon,\ \forall a$ and $\epsilon > 0$.

(iii) The QAP benefit matrix is strictly positive definite with its smallest eigenvalue $\lambda > 0$.

(iv) The convergence criterion at each temperature is $\sqrt{\frac{\sum_{ai}\Delta M_{ai}^2}{N^2}} \leq \delta$ where $\delta > 2\sqrt{\frac{\epsilon\left[\sum_j \max_{a,b,c,i}\left(C_{ai;cj} - C_{bi;cj}\right) + \max_{a,b,i}(A_{ai} - A_{bi}) + \frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}\right]}{\lambda N}}$.

Proof: The change in energy is

$$\Delta L \stackrel{\text{def}}{=} L(M^{(n)}) - L(M^{(n+1)}) =$$
$$-\frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai}^{(n)} M_{bj}^{(n)} + \sum_{ai} A_{ai} M_{ai}^{(n)} + \frac{1}{2}\sum_{aibj} C_{ai;bj} M_{ai}^{(n+1)} M_{bj}^{(n+1)} - \sum_{ai} A_{ai} M_{ai}^{(n+1)}$$
$$+\frac{1}{\beta}\sum_{ai} M_{ai}^{(n)}\log M_{ai}^{(n)} - \frac{1}{\beta}\sum_{ai} M_{ai}^{(n+1)}\log M_{ai}^{(n+1)} \quad (24)$$

which can be rewritten as

$$\Delta L = \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} + \sum_{aibj} C_{aibj} M_{bj}^{(n)}\Delta M_{ai} - \sum_{ai} A_{ai}\Delta M_{ai}$$
$$+\frac{1}{\beta}\sum_{ai} M_{ai}^{(n)} \log \frac{M_{ai}^{(n)}}{M_{ai}^{(n+1)}} - \frac{1}{\beta}\sum_{ai}\Delta M_{ai}\log M_{ai}^{(n+1)} \tag{25}$$

From (23), we may write

$$\frac{1}{\beta}\log M_{ai}^{(n+1)} = \sum_{bj} C_{ai;bj} M_{bj}^{(n)} - A_{ai} - \mu_a - \nu_i - \frac{1}{\beta} \tag{26}$$

This results in a further simplification of the Lyapunov energy difference

$$\Delta L = \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} + \frac{1}{\beta}\sum_{ai}\left(M_{ai}^{(n)}\log\frac{M_{ai}^{(n)}}{M_{ai}^{(n+1)}} - M_{ai}^{(n)} + M_{ai}^{(n+1)}\right)$$
$$+\sum_{ai}\mu_a\Delta M_{ai} + \sum_{ai}\nu_i\Delta M_{ai} \tag{27}$$

When the column constraint $\sum_a M_{ai} = 1$ is kept continuously satisfied (at each Sinkhorn iteration), further simplifications can be made:

$$\Delta L = \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} + \frac{1}{\beta}\sum_{ai} M_{ai}^{(n)}\log\frac{M_{ai}^{(n)}}{M_{ai}^{(n+1)}} + \sum_{ai}\mu_a\Delta M_{ai} \tag{28}$$

using the relation $\sum_a M_{ai} = 1$.

For convergence, we require the discrete-time Lyapunov energy difference to be greater than zero. The first term is strictly positive if $C_{ai;bj}$ is positive definite in the subspace spanned by the column constraint $\sum_a M_{ai} = 1$. The second term in (28) is greater than or equal to zero by the non-negativity of the Kullback-Leibler measure. However, the third term can be positive or negative. By controlling the degree of positive definiteness of the QAP benefit matrix $C_{ai;bj}$, we can ensure that the overall energy difference is positive until convergence. Since we have specified in Section 2 a lower bound λ on the eigenvalues of C, this can be achieved.

We require an upper bound on the absolute value of the third term in (28). Using the row constraint convergence criterion $|\sum_i M_{ai}^{(n)} - 1| < \epsilon$, we can derive an upper bound for each $|\mu_a|$. This derivation can be found in Appendix A:

$$|\mu_a| \leq \mu\max \stackrel{\text{def}}{=} \sum_j \max_{a,b,c,i}\left(C_{ai;cj} - C_{bi;cj}\right) + \max_{a,b,i}\left(A_{ai} - A_{bi}\right) + \frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}. \tag{29}$$

Assuming an overall convergence criterion $\sqrt{\frac{\sum_{ai}\Delta M_{ai}^2}{N^2}} < \delta$ at each temperature, we get by only considering the first and third terms in (28):

$$\begin{aligned}\Delta L &\geq \frac{\lambda N^2\delta^2}{2} - 2N\epsilon\mu\max \\ &\geq 0 \ \text{ provided } \ \delta \geq 2\sqrt{\frac{\epsilon\mu\max}{\lambda N}}\end{aligned} \tag{30}$$

When we substitute the value of μmax from above, we get

$$\delta > 2\sqrt{\frac{\epsilon\left[\sum_j \max_{a,b,c,i}\left(C_{ai;cj} - C_{bi;cj}\right) + \max_{a,b,i}\left(A_{ai} - A_{bi}\right) + \frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}\right]}{\lambda N}}$$

□

One consequence of Theorem 2 is the loss of independence between the constraint satisfaction threshold parameter ϵ and the match matrix convergence (at each temperature) threshold parameter δ. Given ϵ, there exists a lower bound on δ given by condition (iv) above. This lower bound is approximately (constant $\cdot \sqrt{\frac{\epsilon}{\lambda N}}$). Since the smallest eigenvalue λ can be easily shifted (by changing the self-amplification parameter γ), a value of δ that is appropriate for each problem instance can be chosen. In Section 4, we present an example where all the conditions in Theorem 2 are meaningfully met.

4 Experiments

In all experiments, the QAP benefit matrix C was set in the following manner; $C_{ai;bj} = G_{ab}g_{ij}$ or $C = g \otimes G$. This particular decomposition of C is useful since it permits a straightforward manipulation of the eigenspectra of C in the row or column subspaces. Since the linear benefit matrix A does not add any insight into the convergence properties, we set it to zero.

First, we demonstrate the insight gained from Theorem 1. To this end, we generated a quadratic benefit matrix C which was not positive definite. We separately generated the matrices G and g using $\frac{N(N-1)}{2}$ normal (with mean zero and variance 1) random numbers. Since the matrices are symmetric, this completely specifies C. First, we shifted the eigenvalues of G and g such that the eigenvalue spectrum was roughly centered around zero. Then we ran the softassign QAP algorithm. The energy difference shown in Figure 1 is computed using (28) with the Lagrange parameter energy term set to zero. After some transient fluctuations (which are negative as well as positive), the energy difference settles into a limit cycle of length 2. Next, we made G and g positive definite by shifting the spectra upward. (We did not further refine the experiment by making G and g positive definite in the subspaces of the column and row constraints respectively.) After recomputing C, we reran the softassign QAP algorithm. Once again, we used (28) to compute the energy difference at each iteration (shown on the right in Figure 1). As expected, the energy difference is always greater than zero. We have demonstrated that a positive definite C leads to a convergent algorithm.

Next we carefully set the parameters acccording to the criteria derived in Theorem 2. The following parameter values were used; $N = 13, \epsilon = 10^{-6}, \lambda = 0.01$. Since the column constraint is always satisfied, we made C positive definite in the linear subspace of the column constraint with the smallest eigenvalue in the subspace $\lambda = 0.01$. Since the column subspace corresponds to the eigenvalues of G alone, we shifted the spectrum of G such that its smallest eigenvalue in the column subspace is 0.1. We shifted the spectrum of g such that its smallest eigenvalue (unrestricted to any subspace) is 0.1. Since the eigenvalues of C are the product of the eigenvalues of G and g, we achieve our lower bound of $\lambda = 0.01$. (The restriction to a linear subspace does not affect the above.)

With N, ϵ and λ set, we may calculate the lower bound on δ.

$$\delta >= 2\sqrt{\frac{\epsilon\left[\sum_j \max_{a,b,c,i}\left(C_{ai;cj}-C_{bi;cj}\right)+\max_{a,b,i}\left(A_{ai}-A_{bi}\right)+\frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}\right]}{\lambda N}}$$

Since $C_{ai;bj} = G_{ab}g_{ij}$,

$$\sum_j \max_{a,b,c,i}\left(C_{ai;cj}-C_{bi;cj}\right)=\sum_j \max\left[\max_i g_{ij}\max_c\left(\max_a G_{ac}-\min_a G_{ac}\right),\min_i g_{ij}\min_c\left(\min_a G_{ac}-\max_a G_{ac}\right)\right]$$

The lower bound on δ is calculated at each temperature and used as a convergence criterion; $\sqrt{\frac{\sum_{ai}\Delta M_{ai}^2}{N^2}} \leq \delta$. At each temperature, the softassign QAP algorithm is executed until $\sqrt{\frac{\sum_{ai}\Delta M_{ai}^2}{N^2}}$ falls below the lower bound on δ. In all experiments, we used the following linear temperature schedule with $\beta_0 = \beta_r = 0.01$. The overall convergence criterion was $1-\frac{\sum_{ai}M_{ai}^2}{N} \leq 0.1$ and row dominance. At each temperature, we checked to see if $1-\frac{\sum_{ai}M_{ai}^2}{N}$ became less than 0.1. We also checked to see if we obtained a permutation matrix upon executing a winner-take-all on the rows of M. This is called row dominance (Kosowsky and Yuille, 1994). With the parameters set in this manner, $\delta \approx 0.025$. While δ remains a function of the temperature, it does not significantly change over the entire range of temperatures for the particular set of chosen parameters. The energy difference shown in Figure 2 is always greater than zero.

Next, we break the conditions imposed by Theorem 2. The parameter ϵ is changed from its earlier value of 10^{-6} to 0.01. At the same time λ is kept fixed but the convergence criterion parameter δ is dropped to $\delta = 0.001$. Using Theorem 2 to recalculate δ would approximately result in $\delta = 2.5$ which is unacceptable as a convergence threshold for $\sqrt{\frac{\sum_{ai}\Delta M_{ai}^2}{N^2}}$. We executed the softassign QAP algorithm with the above parameters and with all other parameters (like the annealing schedule) kept exactly the same as in the previous experiment.

During the evolution of the dynamical system, we monitored the energy difference derived in (28) corresponding to Theorem 2. Since the second term in (28) is always non-negative, we did not include it in the energy difference computation. The energy difference is a straightforward combination of a quadratic term and a Lagrange parameter energy term. The monitored energy difference corresponding to Theorem 2 is

$$\Delta E = \frac{1}{2}\sum_{aibj} C_{ai;bj}\Delta M_{ai}\Delta M_{bj} + \sum_a \mu_a \sum_i \Delta M_{ai}$$

where we have used the fact that $\sum_a \mu_a^{(\cdot)} = 0$. The energy difference ΔE is plotted in Figure 3. ΔE fluctuates around zero due to the comparatively larger value of ϵ. Finally in Figure 4, we show the relative contributions of the quadratic and Lagrange parameter terms.

5 Discussion

The existence of a discrete time Lyapunov function for the softassign quadratic assignment algorithm is of fundamental importance. Since the existence of the Lyapunov function does not depend on period two limit cycles, it should be possible to show under mild assumptions regarding the number of fixed points (Koiran, 1994) that the softassign QAP algorithm converges to a fixed point. This will apply equally to both the cases of exact and approximate doubly stochastic constraint satisfaction for the exponential neuronal activation function. Also, the extension of constraint satisfaction to the case of outliers in graph matching (Gold and Rangarajan, 1996; Rangarajan et al., 1997b) and the multiple membership constraint in graph partitioning (Peterson and Soderberg, 1989) should present no problems for the construction of a Lyapunov function; the case of exact constraint satisfaction is trivial and only minor modifications are needed to derive the bound on the Lagrange parameter for the case of approximate constraint satisfaction. The results derived for the general quadratic assignment problem can be specialized to the individual cases of TSP, subgraph isomorphism, graph matching and graph partitioning. An initial effort along these lines, specific to exact constraint satisfaction appears in (Rangarajan et al., 1997b).

A Bounds on the Lagrange parameter vector μ

We begin by rewriting the update equation for M [from (7)]:

$$M_{ai}^{(n+1)} = \exp\left[\beta\left(B_{ai}^{(n)} - \mu_a - \nu_i\right) - 1\right] \tag{31}$$

When Sinkhorn balancing approximately converges, we assume that the column constraint $\sum_a M_{ai}^{(n+1)} = 1$ is exactly satisfied and the row constraint is approximately satisfied: $\forall a,\ |\sum_i M_{ai}^{(n+1)} - 1| < \epsilon$. Since the column constraint is exactly satisfied, we may eliminate ν from (31).

$$\begin{aligned}
\sum_a M_{ai}^{(n+1)} = 1 \quad &\Rightarrow \quad \sum_a \exp\left[\beta\left(B_{ai}^{(n+1)} - \mu_a - \nu_i\right) - 1\right] = 1 \\
&\Rightarrow \quad \exp(\beta\nu_i) = \sum_a \exp\left[\beta\left(B_{ai}^{(n+1)} - \mu_a\right) - 1\right] \\
&\Rightarrow \quad M_{ai}^{(n+1)} = \frac{\exp\left[\beta\left(B_{ai}^{(n+1)} - \mu_a\right)\right]}{\sum_a \exp\left[\beta\left(B_{ai}^{(n+1)} - \mu_a\right)\right]}.
\end{aligned} \tag{32}$$

This is identical to the familiar *softmax* nonlinearity (Bridle, 1990) with the understanding that μ has to be set such that the row constraint is approximately satisfied. Before proceeding with the derivation of the bound on μ, note that (32) is invariant to global shifts of μ: the transformation $\mu_a \to \mu_a + \alpha,\ \forall a$ leaves (32) unchanged. Consequently, without loss of generality, we can assume that $\sum_a \mu_a = 0$. Now,

$$\begin{aligned}
M_{ai}^{(n+1)} &= \frac{\exp\left[\beta\left(B_{ai}^{(n+1)} - \mu_a\right)\right]}{\sum_b \exp\left[\beta\left(B_{bi}^{(n+1)} - \mu_b\right)\right]} \\
&= \frac{1}{1 + \sum_{b\neq a} \exp\left[\beta\left(\left\{B_{bi}^{(n+1)} - B_{ai}^{(n+1)}\right\} + \{\mu_a - \mu_b\}\right)\right]} \\
&\leq \frac{1}{1 + \min_i \sum_{b\neq a} \exp\left[\beta\left(\left\{B_{bi}^{(n+1)} - B_{ai}^{(n+1)}\right\} + \{\mu_a - \mu_b\}\right)\right]}
\end{aligned} \tag{33}$$

Since approximate convergence of the row constraint implies that $1-\epsilon \leq \sum_i M_{ai}^{(n+1)}$, we may write

$$\begin{aligned} 1-\epsilon &\leq \sum_i \frac{1}{1+\sum_{b\neq a}\exp\left[\beta\left(\left\{B_{bi}^{(n+1)}-B_{ai}^{(n+1)}\right\}+\{\mu_a-\mu_b\}\right)\right]} \\ &\leq \frac{N}{1+\min_i\sum_{b\neq a}\exp\left[\beta\left(\left\{B_{bi}^{(n+1)}-B_{ai}^{(n+1)}\right\}+\{\mu_a-\mu_b\}\right)\right]}. \end{aligned} \tag{34}$$

This can be rearranged to give

$$\min_i \sum_{b\neq a}\exp\left[\beta\left(\left\{B_{bi}^{(n+1)}-B_{ai}^{(n+1)}\right\}+\{\mu_a-\mu_b\}\right)\right] \leq \frac{N-1+\epsilon}{1-\epsilon} \tag{35}$$

The above inequality remains true for each term in the summation (on the left). Hence,

$$\forall a,b,\ \min_i\left(B_{bi}^{(n+1)}-B_{ai}^{(n+1)}\right)+(\mu_a-\mu_b) \leq \frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}$$

from which we get

$$\forall a,b,\ \mu_a-\mu_b \leq \max_i\left(B_{ai}^{(n+1)}-B_{bi}^{(n+1)}\right)+\frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}.$$

Since $\sum_a \mu_a = 0$ can be assumed without loss of generality, we may write

$$\forall a,\ |\mu_a| \leq \max_{b,i}\left(B_{ai}^{(n+1)}-B_{bi}^{(n+1)}\right)+\frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon} \tag{36}$$

A bound on μmax, the maximum value of μ follows from (36):

$$\mu\text{max} \leq \max_{a,b,i}\left(B_{ai}^{(n+1)}-B_{bi}^{(n+1)}\right)+\frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon} \tag{37}$$

This result was first reported in (Yuille and Kosowsky, 1991). For the sake of completion, we have rederived the above bound on the Lagrange parameter vector μ.

Thus far, we have sought bounds on μ with respect to the *support matrix* B. However, B in QAP is not a pre-specified constant. Instead, B depends on the current estimate of M:

$$B_{ai}^{(n+1)} = \sum_{bj} C_{ai;bj}M_{bj}^{(n)} - A_{ai} \tag{38}$$

Substituting (38) in (37), we get

$$\mu\text{max} \leq \max_{a,b,i}\left(\sum_{cj}C_{ai;cj}M_{cj}^{(n)}-\sum_{cj}C_{bi;cj}M_{cj}^{(n)}-A_{ai}+A_{bi}\right)+\frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}$$

Since $\max$ and $\sum$ commute when the entries of $M^{(n)}$ are non-negative,

$$\mu\text{max} \leq \sum_{cj}\max_{a,b,i}\left(C_{ai;cj}-C_{bi;cj}\right)M_{cj}^{(n)}+\max_{a,b,i}\left(A_{ai}-A_{bi}\right)+\frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}.$$

Define $D_{cj} \stackrel{\text{def}}{=} \max_{a,b,i}\left(C_{ai;cj}-C_{bi;cj}\right)$ and $\delta \stackrel{\text{def}}{=} \max_{a,b,i}(A_{ai}-A_{bi})$. Now,

$$\mu\text{max} \leq \sum_{ai} D_{ai}M_{ai}^{(n)}+\delta+\frac{1}{\beta}\log\frac{N-1+\epsilon}{1-\epsilon}. \tag{39}$$

The dependence of μmax on the time step n in (39) is obviously unsatisfactory. From the constraint $\sum_a M_{ai}^{(n)} = 1$, we get

$$\sum_{ai} D_{ai} M_{ai}^{(n)} \leq \sum_i \max_a D_{ai} \tag{40}$$

Using (40), we get

$$\begin{aligned} \mu\text{max} &\leq \sum_i \max_a D_{ai} + \delta + \frac{1}{\beta} \log \frac{N-1+\epsilon}{1-\epsilon} \\ &\leq \sum_j \max_{a,b,c,i} (C_{ai;cj} - C_{bi;cj}) + \max_{a,b,i} (A_{ai} - A_{bi}) + \frac{1}{\beta} \log \frac{N-1+\epsilon}{1-\epsilon}. \end{aligned} \tag{41}$$

References

Blum, E. K. and Wang, X. (1992). Stability of fixed points and periodic orbits and bifurcations in analog neural networks. *Neural Networks*, 5:577–587.

Bridle, J. S. (1990). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2*, pages 211–217, San Mateo, CA. Morgan Kaufmann.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. Ser. B*, 39:1–38.

Fogelman-Soulie, F., Mejia, C., Goles, E., and Martinez, S. (1989). Energy functions in neural networks with continuous local functions. *Complex Systems*, 3:269–293.

Gee, A. H., Aiyer, S., and Prager, R. W. (1993). An analytical framework for optimizing neural networks. *Neural Networks*, 6:79–97.

Gee, A. H. and Prager, R. W. (1994). Polyhedral combinatorics and neural networks. *Neural Computation*, 6(1):161–180.

Gold, S. and Rangarajan, A. (1995). Softmax to softassign: Neural network algorithms for combinatorial optimization. *Journal of Artificial Neural Networks*, 2(4):381–399.

Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(4):377–388.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81:3088–3092.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214.

Koiran, P. (1994). Dynamics of discrete time, continuous state Hopfield networks. *Neural Computation*, 6(3):459–468.

Kosowsky, J. J. and Yuille, A. L. (1994). The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks*, 7(3):477–490.

Luenberger, D. (1984). *Linear and Nonlinear Programming*. Addison–Wesley, Reading, MA.

Marcus, C. M. and Westervelt, R. M. (1989). Dynamics of iterated-map neural networks. *Physical Review A*, 40:501–504.

Mjolsness, E. and Garrett, C. (1990). Algebraic transformations of objective functions. *Neural Networks*, 3:651–669.

Peterson, C. and Soderberg, B. (1989). A new method for mapping optimization problems onto neural networks. *Intl. Journal of Neural Systems*, 1(1):3–22.

Rangarajan, A., Chui, H., Mjolsness, E., Pappu, S., Davachi, L., Goldman-Rakic, P., and Duncan, J. (1997a). A robust point matching algorithm for autoradiograph alignment. *Medical Image Analysis*. (in press).

Rangarajan, A., Gold, S., and Mjolsness, E. (1996). A novel optimizing network architecture with applications. *Neural Computation*, 8(5):1041–1060.

Rangarajan, A., Yuille, A. L., Gold, S., and Mjolsness, E. (1997b). A convergence proof for the softassign quadratic assignment algorithm. In *Advances in Neural Information Processing Systems (NIPS) 9*, pages 620–626. MIT Press.

Simic, P. D. (1990). Statistical mechanics as the underlying theory of 'elastic' and 'neural' optimisations. *Network*, 1:89–103.

Sinkhorn, R. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879.

Urahama, K. (1996). Mathematical Programming Formulations for Neural Combinatorial Optimization Algorithms. *Journal of Artificial Neural Networks*.

Van den Bout, D. E. and Miller III, T. K. (1990). Graph partitioning using annealed networks. *IEEE Trans. Neural Networks*, 1(2):192–203.

von der Malsburg, C. (1990). Network self-organization. In Zornetzer, S. F., Davis, J. L., and Lau, C., editors, *An Introduction to Neural and Electronic Networks*, pages 421–432. Academic Press, San Diego, CA.

Wang, X., Jagota, A., Botelho, F., and Garzon, M. (1996). Absence of cycles in symmetric neural networks. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems (NIPS) 8*, pages 372–378. MIT Press.

Waugh, F. R. and Westervelt, R. M. (1993). Analog neural networks with local competition. I. Dynamics and stability. *Physical Review E*, 47(6):4524–4536.

Wolfe, W. J., Parry, M. H., and MacMillan, J. M. (1994). Hopfield-style neural networks and the TSP. In *IEEE Intl. Conf. on Neural Networks*, volume 7, pages 4577–4582. IEEE Press.

Yuille, A. L. and Kosowsky, J. J. (1991). The invisible hand algorithm: Time convergence and temperature tracking. Technical Report 91–10, Harvard Robotics Laboratory.

Yuille, A. L. and Kosowsky, J. J. (1994). Statistical physics algorithms that converge. *Neural Computation*, 6(3):341–356.

List of Figures

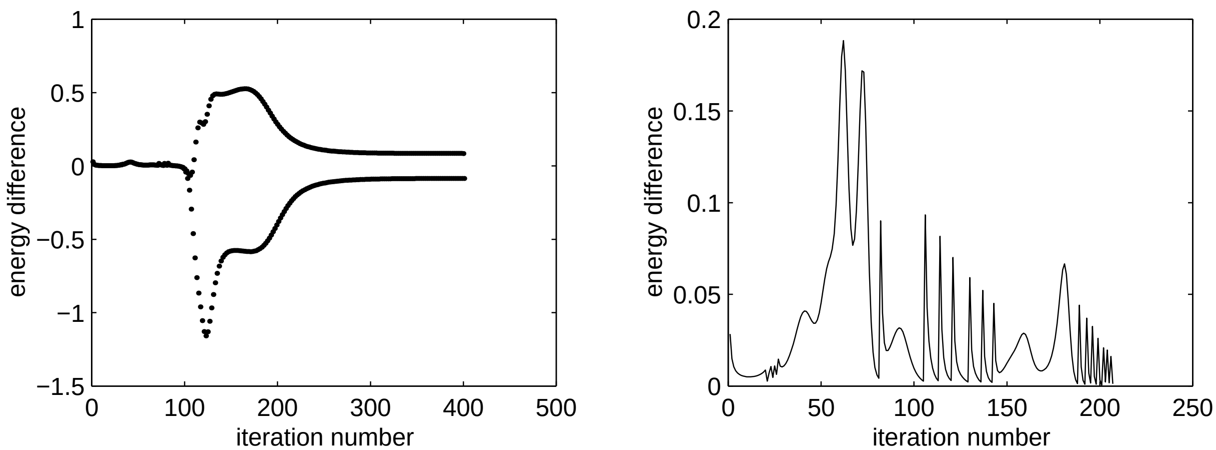

Figure 1: **Energy difference plot.** Left: The change in energy is not always positive when C is not positive definite. Right: The change in energy is always positive when C is positive definite. The energy difference (on the left) implies that the energy function sometimes increases whereas the positive energy difference (on the right) implies that the energy function never increases.

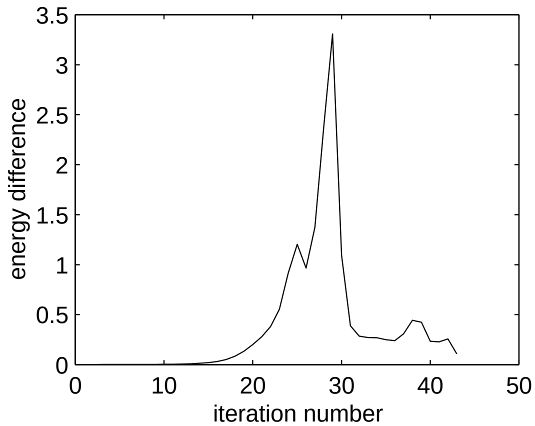

Figure 2: **Energy difference plot:** The change in energy is always positive when the conditions established by Theorem 2 are imposed.

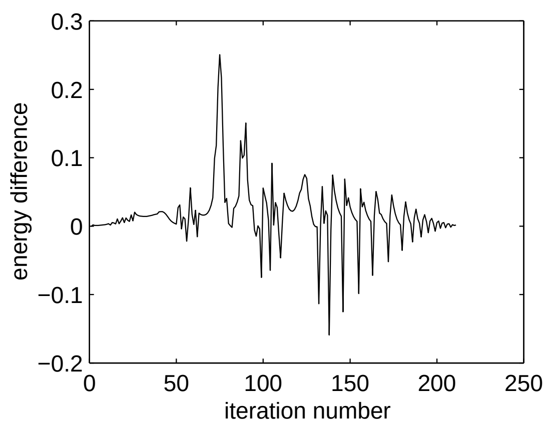

Figure 3: **Energy difference plot:** ΔE. $\epsilon = 0.01,\ \delta = 0.001$. The energy difference plot clearly shows that carefully setting ϵ and δ is needed to ensure convergence.

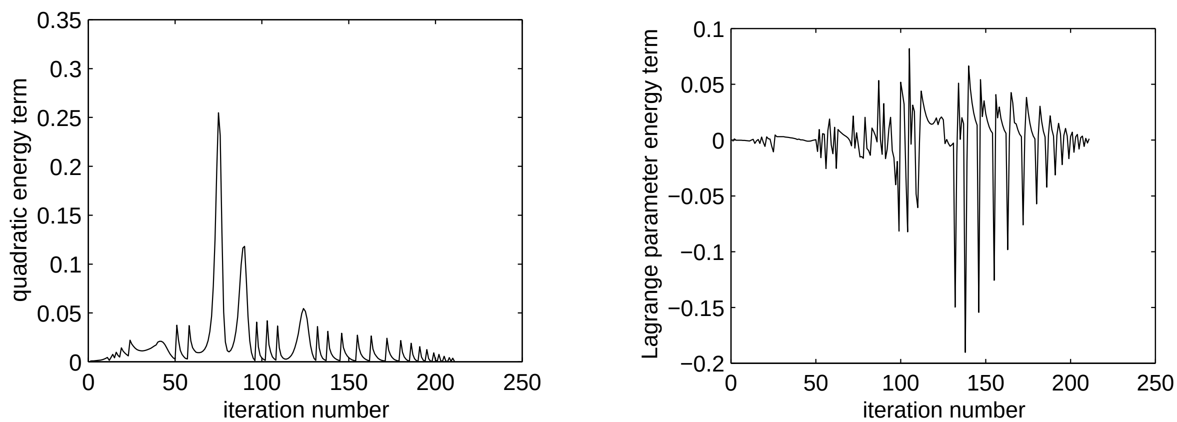

Figure 4: **Energy difference plot:** Left: Quadratic term and Right: Lagrange parameter energy difference term $\sum_a \mu_a \sum_i \Delta M_{ai}$ in ΔE.