Self Annealing: Unifying deterministic annealing and relaxation labeling

Anand Rangarajan

Department of Diagnostic Radiology, Yale University, New Haven, CT, USA

Abstract. Deterministic annealing and relaxation labeling algorithms for classification and matching are presented and discussed. A new approach —self annealing—is introduced to bring deterministic annealing and relaxation labeling into accord. Self annealing results in an emergent linear schedule for winner-take-all and assignment problems. Also, the relaxation labeling algorithm can be seen as an approximation to the self annealing algorithm for matching and labeling problems.

1 Introduction

Labeling and matching problems abound in computer vision and pattern recognition (CVPR). It is not an exaggeration to state that some form or the other of the basic problems of template matching or data clustering has remained central to the CVPR and neural networks communities for about three decades. Due to the somewhat disparate natures of these communities, different frameworks for formulating and solving these two problems have emerged and it is not immediately obvious how to go about reconciling some of the differences between these frameworks so that they can benefit from each other.

In this paper, we pick two such frameworks, deterministic annealing [18, 24, 8, 19] and relaxation labeling [21] which arose mainly in the neural networks and pattern recognition communities respectively. Deterministic annealing has its origins in statistical physics and more recently in Hopfield networks [10]. It has been applied with varying degrees of success to a variety of image matching and labeling problems. In the field of neural networks, deterministic annealing developed from its somewhat crude origins in the Hopfield-Tank networks [10] to include fairly sophisticated treatment of constraint satisfaction and mean-field dynamics by drawing from statistical physics. Recently, for both matching and classification problems, a fairly coherent framework and suite of algorithms has emerged. These algorithms range from using the softmax or softassign for constraint satisfaction and discrete-time dynamics that mimic the Expectation–Maximization (EM) algorithm. The term relaxation labeling originally referred to a heuristic technique developed in [21] in the mid 70's. Relaxation labeling specified a discrete-time update rule by which class labels (typically in image segmentation problems) were refined while taking relationships in the pixel and label array into account. As interest in the technique grew, many bifurcations and off shoots of the basic idea developed, spanning the spectrum from *ad hoc* fixes to principled modifications and justifications [5, 11, 9, 17, 16, 3] based on probability, optimization and dynamical systems theories. Relaxation labeling in its basic form is a discrete-time update equation that is suitably (and fairly obviously) modified depending on the problem of interest—image matching, segmentation, or classification. Deviations from the basic form of relaxation labeling replaced the discrete-time update rule by gradient descent and projected gradient descent [11, 5] on the objective functions. Much of this development prefigured the evolution of optimizing neural networks; from the original Hopfield–Tank dynamics via the softmax dynamics [18, 7] to projected gradient descent [6] or softassign dynamics for the quadratic assignment problem [19, 8].

Here, we return to the heuristic origins of relaxation labeling since ironically, it is in the original discrete-time RL dynamical system that we find the closest parallel to recent deterministic annealing algorithms (which have a completely different line of development from energy functions via mean field theory to algorithms). A new approach—self annealing (SA)—is presented which promises to unify relaxation labeling (RL) and deterministic annealing (DA).

2 Deterministic Annealing

Deterministic annealing arose as a computational shortcut to simulated annealing. Closely related to *mean field* theory, the method consists of minimizing the *free energy* at each temperature setting. The free energy is separately constructed for each problem. The temperature is reduced according to a pre-specified annealing schedule. DA has been applied to a variety of combinatorial optimization problems—winner-take-all (WTA), linear assignment (AP), quadratic assignment (QAP) including the traveling salesman problem, graph matching and graph partitioning, quadratic winner-take-all (QWTA) problems including pairwise clustering, line process models in visual reconstruction etc. with varying degrees of success.

In this paper, we focus on the relationship between DA and RL with emphasis on matching and labeling problems. The archetypal problem at the heart of labeling problems is the winner-take-all and similarly for matching problems, it is linear assignment that is central. Consequently, our development dwells considerably on these two problems.

2.1 The winner take all

The WTA problem is stated as follows: Given a set T_i, $i \in \{1, \ldots, N\}$, find $i^* = \arg\max_i(T_i,\ i \in \{1, \ldots, N\})$ or in other words, find the index of the maximum number. Using N binary variables s_i, $i \in \{1, \ldots, N\}$, the problem is restated as:

$$\max_s \sum_i T_i s_i$$
$$\text{s. to } \sum_i s_i = 1, \text{ and } s_i \in \{0,1\},\ \forall\, i\ . \tag{1}$$

The DA free energy is written as follows:

$$F_{\text{wta}}(v) = -\sum_i T_i v_i + \lambda(\sum_i v_i - 1) + \frac{1}{\beta}\sum_i v_i \log v_i \ . \tag{2}$$

In (2), v is a new set of *analog* mean field variables summing to one. The transition from binary variables s to analog variables v is deliberately highlighted here. Also, β is the *inverse temperature* to be varied according to an annealing schedule. λ is a Lagrange parameter satisfying the WTA constraint. The $x \log x$ form of the barrier function keeps the v variables positive and is also referred to as an *entropy* term.

We now proceed to solve for the v variables and the Lagrange parameter λ. We get (after eliminating λ)

$$v_i^{(\beta)} = \frac{\exp(\beta T_i)}{\sum_j \exp(\beta T_j)}, \ \forall i \ \in \{1, \ . \ . \ . \ , N\} \ . \tag{3}$$

This is referred to as the *softmax* nonlinearity [1]. DA WTA uses the nonlinearity within an annealing schedule. (Here, we gloss over the technical issue of propagating the solution at a given temperature $v^{(\beta_n)}$ to be the initial condition at the next temperature β_{n+1}.) When there are no ties, this algorithm finds the single winner for any reasonable annealing schedule—quenching at high β being one example of an "unreasonable" schedule.

2.2 The linear assignment problem

The AP is written as follows: Given a matrix of numbers $A_{ai}, \ a, i \in \{1, \ . \ . \ . \ , N\}$, find the *permutation* that maximizes the assignment. Using N^2 binary variables $s_{ai}, \ a, i \in \{1, \ . \ . \ . \ , N\}$, the problem is restated as:

$$\max_s \sum_{ai} A_{ai} s_{ai}$$
$$\text{s. to } \sum_i s_{ai} = 1, \sum_a s_{ai} = 1, \text{ and } s_{ai} \in \{0, 1\}, \ \forall \ a, i \ . \tag{4}$$

The DA AP free energy is written as follows:

$$F_{\text{ap}}(v) = -\sum_{ai} A_{ai} v_{ai} + \sum_a \mu_a(\sum_i v_{ai} - 1) + \sum_i \nu_i(\sum_a v_{ai} - 1) + \frac{1}{\beta}\sum_{ai} v_{ai} \log v_{ai} \ . \tag{5}$$

In (5), v is a doubly stochastic mean field matrix with rows and columns summing to one. (μ, ν) are Lagrange parameters satisfying the row and column WTA constraints. As in the WTA case, the $x \log x$ form of the barrier function keeps the v variables positive.

We now proceed to solve for the v variables and the Lagrange parameters (μ, ν) [15, 24]. We get

$$v_{ai}^{(\beta)} = \exp(\beta A_{ai} - \beta[\mu_a + \nu_i]) \ \forall a, i \ \in \{1, \ . \ . \ . \ , N\} \ . \tag{6}$$

AP is distinguished from the WTA by requiring the satisfaction of two-way WTA constraints as opposed to one. Consequently, the Lagrange parameters cannot be solved for in closed form. Rather than solving for the Lagrange parameters using steepest ascent, an iterated row and column normalization method is used to obtain a doubly stochastic matrix at each temperature [15, 19]. Sinkhorn's theorem [22] guarantees the convergence of this method. (This method can be independently derived as coordinate ascent w.r.t. the Lagrange parameters.) With Sinkhorn's method in place, the overall dynamics at each temperature is referred to as the *softassign* [19]. DA uses the softassign within an annealing schedule. (Here, we gloss over the technical issue of propagating the solution at a given temperature $v^{(\beta_n)}$ to be the initial condition at the next temperature β_{n+1}.) When there are no ties, this algorithm finds the optimal permutation for any reasonable annealing schedule.

2.3 Related problems

Having specified the two archetypal problems, the WTA and AP, we turn to other optimization problems which frequently arise in computer vision, pattern recognition and neural networks.

2.4 Clustering and Labeling

Clustering is a very old problem in pattern recognition [4, 12]. In its simplest form, the problem is to separate a set of N vectors in dimension d into K categories. The precise statement of the problem depends on whether central or pairwise clustering is the goal. In central clustering, prototypes are required, in pairwise clustering, a distance measure between any two patterns is needed [2]. Closely related to pairwise clustering is the labeling problem where a set of compatibility coefficients are given and we are asked to assign one unique label to each pattern vector. In both cases, we can write down the following general energy function:

$$\max_s \frac{1}{2} \sum_{aibj} C_{ai;bj} s_{ai} s_{aj}$$

$$\text{s. to} \sum_a s_{ai} = 1, \text{ and } s_{ai} \in \{0,1\}, \; \forall\, a, i \; . \tag{7}$$

(This energy function is a simplification of the pairwise clustering objective function used in [2], but it serves our purpose here.) If the set of compatibility coefficients C is positive definite in the subspace of the one-way WTA constraint, the local minima are WTAs with binary entries. We call this the quadratic WTA (QWTA) problem, emphasizing the quadratic objective with a one-way WTA constraint.

For the first time, we have gone beyond objective functions that are linear in the binary variables s to objective functions quadratic in s. This transition is very important and entirely orthogonal to the earlier transition from the WTA constraint to the permutation constraint. Quadratic objectives with binary variables obeying simplex like constraints are usually much more difficult to minimize than their linear objective counterparts. The DA QWTA free energy is written as follows

$$F_{\text{qwta}}(v) = -\frac{1}{2}\sum_{aibj} C_{ai;bj} v_{ai} v_{bj} + \sum_i \lambda_i (\sum_a v_{ai} - 1) + \frac{1}{\beta}\sum_{ai} v_{ai} \log v_{ai}. \quad (8)$$

Notwithstanding the increased difficulty of this problem, a DA algorithm which is fairly adept at avoiding poor local minima is:

$$q_{ai} \stackrel{\text{def}}{=} \sum_{bj} C_{ai;bj} v_{bj}, \quad (9)$$

$$v_{ai}^{(\beta)} = \frac{\exp(\beta q_{ai})}{\sum_b \exp(\beta q_{bi})} \ . \quad (10)$$

The intermediate q variables have an increased significance in our later discussion on RL. The algorithm consists of iterating the above equations at each temperature. It has been shown to converge to a fixed point provided C is positive definite in the subspace of the WTA constraint [23]. Central and pairwise clustering energy functions have been used in image classification and segmentation or labeling problems in general.

2.5 Matching

Template matching is also one of the oldest problems in vision and pattern recognition. Consequently, the subfield of image matching has become increasingly variegated over the years. In our discussion, we restrict ourselves to feature matching. Akin to labeling or clustering, there are two different styles of matching depending on whether a *spatial mapping* exists between the features in one image and the other. When a spatial mapping exists (or is explicitly modeled), it acts as a strong constraint on the matching. The situation when no spatial mapping is known between the features is similar to the pairwise clustering case. Here, a distance measure between pairs of features in the model and pairs of features in the image is assumed. This results in the QAP objective function—for more details see [8]:

$$\max \frac{1}{2}\sum_{aibj} C_{aibj} s_{ai} s_{bj}$$
$$\text{s. to } \sum_i s_{ai} = 1, \sum_a s_{ai} = 1, \text{ and } s_{ai} \in \{0,1\}, \ \forall\, a, i \quad (11)$$

If the quadratic benefit matrix C is positive definite in the subspace spanned by the row and column constraints, the minima are permutation matrices. This result was shown in [24]. Once again, a DA free energy and algorithm can be written down after spotting the basic form (linear or quadratic objective, one-way or two-way constraint): The DA QAP free energy is written as follows:

$$F_{\mathrm{qap}}(v) = -\frac{1}{2}\sum_{aibj} C_{ai;bj} v_{ai} v_{bj} + \sum_a \mu_a \left(\sum_i v_{ai} - 1\right) + \sum_i \nu_i \left(\sum_a v_{ai} - 1\right) + \frac{1}{\beta}\sum_{ai} v_{ai}\log v_{ai} \quad (12)$$

And the DA QAP algorithm is

$$q_{ai} \stackrel{\text{def}}{=} \sum_{bj} C_{ai;bj} v_{bj}, \quad (13)$$

$$v_{ai}^{(\beta)} = \exp(\beta q_{ai} - \beta[\mu_a + \nu_i]) \ . \quad (14)$$

The two Lagrange parameters μ and ν are specified by Sinkhorn's theorem and the softassign. These two equations (one for the q and one for the v) are iterated until convergence at each temperature. The softassign QAP algorithm is guaranteed to converge to a local minimum provided the Sinkhorn procedure always returns a doubly stochastic matrix [20].

We have written down DA algorithms for two problems (QWTA and QAP) while drawing on the basic forms given by the WTA and the AP. The common features in the two DA algorithms and their differences (one-way versus two-way constraints) [13] have been highlighted as well. We now turn to relaxation labeling.

3 Relaxation Labeling

Relaxation labeling as the name suggests began as a method for solving labeling problems [21]. While the framework has been extended to many applications [17, 3] the basic feature of the framework remains: Start with a set of nodes i (in feature or image space) and a set of labels λ. Derive a set of compatibility coefficients (as in Section 2.4) r for each problem of interest and then apply the basic recipe of RL for updating the node-label (i to λ) assignments:

$$q_i(\lambda) = \sum_{j\mu} r_{ij}(\lambda,\mu) p_j(\mu), \quad (15)$$

$$p_i^{(n+1)}(\lambda) = \frac{p_i^{(n)}(\lambda)(1+\alpha q_i^{(n)}(\lambda))}{\sum_\mu p_i^{(n)}(\mu)(1+\alpha q_i^{(n)}(\mu))} \ . \quad (16)$$

Here the p's are the node-label (i to λ) labeling probabilities, the q are intermediate variables similar to the q's defined earlier in DA. α is a parameter greater than zero used to make the numerator positive (and keep the probabilities positive.) We have deliberately written the RL update equation in a quasi-canonical form while suggesting (at this point) similarities most notably to the pairwise clustering update equation. To make the semantic connection to DA more obvious, we now switch to the old usage of the v variables rather than the p's in RL.

$$q_{ia}^{(n)} = \sum_{jb} C_{ai;bj} v_{bj}, \tag{17}$$

$$v_{ia}^{(n+1)} = \frac{v_{ia}^{(n)}(1+\alpha q_{ia}^{(n)})}{\sum_b v_{ib}^{(n)}(1+\alpha q_{ib}^{(n)})} \ . \tag{18}$$

As in the QAP and QWTA DA algorithms, a Lyapunov function exists [16] for RL.

We can now proceed in the reverse order from the previous section on DA. Having written down the basic recipe for RL, specialize to WTA, AP, QWTA and QAP. While the contraction to WTA and QWTA may be obvious, the case of AP and QAP are not so clear. The reason: two-way constraints in AP are not handled by RL. We have to invoke something analogous to the Sinkhorn procedure. Also, there is no clear analog to the iterative algorithms obtained at each temperature setting. Instead the label probabilities directly depend on their previous state which is never encountered in DA. How do we reconcile this situation so that we can clearly state just where these two algorithms are in accord? The introduction of self annealing promises to answer some of these questions and we now turn to its development.

4 Self annealing

Self annealing has one goal, namely, the elimination of a temperature schedule. As a by-product we show that the resulting algorithm bears a close similarity to both DA and RL. The SA update equation for any of the (matching or labeling) problems we have discussed so far is derived [14] by minimizing

$$F(v,\sigma) = E(v) + \frac{1}{\alpha} d(v,\sigma) \tag{19}$$

where $d(v,\sigma)$ is a distance measure between v and an "old" value σ. (The explanation of the "old" value will follow shortly.) When F is minimized w.r.t v, both terms in (19) come into play. Indeed, the distance measure $d(v,\sigma)$ serves as an "inertia" term with the degree of fidelity between v and σ determined by the parameter α. For example, when $d(v,\sigma)$ is $\frac{1}{2}\|v-\sigma\|^2$, the update equation obtained after taking derivatives w.r.t. v and σ and setting the results to zero is

$$\begin{aligned} \sigma_i &= v_i^{(n)} \\ v_i^{(n+1)} &= \sigma_i - \alpha \left. \frac{\partial E(v)}{\partial v_i} \right|_{v=v^{(n+1)}} \ . \end{aligned} \tag{20}$$

This update equation reduces to "vanilla" gradient descent provided we approximate $\left.\frac{\partial E(v)}{\partial v_i}\right|_{v=v^{(n+1)}}$ by $\left.\frac{\partial E(v)}{\partial v_i}\right|_{v=v^{(n)}}$. α becomes a step-size parameter. However, the distance measure is not restricted to just quadratic error measures. Especially, when positivity of the v variables is desired, a Kullback-Leibler (KL) distance measure can be used for $d(v, \sigma)$. In [14], the authors derive many linear on-line prediction algorithms using the KL divergence. Here, we apply the same approach to the QWTA and QAP.

Examine the following QAP objective function using the KL divergence as the distance measure:

$$F_{\text{saqap}}(v, \sigma, \mu, \nu, \alpha) = -\frac{1}{2} \sum_{aibj} C_{ai;bj} v_{ai} v_{bj} + \frac{1}{\alpha} \sum_{ai} \left(v_{ai} \log \frac{v_{ai}}{\sigma_{ai}} - \sigma_{ai} + v_{ai} \right) + \sum_a \mu_a \left(\sum_i v_{ai} - 1\right) + \sum_i \nu_i \left(\sum_a v_{ai} - 1\right) . \tag{21}$$

We have used the generalized KL divergence $d(x, y) = \sum_i (x_i \log \frac{x_i}{y_i} - x_i + y_i)$ which is guaranteed to be greater than or equal to zero without requiring the usual constraints $\sum_i x_i = \sum_i y_i = 1$. This energy function looks very similar to the earlier DA energy function (12) for QAP. However, it has no temperature parameter. The parameter α is fixed and positive. Instead of the entropy barrier function, this energy function has a new KL measure between v and a new variable σ. Without trying to explain the SA algorithm in its most complex form (QAP), we specialize immediately to the WTA.

$$F_{\text{sawta}}(v, \sigma, \lambda, \alpha) = -\sum_i T_i v_i + \lambda\left(\sum_i v_i - 1\right) + \frac{1}{\alpha} \sum_i \left(v_i \log \frac{v_i}{\sigma_i} - \sigma_i + v_i \right) . \tag{22}$$

Equation (22) can be alternately minimized w.r.t. v and σ (using a closed form solution for the Lagrange parameter λ) resulting in

$$v_i^{(n+1)} = \frac{v_i^{(n)} \exp(\alpha T_i)}{\sum_j v_j^{(n)} \exp(\alpha T_j)},\ v_i^{(0)} > 0,\ \forall i,\ i \in \{1, \ldots, N\} . \tag{23}$$

The new variable σ is identified with $v_i^{(n)}$ in (23). When an alternating minimization (between v and σ) is prescribed for F_{sawta}, the update equation (23) results. Initial conditions are an important factor. A reasonable choice is $v_i^{(0)} = 1/N$, $\sigma_i^{(0)} = v_i^{(0)}$, $\forall i,\ i \in \{1, \ldots, N\}$ but other positive, initial conditions may work as well. To summarize, in the WTA, the new variable σ is identified with the "past" value of v. We have not yet shown any relationship to DA or RL.

Moving to the QAP, the main update equation used by the algorithm is

$$q_{ai} \stackrel{\text{def}}{=} \sum_{bj} C_{ai;bj} v_{bj}^{(n)}, \tag{24}$$

$$v_{ai}^{(n+1)} = \sigma_{ai} \exp(\alpha q_{ai} - \alpha[\mu_a + \nu_i]) . \tag{25}$$

Convergence of the SA QAP algorithm to a local minimum can be easily shown when we assume that the Sinkhorn procedure always returns a doubly stochastic matrix. Our treatment follows [20]. A discrete-time Lyapunov function for the SA QAP algorithm is (21). (The Lagrange parameter terms can be eliminated since we are restricting v to be doubly stochastic.) The change in energy is written as

$$F_{\text{saqap}}(v^{(n)},\sigma) - F_{\text{saqap}}(v^{(n+1)},\sigma) \stackrel{\text{def}}{=} \Delta F_{\text{SAQAP}} =$$
$$= -\frac{1}{2}\sum_{aibj} C_{ai;bj} v_{ai}^{(n)} v_{bj}^{(n)} + \frac{1}{\alpha}\sum_{ai} v_{ai}^{(n)} \log \frac{v_{ai}^{(n)}}{\sigma_{ai}}$$
$$+\frac{1}{2}\sum_{aibj} C_{ai;bj} v_{ai}^{(n+1)} v_{bj}^{(n+1)} - \frac{1}{\alpha}\sum_{ai} v_{ai}^{(n+1)} \log \frac{v_{ai}^{(n+1)}}{\sigma_{ai}} \quad . \tag{26}$$

The Lyapunov energy difference has been simplified using the relation $\sum_{ai} v_{ai} = N$. Using the update equation for SA in (25), the energy difference is rewritten as

$$\Delta F_{\text{saqap}} = \frac{1}{2}\sum_{aibj} C_{ai;bj} \Delta v_{ai} \Delta v_{bj} + \sum_{ai} v_{ai}^{(n)} \log \frac{v_{ai}^{(n)}}{v_{ai}^{(n+1)}} \geq 0 \tag{27}$$

where $\Delta v_{ai} \stackrel{\text{def}}{=} v_{ai}^{(n+1)} - v_{ai}^{(n)}$. The first term in (27) is non-negative due to the positive definiteness of C in the subspace spanned by the row and column constraints. The second term is non-negative by virtue of being a KL distance measure. We have shown the convergence to a fixed point of the SA QAP algorithm.

We now write down the QAP SA algorithm:

Self annealing QAP
Initialize v_{ai} to $\frac{1}{N}$, σ_{ai} to v_{ai}
Begin A: Do A until row dominance and $(1 - p_{\text{norm}}) < p_{\text{thr}}$.
  Begin B: Do B until $e_{\text{diff}} < e_{\text{thr}}$.
  $q_{ai} \leftarrow \sum_{bj} C_{ai;bj} v_{bj}$
  $v_{ai} \leftarrow \sigma_{ai} \exp(\alpha q_{ai})$
    Begin C: Do C until $s_{\text{norm}} < s_{\text{thr}}$.
    Update v_{ai} by normalizing the rows:
    $v_{ai} \leftarrow \frac{v_{ai}}{\sum_i v_{ai}}$
    Update v_{ai} by normalizing the columns:
    $v_{ai} \leftarrow \frac{v_{ai}}{\sum_a v_{ai}}$
    End C
  End B
$\sigma_{ai} \leftarrow v_{ai}$
End A

The various parameters are defined as: $p_{\text{norm}} \stackrel{\text{def}}{=} \frac{\sum_{ai} v_{ai}^2}{N}$, $e_{\text{diff}} \stackrel{\text{def}}{=} \Delta F_{\text{saqap}}$, and $s_{\text{norm}} \stackrel{\text{def}}{=} \sqrt{\frac{\sum_a (\sum_i v_{ai} - 1)^2}{N}}$. p_{thr}, e_{thr}, and s_{thr} are the permutation, energy difference and Sinkhorn convergence thresholds respectively. Row dominance implies that thresholding v returns a permutation matrix [15]. This is the full blown SA QAP algorithm with Sinkhorn's method and the softassign used for the constraints but more importantly a built in delay between the "old" value of v namely σ and the current value of v.

5 Self annealing and deterministic annealing

SA and DA are closely related. To see this, we return to our favorite example—the WTA. The SA and DA WTAs are now brought into accord: Assume uniform rather than random initial conditions for SA. $v_i^{(0)} = 1/N,\ \forall i,\ i \in \{1, \ldots, N\}$. With uniform initial conditions, it is trivial to solve for $v_i^{(n)}$:

$$v_i^{(n)} = \frac{\exp(n\alpha T_i)}{\sum_j \exp(n\alpha T_j)},\ \forall i,\ i \in \{1, \ldots, N\}\ . \tag{28}$$

The correspondence between SA and DA is clearly established by setting $\beta_n = n\alpha,\ n = 1, 2, \ldots$ We have shown that the SA WTA corresponds to a particular *linear* schedule for the DA WTA.

Since the case of AP is more involved than WTA, we present anecdotal experimental evidence that SA and DA are closely related. In Figure 1, we have shown the evolution of the permutation norm and the AP free energies. A linear schedule with $\beta = n\alpha$ was used. The correspondence between DA and SA is nearly exact for the permutation norm despite the fact that the free energies evolve in a different manner. The correspondence is exact only when we match the linear schedule DA parameter α to the SA parameter α. It is important that SA and DA be in lockstep, otherwise we cannot make the claim that SA corresponds to DA with an emergent linear schedule.

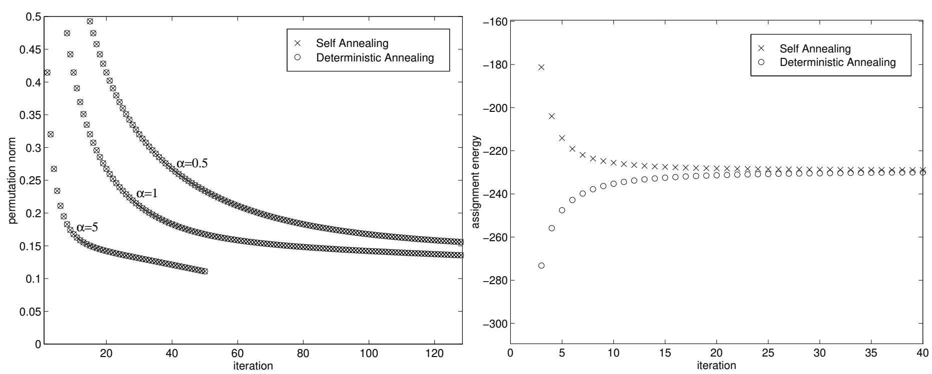

Fig. 1. Left: 100 node AP with three different schedules. The agreement between SA and DA is obvious. Right: The evolution of the SA and DA AP free energies for one schedule.

The SA and DA QAP objective functions are also quite general. The QAP or QWTA benefit matrix $C_{ai;bj}$ is preset based on the chosen problem—weighted, graph matching, or pairwise clustering. Note the basic similarity between the SA and DA QAP algorithms. In SA, a separation between past (σ) and present (v) replaces relaxation at a fixed temperature. Moreover, in the WTA and AP, SA results in an emergent linear schedule. A similar argument can be made for QAP as well but requires experimental validation due to the presence of bifurcations. We return to this topic in Section 7.

6 Self annealing and relaxation labeling

Rather than present the RL update equation in its "canonical" labeling problem form, we once again return to the WTA problem where the similarities between SA and RL are fairly obvious. The RL WTA update equation is

$$v_i^{(n+1)} = \frac{v_i^{(n)}(1+\alpha T_i)}{\sum_j v_j^{(n)}(1+\alpha T_j)},\ v_i^{(0)} > 0,\ \forall i,\ i \in \{1,\ \ldots, N\}\ . \tag{29}$$

Equations (23) and (29) are very similar. The main difference is the $1 + \alpha T_j$ factor in RL instead of the $\exp(\alpha T_j)$ factor in SA Expanding $\exp(\alpha T_j)$ using the Taylor-MacLaurin series gives

$$f(\alpha) = \exp(\alpha T_j) = 1 + \alpha T_j + R_2(\alpha) \tag{30}$$

where

$$R_2(\alpha) \leq \frac{\exp(\alpha |T_j|)\alpha^2 T_j^2}{2}\ . \tag{31}$$

If the remainder $R_2(\alpha)$ is small, the RL WTA closely approximates SA WTA. This will be true for small values of α. Increased divergence between RL and SA can be expected as α is increased—faster the rate of the *linear* schedule, faster the divergence. If $|T_j| > \frac{1}{\alpha}$, the non-negativity constraint is violated leading to breakdown of the RL algorithm.

Comparison at the WTA level is not the end of the story. RL in its heyday was applied to image matching, registration, segmentation and classification problems. Similar to the QAP formulation, the benefit matrix C called the compatibility coefficients in the RL literature was introduced and preset depending on the chosen problem. Because of the bias towards labeling problems, the all important distinction between matching and labeling was blurred. In model matching problems (arising in object recognition and image registration), a two way constraint is required. Setting up one-to-one correspondence between features on the model and features in the image requires such a two-way assignment constraint. On the other hand, only a one way constraint is needed in segmentation, classification, clustering and coloring problems since i) the label and the data fields occupy different spaces and ii) many data features share membership under the same label. (Despite sharing the multiple membership feature of these labeling problems, graph partitioning has a two-way constraint because of the requirement that all multiple memberships be equal in number—an arbitrary requirement from the standpoint of labeling problems arising in pattern recognition.)

Due to the bias towards labeling, RL almost never tried to enforce two-way constraints either using something like the Sinkhorn procedure in discrete-time algorithms or using projected gradient descent in continuous time algorithms. This is an important difference between SA and DA on one hand and RL on the other.

Another important difference is the separation of past and present. Due to the close ties of both SA and DA to simulated annealing, the importance of relaxation at a fixed temperature is fairly obvious. Otherwise, a very slow annealing schedule has to be prescribed to avoid poor local minima. Due to the entirely heuristic origin of RL and due to the lack of an analog of a temperature parameter, the importance of relaxation at fixed temperature was not recognized. Examining the SA and RL QAP algorithms, it is clear that RL roughly corresponds to one iteration at each temperature. This issue is orthogonal to constraint satisfaction. Even if Sinkhorn's procedure is implemented in RL—and all that is needed is non-negativity of each entry of the matrix $1 + \alpha Q_{ai}$—the separation of past (σ) and present (v) is still one iteration. Put succinctly, step B in SA is allowed only one iteration.

A remaining difference is the positivity constraint, We have already discussed the relationship between the exponential in SA and the $(1 + \alpha T_i)$ RL term in the WTA context. There is no need to repeat the analysis for QAP—note that positivity is guaranteed by the exponential whereas it must be checked in RL.

In summary, there are three principal differences between SA and RL: (i) The positivity constraint is strictly enforced by the exponential in SA and loosely enforced in RL, (ii) the use of the softassign rather than the softmax in matching problems has no parallel in RL and finally (iii) the discrete-time SA QAP update equation introduces an all important delay between past and present (roughly corresponding to multiple iterations at each temperature) whereas RL having no such delay forces one iteration per temperature with consequent loss of accuracy (as demonstrated in the next section).

7 Results

We conducted several hundreds of experiments comparing the performance of DA, RL, and SA discrete-time algorithms. The chosen problems were QAP and QWTA.

In QAP, we randomly generated benefit matrices C (of size $N \times N \times N \times N$) that are positive definite in the subspace spanned by the row and column constraints. The procedure is as follows: Define a matrix $r \stackrel{\text{def}}{=} I_N - e_N e_N^T / N$ where e_N is the vector of all ones. Generate a matrix R by taking the Kronecker product of r with itself ($R \stackrel{\text{def}}{=} r \otimes r$). Rewrite $\hat{C}$ as a two-dimensional $N^2 \times N^2$ matrix $\hat{c}$. Project $\hat{c}$ into the subspace of the row and column constraints by forming the matrix $R\hat{c}R$. Determine the smallest eigenvalue $\lambda_{\min}(R\hat{c}R)$. Then the matrix $c \stackrel{\text{def}}{=} \hat{c} - \lambda_{\min}(R\hat{c}R)I_{N^2} + \epsilon I_{N^2}$ (where ϵ is a small, positive quantity) is positive definite in the subspace spanned by the row and column constraints.

Four algorithms were executed on the QAP. Other than the three algorithms mentioned previously, we added a new algorithm called exponentiated relaxation (ER). ER is closely related to SA. The only difference is that the inner B loop in SA is performed just once ($I_B = 1$). ER is also closely related to RL. The main difference is that the positivity constraint is enforced via the exponential. Since the QAP has both row and column constraints, the Sinkhorn procedure is used in ER just as in SA. However, RL enforces just one set of constraints. To avoid this asymmetry in algorithms, we replaced the normalization procedure in RL by the Sinkhorn procedure, thereby avoiding unfair comparisons. As long as the positivity constraint is met in RL, we are guaranteed to obtain doubly stochastic matrices. There is overall no proof of convergence, however, for this "souped up" version of RL.

The common set of parameters shared by the four algorithms were kept exactly the same: $N = 25$, $\epsilon = 0.001$, Sinkhorn norm threshold $s_{\text{thr}} = 0.0001$, energy difference threshold $e_{\text{thr}} = 0.001$, permutation norm threshold $p_{\text{thr}} = 0.001$, and initial condition $v^{(0)} = e_N e_N^T / N$. The stopping criterion chosen was $p_{\text{thr}} = 0.001$ and row dominance [15]. In this way, we ensured that all four algorithms returned permutation matrices. A linear schedule $\beta = n\alpha$ was used in DA. The parameter α was varied logarithmically from $\log(\alpha) = -2$ to $\log(\alpha) = 1$ in steps of 0.1. 100 experiments were run for each of the four algorithms. The common benefit matrix $\hat{c}$ shared by the four algorithms was generated using independent, Gaussian random numbers. $\hat{c}$ was then made symmetric by forming $\frac{\hat{c}+\hat{c}^T}{2}$. The results are shown in Figure 2(a).

The most interesting feature emerging from the experiments is that there is an intermediate range of α in which self annealing performs at its best. (The negative of the QAP minimum energy is plotted on the ordinate.) Contrast this with ER and RL which do not share this feature. We conjecture that this is due to the "one iteration per temperature" policy of both these algorithms. RL could not be executed once the positivity constraint was violated but ER had no such problems. Also, notice that the performances of both SA and DA are nearly identical after $\alpha = 0.2$. The emergent linear schedule in SA derived analytically for the WTA and demonstrated in AP seems to be valid only after a certain value of α in both QAP and QWTA.

Figure 2(b) shows the results of QWTA. The behavior is very similar to the QAP. In QWTA the benefit matrices were projected onto the subspace of only one of the constraints (row or column). In other respects, the experiments were carried out in exactly the same manner as QAP. Since there is only one set of constraints, the canonical version of RL [21] was used. Note that the negative of the minimum energy is consistently higher in QWTA than QAP; this is due to the absence of the second set of constraints.

Next we studied the behavior of self annealing with changes in problem size. In Figure 3(a), the problem size is varied from $N = 2$ to $N = 25$ in steps of one. We normalized the QAP minimum energy at $\log(\alpha) = -2$ for all values of N.

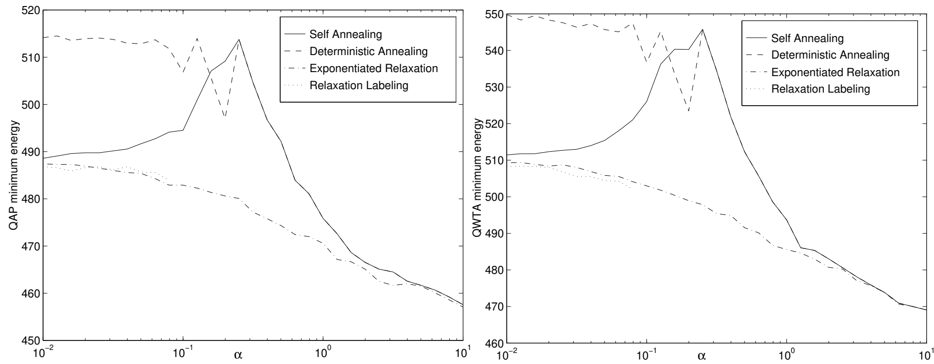

Fig. 2. Median of 100 experiments at each value of α. Left: (a) QAP. Right (b) QWTA. The negative of the QAP and QWTA minimum energies is plotted on the ordinate.

Not only is the overall pattern of behavior more or less the same, in addition there is an impressive invariance to the choice of the broad range of α. This evidence is very anecdotal however.

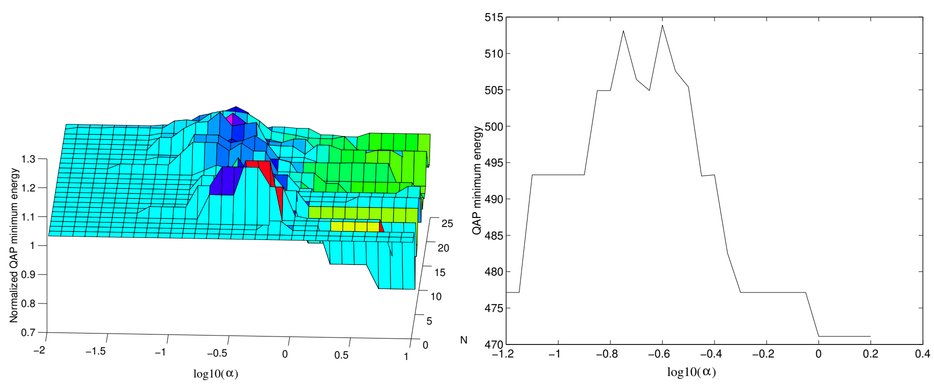

Fig. 3. Self annealing: Left: (a) Normalized negative QAP minimum energy plot for problem size N varying from 2 to 25 in steps of one. The performance is somewhat invariant to the broad range of α. Right. (b) Negative QAP minimum energy plot in a more finely sampled range of α.

Finally, we present some evidence to show that there is a qualitative change in the behavior of the self annealing algorithm roughly around $\alpha = 0.15$. The energy plot in Figure 3(b), the contour and "waterfall" plots in Figure 4 indicate the presence of different regimes in SA. The change in the permutation norm with iteration and α is a good qualitative indicator of this change in regime. Our results are very preliminary and anecdotal here. We do not as yet have any understanding of this qualitative change in behavior of SA with change in α.

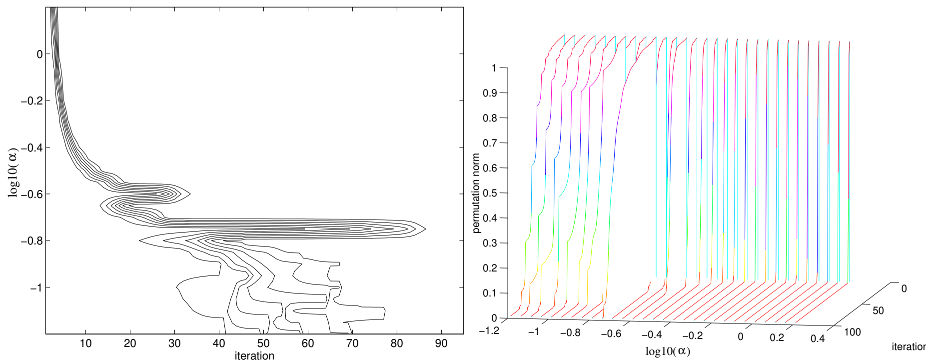

Fig. 4. Self Annealing: Left: A contour plot of the permutation norm versus α and the number of iterations. Right: A "waterfall" plot of the permutation norm versus α and the number of iterations. Both plots illustrate the abrupt change in behavior around $\alpha = 0.1$.

8 Conclusions

We have demonstrated that self annealing has the potential to reconcile relaxation labeling with deterministic annealing when applied to matching and labeling problems. While the relaxation labeling dynamical system has a Lyapunov energy function [16], we have shown that there exists a class of hitherto unsuspected self annealing energy functions that are also closely related to relaxation labeling. Our experiments and analyses suggest that relaxation labeling can be extended in a self annealing direction until the two become almost indistinguishable. The same cannot be said for deterministic annealing since it has more formal origins in mean field theory. Also, it remains to be seen if some of the more recent modifications to relaxation labeling like probabilistic relaxation [3] can be brought under the same rubric as deterministic annealing.

Acknowledgements

We acknowledge Manfred Warmuth for a helpful conversation. We thank Haili Chui, Steven Gold, Eric Mjolsness and Paul Stolorz for stimulating discussions.

References

1. J. S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 211–217, San Mateo, CA, 1990. Morgan Kaufmann.
2. J. Buhmann and T. Hofmann. Central and pairwise data clustering by competitive neural networks. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 104–111. Morgan Kaufmann, San Francisco, CA, 1994.

3. W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(5):749–764, Aug. 1995.
4. R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, NY, 1973.
5. O. Faugeras and M. Berthod. Improving consistency and reducing ambiguity in stochastic labeling: an optimization approach. *IEEE Trans. Patt. Anal. Mach. Intell.*, 3(4):412–424, Jul. 1981.
6. A. H. Gee and R. W. Prager. Polyhedral combinatorics and neural networks. *Neural Computation*, 6(1):161–180, Jan. 1994.
7. D. Geiger and A. L. Yuille. A common framework for image segmentation. *Intl. Journal of Computer Vision*, 6(3):227–243, Aug. 1991.
8. S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.
9. E. R. Hancock and J. Kittler. Discrete relaxation. *Pattern Recognition*, 23(7):711–733, 1990.
10. J. J. Hopfield and D. Tank. 'Neural' computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.
11. R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 5(3):267–287, May 1983.
12. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, NJ, 1988.
13. B. Kamgar-Parsi and B. Kamgar-Parsi. On problem solving with Hopfield networks. *Biological Cybernetics*, 62:415–423, 1990.
14. J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. Technical Report UCSC-CRL-94-16, Univ. Calif. Santa Cruz, June 1994.
15. J. J. Kosowsky and A. L. Yuille. The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks*, 7(3):477–490, 1994.
16. M.Pelillo. On the dynamics of relaxation labeling processes. In *IEEE Intl. Conf. on Neural Networks (ICNN)*, volume 2, pages 606–1294. IEEE Press, 1994.
17. M. Pelillo. Learning compatibility coefficients for relaxation labeling processes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 16(9):933–945, Sept. 1994.
18. C. Peterson and B. Söderberg. A new method for mapping optimization problems onto neural networks. *Intl. Journal of Neural Systems*, 1(1):3–22, 1989.
19. A. Rangarajan, S. Gold, and E. Mjolsness. A novel optimizing network architecture with applications. *Neural Computation*, 8(5):1041–1060, 1996.
20. A. Rangarajan, A. L. Yuille, S. Gold, and E. Mjolsness. A convergence proof for the softassign quadratic assignment algorithm. In *Advances in Neural Information Processing Systems (NIPS) 9.* MIT Press, 1997. (in press).
21. A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Syst. Man, Cybern.*, 6(6):420–433, Jun. 1976.
22. R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.
23. F. R. Waugh and R. M. Westervelt. Analog neural networks with local competition. I. Dynamics and stability. *Physical Review E*, 47(6):4524–4536, June 1993.
24. A. L. Yuille and J. J. Kosowsky. Statistical physics algorithms that converge. *Neural Computation*, 6(3):341–356, May 1994.