Self annealing and self annihilation: Unifying deterministic annealing and relaxation labeling

Anand Rangarajan
Image Processing and Analysis Group
Department of Diagnostic Radiology
Yale University School of Medicine
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 linear assignment problems. Self annihilation, a generalization of self annealing is capable of performing the useful function of symmetry breaking. The original relaxation labeling algorithm is then shown to arise from an approximation to either the self annealing energy function or the corresponding dynamical system. With this relationship in place, self annihilation can be introduced into the relaxation labeling framework. Experimental results on synthetic matching and labeling problems clearly demonstrate the three-way relationship between deterministic annealing, relaxation labeling and self annealing.

Keywords: Deterministic annealing, relaxation labeling, self annealing, self amplification, self annihilation, softmax, softassign.

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 and data clustering has remained central to the CVPR and neural networks (NN) communities for about three decades [1]. 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 [2] and relaxation labeling [3] 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 [4]. 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 [4] to include fairly sophisticated treatment of constraint satisfaction and energy minimization by drawing on well established principles in statistical physics [5]. Recently, for both matching [6] and classification [7] problems, a fairly coherent framework and set of algorithms have emerged. These algorithms range from using the softmax [8] or softassign [9] for constraint satisfaction and dynamics that are directly derived from or merely mimic the Expectation–Maximization (EM) approach [10].

The term relaxation labeling (RL) originally referred to a heuristic dynamical system developed in [11]. RL specified a discrete time dynamical system in 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, off shoots and generalizations of the basic idea developed; examples are the product combination rule [12], the optimization approach [13], projected gradient descent [3], discrete relaxation [14], and probabilistic relaxation [15]. RL 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. The more principled deviations from the basic form of RL replaced the discrete time update rule by gradient descent and projected gradient descent [3, 13] on energy functions. However, recently it has been shown [16] that the original heuristic RL dynamical system minimizes the labeling energy function. It is now fairly clear that both continuous time projected gradient descent and discrete time RL dynamical systems can be used to minimize the same labeling energy function.

Much of this development prefigured or ran parallel to the evolution of deterministic annealing (DA) dynamical systems with at least one major difference. While the concerns of continuous time versus discrete time dynamics were common to both RL and DA approaches, within the DA approaches a fundamental distinction was drawn between matching and labeling problems [17]. This distinction was almost never emphasized in RL. In labeling problems, a set of labels have to be assigned to a set of nodes with the constraint that a node should be assigned only one label. A variety of problems not necessarily restricted to CVPR require labeling constraints; some examples are central and pairwise clustering [7, 18], consistent labeling [3], and graph coloring. In matching problems on the other hand, a set of model nodes have to be assigned to a set of data nodes with the constraint that each model node should be assigned to one and only one data node and *vice versa*. A variety of problems require matching constraints; some examples are quadratic assignment [2, 19], TSP [20, 9], graph matching [21, 22], graph partitioning (with minor differences) [20, 23] and point matching [24, 25]. The original neural network approaches used a penalty function approach at fixed temperature [4]. With the importance of deterministic annealing and exact constraint satisfaction becoming clear, these approaches quickly gave way to the softmax [26, 20, 23, 27, 28], softassign [29, 9, 22], Lagrangian relaxation [29, 30] and projected gradient descent [31, 32, 33, 34] approaches usually performed within deterministic annealing.

Here, we return to the original relaxation labeling dynamical system since ironically, it is in the RL discrete time dynamical system that we find a closest parallel to recent discrete-time deterministic annealing algorithms. Even after restricting our focus to discrete time dynamical systems, important differences like the manner in which constraint satisfaction is performed, relaxation at a fixed temperature and the nature of the update mechanism remain. A new approach—self annealing—is presented to unify relaxation labeling and deterministic annealing. We show that the self annealing dynamical system which is derived from a corresponding energy function corresponds to deterministic annealing with a linear schedule. Also, the original RL update equation can be derived from the self annealing dynamical system via a Taylor series approximation. This suggests that a close three-way relationship exists between DA, RL and self annealing with self annealing acting as a bridge between DA and RL.

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. Deterministic annealing has been applied to a variety of combinatorial optimization problems—winner-take-all (WTA), linear assignment, quadratic assignment including the traveling salesman problem, graph matching and graph partitioning, clustering (central and pairwise), the Ising model etc.—and to nonlinear optimization problems as well with varied success. In this paper, we focus on the relationship between deterministic annealing and relaxation labeling 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 winner-take-all problem is stated as follows: Given a set of numbers $T_i,\ i \in \{1, \ldots, N\}$, 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:

$$\max_s \sum_i T_i s_i \tag{1}$$

$$\text{s. to } \sum_i s_i = 1, \text{ and } s_i \in \{0, 1\},\ \forall\, i \tag{2}$$

The deterministic annealing free energy is written as follows:

$$F_{\text{wta}}(v) = -\sum_i T_i v_i + \lambda(\sum_i v_i - 1) + \frac{1}{\beta} \sum_i v_i \log v_i. \tag{3}$$

In equation (3), 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)

$$v_i^{(\beta)} = \frac{\exp(\beta T_i)}{\sum_j \exp(\beta T_j)}, \; \forall i, \; i \in \{1, \ldots, N\} \tag{4}$$

This is referred to as the *softmax* nonlinearity [8]. Deterministic annealing WTA uses the nonlinearity within an annealing schedule. (Here, we gloss over the technical issue of propagating the solution at a given temperature v^{β_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 linear assignment problem is written as follows: Given a matrix of numbers $A_{ai}, \; a, i \in \{1, \ldots, N\}$, find the *permutation* that maximizes the assignment. Using N^2 binary variables $s_{ai}, \; a, i \in \{1, \ldots, N\}$, the problem is restated as:

$$\max_s \sum_{ai} A_{ai} s_{ai} \tag{5}$$

$$\text{s. to} \sum_i s_{ai} = 1, \sum_a s_{ai} = 1, \text{ and } s_{ai} \in \{0, 1\}, \; \forall \; a, i \tag{6}$$

The deterministic annealing free energy is written as follows:

$$F_{\text{ap}}(v) = -\sum_{ai} A_{ai} v_{ai} + \sum_a \mu_a (\sum_i v_{ai} - 1) + \sum_i \nu_i (\sum_a v_{ai} - 1) + \frac{1}{\beta} \sum_{ai} v_{ai} \log v_{ai}. \tag{7}$$

In equation (7), 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). [29, 2]. We get

$$v_{ai}^{(\beta)} = \exp(\beta A_{ai} - \beta[\mu_a + \nu_i]) \; \forall a, i, \; a, i \in \{1, \ldots, N\} \tag{8}$$

The assignment problem 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 [29, 9]. Sinkhorn's theorem [35] 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* [9]. Deterministic annealing assignment uses the softassign within an annealing schedule.

(Here, we gloss over the technical issue of propagating the solution at a given temperature v^{β_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 winner-take-all and assignment, we turn to other optimization problems which frequently arise in computer vision, pattern recognition and neural networks.

2.3.1 Clustering and labeling

Clustering is a very old problem in pattern recognition [1, 36]. 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 [37, 18]. 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:

$$\min_s E_{\text{lab}}(s) = -\frac{1}{2}\sum_{aibj} C_{ai;bj} s_{ai} s_{bj} \tag{9}$$

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

(This energy function is a simplification of the pairwise clustering objective function used in [37, 18], 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. Notwithstanding the increased difficulty of this problem, a deterministic annealing algorithm which is fairly adept at avoiding poor local minima is:

$$q_{ai} \overset{\text{def}}{=} -\frac{\partial E_{\text{lab}}(v)}{\partial v_{ai}} = \sum_{bj} C_{ai;bj} v_{bj} \tag{10}$$

$$v_{ai}^{(\beta)} = \frac{\exp(\beta q_{ai})}{\sum_b \exp(\beta q_{bi})} \tag{11}$$

The intermediate q variables have an increased significance in our later discussion on relaxation labeling. The algorithm consists of iterating the above equations at each temperature. Central and pairwise clustering energy functions have been used in image classification and segmentation or labeling problems in general [18].

2.3.2 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 [24]. The situation when no spatial mapping is known between the features is similar to the pairwise clustering case. Instead, a distance measure between pairs of features in the model and pairs of features in the image are assumed. This results in the quadratic assignment objective function—for more details see [22]:

$$\min_s E_{\mathrm{gm}}(s) = -\frac{1}{2}\sum_{aibj} C_{ai;bj}\, s_{ai} s_{bj}$$

$$\text{s. to } \sum_i s_{ai} = 1, \sum_a s_{ai} = 1, \text{ and } s_{ai} \in \{0,1\},\ \forall\, a, i \tag{12}$$

If the quadratic benefit matrix $\{C_{ai;bj}\}$ is positive definite in the subspace spanned by the row and column constraints, the minima are permutation matrices. This result was shown in [2]. Once again, a deterministic annealing free energy and algorithm can be written down after spotting the basic form (linear or quadratic objective, one-way or two-way constraint):

$$q_{ai} \stackrel{\mathrm{def}}{=} -\frac{\partial E_{\mathrm{gm}}(v)}{\partial v_{ai}} = \sum_{bj} C_{ai;bj} v_{bj} \tag{13}$$

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

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 quadratic assignment algorithm is guaranteed to converge to a local minimum provided the Sinkhorn procedure always returns a doubly stochastic matrix [19].

We have written down deterministic annealing algorithms for two problems (QWTA and QAP) while drawing on the basic forms given by the WTA and linear assignment problems. The common features in the two deterministic annealing algorithms and their differences (one-way versus two-way constraints) [17] 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 [11]. While the framework has been extended to many applications [38, 39, 40, 41, 16, 15] 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 2.3.1) r for each problem of interest and then apply the basic recipe of relaxation labeling for updating the node-label (i to λ) assignments:

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

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

Here the p's are the node-label (i to λ) label variables, the q are intermediate variables similar to the q's defined earlier in deterministic annealing. α is a parameter greater than zero used to make the numerator positive (and keep the probabilities positive.) The update equation is typically written in the form of a discrete dynamical system. In particular, note the multiplicative update and the normalization step involved in the transition from step n to step $(n + 1)$. We have deliberately written the relaxation labeling update equation in a quasi-canonical form while suggesting (at this point) similarities most notably to the pairwise clustering discrete time update equation. To make the semantic connection to deterministic annealing more obvious, we now switch to the old usage of the v variables rather than the p's in relaxation labeling.

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

$$v_{ai}^{(n+1)} = \frac{v_{ai}^{(n)}(1 + \alpha q_{ai}^{(n)})}{\sum_b v_{bi}^{(n)}(1 + \alpha q_{bi}^{(n)})} \tag{18}$$

As in the QAP and QWTA deterministic annealing algorithms, a Lyapunov function exists [42, 43] for relaxation labeling.

We can now proceed in the reverse order from the previous section on deterministic annealing. Having written down the basic recipe for relaxation labeling, 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 relaxation labeling. 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 variables directly and multiplicatively depend on their previous state which is never encountered in deterministic annealing. 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 deterministic annealing and relaxation labeling. The self annealing update equation for any of the (matching or labeling) problems we have discussed so far is derived by minimizing [44]

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

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 [44], 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:

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

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 deterministic annealing 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 self annealing algorithm in its most complex form (QAP), we specialize immediately to the WTA.

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

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 $v_i^0 = 1/N + \xi_i$, $\sigma_i^0 = v_i^0$, $\forall i$, $i \in \{1, \ldots, N\}$ but other initial conditions may work as well. A small random factor ξ is included in the initial condition specification. 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 deterministic annealing or relaxation labeling.

We now write down the quadratic assignment self annealing algorithm:

Pseudo-code for self annealing QAP

```
Initialize $v_{ai}$ to $\frac{1}{N} + \xi_{ai}$, $\sigma_{ai}$ to $v_{ai}$

Begin A: Do A until integrality condition is met or number of iterations $> I_A$.

    Begin B: Do B until all $v_{ai}$ converge or number of iterations $> I_B$

    $q_{ai} \leftarrow \sum_{bj} C_{ai;bj} v_{bj}$

    $v_{ai} \leftarrow \sigma_{ai} \exp(\alpha q_{ai})$

        Begin C: Do C until all $v_{ai}$ converge or number of iterations $> I_C$
        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
```

This is the full blown self annealing 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. The main update equation used by the algorithm is

$$\frac{1}{\alpha} \log v_{ai}^{(n+1)} = \sum_{bj} C_{ai;bj} v_{bj}^{(n)} - \mu_a - \nu_i + \frac{1}{\alpha} \log \sigma_{ai} \tag{24}$$

Convergence of the self annealing quadratic assignment 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 [19]. A discrete-time Lyapunov function for the self annealing quadratic assignment 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

$$\Delta F_{\text{saqap}} \overset{\text{def}}{=} F_{\text{saqap}}(v^{(n)}, \sigma) - F_{\text{saqap}}(v^{(n+1)}, \sigma)$$

$$= -\frac{1}{2}\sum_{aibj} C_{ai;bj} v_{ai}^{(n)} v_{bj}^{(n)} + \frac{1}{\alpha}\sum_{ai} v_{ai}^{(n)} \log \frac{v_{ai}^{(n)}}{\sigma_{ai}}$$

$$+\frac{1}{2}\sum_{aibj} C_{ai;bj} v_{ai}^{(n+1)} v_{bj}^{(n+1)} - \frac{1}{\alpha}\sum_{ai} v_{ai}^{(n+1)} \log \frac{v_{ai}^{(n+1)}}{\sigma_{ai}} \quad (25)$$

The Lyapunov energy difference has been simplified using the relations $\sum_{ai} v_{ai} = N$. Using the update equation for self annealing in (24), the energy difference is rewritten as

$$\Delta F_{\text{saqap}} = \frac{1}{2}\sum_{aibj} C_{ai;bj} \Delta v_{ai} \Delta v_{bj} + \frac{1}{\alpha}\sum_{ai} v_{ai}^{(n)} \log \frac{v_{ai}^{(n)}}{v_{ai}^{(n+1)}} \geq 0 \quad (26)$$

where $\Delta v_{ai} \stackrel{\text{def}}{=} v_{ai}^{(n+1)} - v_{ai}^{(n)}$. The first term in (26) is non-negative due to the positive definiteness of $\{C_{ai;bj}\}$ in the subspace spanned by the row and column constraints. The second term is non-negative by virtue of being a Kullback-Leibler distance measure. We have shown the convergence to a fixed point of the self annealing QAP algorithm.

5 Self annealing and deterministic annealing

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

$$v_i^{(n)} = \frac{\exp(n\alpha T_i)}{\sum_j \exp(n\alpha T_j)}, \forall i, i \in \{1, \ldots, N\}. \quad (27)$$

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

Since the case of AP is more involved than WTA, we present anecdotal experimental evidence that self annealing and deterministic annealing are closely related. In Figure 1, we have shown the evolution of the *permutation norm* $(1 - \frac{\sum_{ai} v_{ai}^2}{N})$ and the AP free energies. A linear schedule is used for the inverse temperature β with the initial inverse temperature $\beta_0 = \alpha$ and the linear increment β_r also set to α. 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 self annealing 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.

The self annealing and deterministic annealing QAP objective functions are quite general. The QAP benefit matrix $C_{ai;bj}$ is preset based on the chosen problem—inexact, weighted, graph matching, or pairwise clustering. The deterministic annealing pseudo-code follows (we have already written down the self annealing pseudo-code in the previous section):

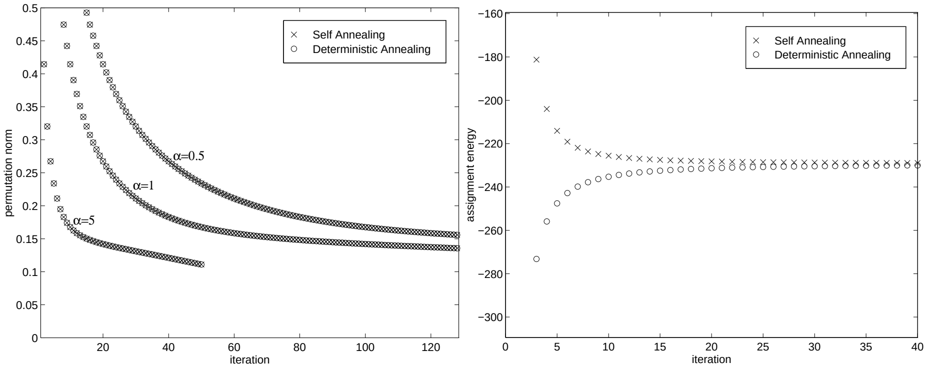

Figure 1: Left: 100 node AP with three different schedules. The agreement between self and deterministic annealing is obvious. Right: The evolution of the self and deterministic annealing AP free energies for one schedule.

Pseudo-code for deterministic annealing QAP

```
Initialize $\beta$ to $\beta_0$, $v_{ai}$ to $\frac{1}{N} + \xi_{ai}$
Begin A: Do A until $\beta \geq \beta_f$
    Begin B: Do B until all $v_{ai}$ converge or number of iterations > $I_B$
    $q_{ai} \leftarrow \sum_{bj} C_{ai;bj} v_{bj}$
    $v_{ai} \leftarrow \exp(\beta q_{ai})$
        Begin C: Do C until all $v_{ai}$ converge or number of iterations > $I_C$
        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
$\beta \leftarrow \beta_r + \beta$
End A
```

Note the basic similarity between the self annealing and deterministic annealing QAP algorithms. In self annealing, a separation between past (σ) and present (v) replaces relaxation at a fixed temperature. Moreover, in the WTA and AP, self annealing 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.

Self Annihilation

Self annealing results in an emergent linear schedule for the WTA and AP. In Section 2 and in the preceding discussion of the relationship between self annealing and deterministic annealing, we glossed over the important issue of *symmetry breaking*.

The problem of resolving ties or symmetries arises in both the WTA and AP and in graph isomorphism (a special case of QAP) [30]. Examine the special case of the WTA objective function (1) with at least two T_i being equal maxima. Neither the DA update equation (4) nor the SA update equation (23) is capable of breaking symmetry. To break symmetry in DA, it is necessary to add a *self amplification* term $-\frac{\gamma}{2}\sum_i v_i^2$ which is functionally equivalent to adding the term $\frac{\gamma}{2}\sum_i v_i(1-v_i)$ (to the WTA) [30]. A similar situation obtains for AP as well. Here, two or more competing permutations may maximize the AP energy and again it is necessary to break symmetry. Otherwise, we obtain a doubly stochastic matrix which is an average over all the equally optimal permutations. A self amplification term of the same form as in the WTA can be added to the energy function in order to break symmetry in DA.

Self annihilation is a different route to symmetry-breaking than self amplification. The basic idea is to make the entropy term in SA become negative, roughly corresponding to a *negative temperature* [34]. We illustrate this idea with the WTA. Examine the SA self annihilation WTA energy function shown below:

$$F_{\text{sannwta}}(v,\sigma) = -\sum_i T_i v_i + \lambda\left(\sum_i v_i - 1\right) + \frac{1}{\alpha}\sum_i\left(v_i \log\frac{v_i}{\sigma_i^\delta} - v_i + \delta\sigma_i\right) \tag{28}$$

In (28), the KL divergence between $v$and the "old" value σ has been modified. Nevertheless, the new WTA objective function can still be minimized w.r.t. v and σ and the earlier interpretation of σ as the "old" value of v still holds. Minimizing (28) by differentiating w.r.t. v and σ and setting the results to zero, we get:

$$\begin{aligned}\frac{\partial F}{\partial v_i} &= 0 \quad \Rightarrow \quad v_i = \frac{\sigma_i^\delta \exp(\alpha T_i)}{\sum_j \sigma_j^\delta \exp(\alpha T_j)} \\ \frac{\partial F}{\partial \sigma_i} &= 0 \quad \Rightarrow \quad \sigma_i = v_i\end{aligned} \tag{29}$$

It is fairly straightforward to show that $\sigma = v$ is a minimum. Substituting the relation $\sigma = v$ in the self annihilation objective function, we get:

$$F_{\text{sannwta}}(v,\sigma(v)) = -\sum_i T_i v_i + \lambda\left(\sum_i v_i - 1\right) + \frac{(1-\delta)}{\alpha}\sum_i\left(v_i\log v_i - v_i\right) \tag{30}$$

The crucial term in the above energy function is the summation over $(1-\delta)v_i\log v_i$. For $\delta \neq 1$, this term is not equal to zero if and only if $v_i \neq 0$ or 1. For $\delta > 1$ this term is strictly greater than zero for $v_i \in (0,1)$. Consequently, in a symmetry breaking situation, the energy can be further reduced by breaking ties while preserving the constraint that $\sum_i v_i = 1$. The update equation after setting

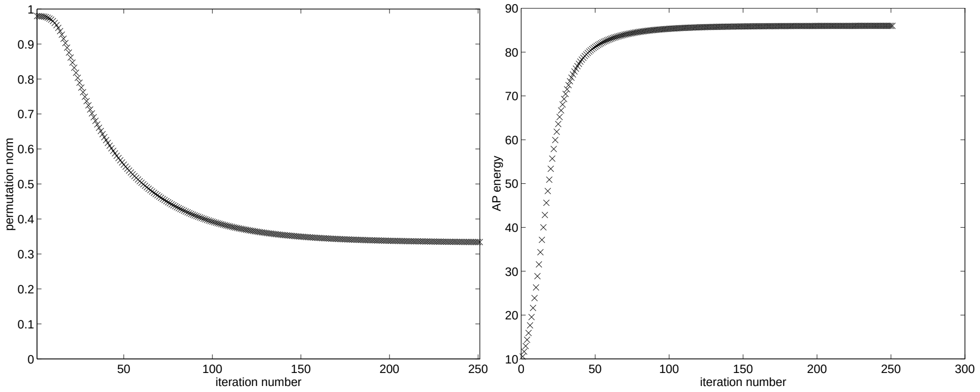

Figure 2: **Self annealing:** 50 node AP with ties. Left: permutation norm. Right: AP energy

$\sigma = v$ is:

$$v_i^{(n+1)} = \frac{\left(v_i^{(n)}\right)^\delta \exp(\alpha T_i)}{\sum_j \left(v_j^{(n)}\right)^\delta \exp(\alpha T_j)},\ v_i^{(0)} > 0,\ \forall i,\ i \in \{1, \ldots, N\}. \tag{31}$$

Once again assuming uniform initial conditions for v, we solve for $v^{(n)}$ to obtain:

$$v_i^{(n)} = \frac{\exp\left[\alpha\left(\frac{\delta^n - 1}{\delta - 1}\right) T_i\right]}{\sum_i \exp\left[\alpha\left(\frac{\delta^n - 1}{\delta - 1}\right) T_i\right]},\ \forall i,\ i \in \{1, \ldots, N\}. \tag{32}$$

The above closed-form solution for v at the nth step in the self annihilation update does not have a limiting form as $n \to \infty$ for $\delta > 1$. For $\delta = 1$, we obtain the emergent linear schedule of the previous section. Examining the self annihilation energy function (30), we may assign the final temperature to be $-\frac{(\delta-1)}{\alpha}$ which is the equivalent negative temperature. The reason we call this process self annihilation is that for any $v_i \in (0,1)$, $v_i^\delta < v_i$ for $\delta > 1$.

We now demonstrate the ability of self annihilation to perform symmetry breaking. In Figure 1, we showed the evolution of the AP self annealing algorithm when there were no ties. The permutation norm $(1 - \frac{\sum_{ai} v_{ai}^2}{N})$ decreases as expected and the AP energy $(\sum_{ai} A_{ai} v_{ai})$ increases to the maximum value. Next, we created a situation where there were multiple maxima and reran the SA algorithm. This result shown in Figure 3 demonstrates the inability of SA to break symmetry. However, when we set $\delta = 1.1$, the algorithm had no difficulty in breaking symmetry (Figure 3).

The tradeoff in using self annihilation is between local minima and speed of convergence to an integer solution. Symmetry breaking can usually be performed in linear problems like WTA and AP by adding some noise to the WTA set T or to the AP benefit matrix A. However, self annihilation is an attractive alternative due to the increased speed with which integer solutions are found.

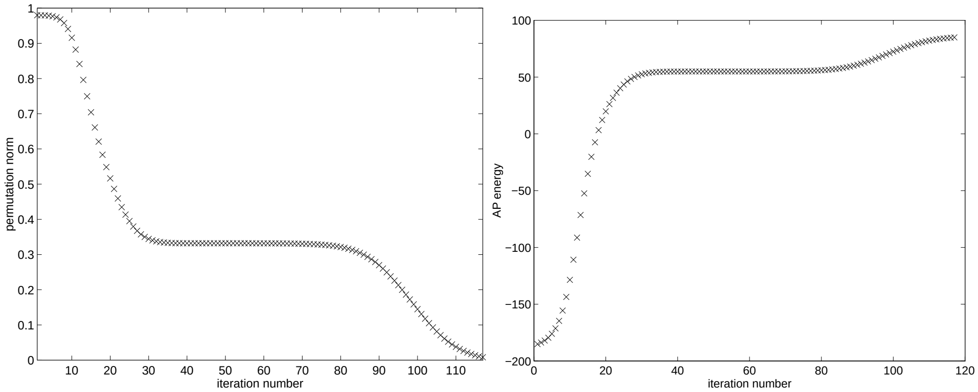

Figure 3: **Self annihilation:** 50 node AP with ties. $\delta = 1.1$. Left: permutation norm. Right: AP energy

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 winner-take-all problem where the similarities between self annealing and RL are fairly obvious. The RL WTA update equation is

$$v_i^{(n+1)} = \frac{v_i^{(n)}(1+\alpha T_i)}{\sum_j v_j^{(n)}(1+\alpha T_j)},\ v_i^{(0)} > 0,\ \forall i,\ i \in \{1,\ \ldots, N\}. \tag{33}$$

Equations (23) and (33) are very similar. The main difference is the $1 + \alpha T_j$ factor in RL instead of the $\exp(\alpha T_j)$ factor in self annealing. Expanding $\exp(\alpha T_j)$ using the Taylor-MacLaurin series gives

$$f(\alpha) = \exp(\alpha T_j) = 1 + \alpha T_j + R_2(\alpha) \tag{34}$$

where

$$R_2(\alpha) \le \frac{\exp(\alpha|T_j|)\alpha^2 T_j^2}{2} \tag{35}$$

If the remainder $R_2(\alpha)$ is small, the RL WTA closely approximates self annealing WTA. This will be true for small values of α. Increased divergence between RL and self annealing 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.

Instead of using a Taylor series expansion at the algorithmic level, we can directly approximate the self annealing energy function. A Taylor series expansion of the KL divergence between the current (v) and previous estimate evaluated at $v = \sigma$ yields:

$$\sum_i \left(v_i \log \frac{v_i}{\sigma_i} - v_i + \sigma_i \right) \approx \sum_i \frac{(v_i - \sigma_i)^2}{2\sigma_i} + \sum_i O\left[(v_i - \sigma_i)^3 \right] \tag{36}$$

This has the form of a χ^2 distance [44]. Expanding the self annealing energy function upto second order (at the current estimate σ), we get:

$$E_{\chi^2}(v, \sigma, \lambda, \alpha) = -\sum_i T_i v_i + \lambda \left(\sum_i v_i - 1 \right) + \frac{1}{\alpha} \sum_i \frac{(v_i - \sigma_i)^2}{2\sigma_i} \tag{37}$$

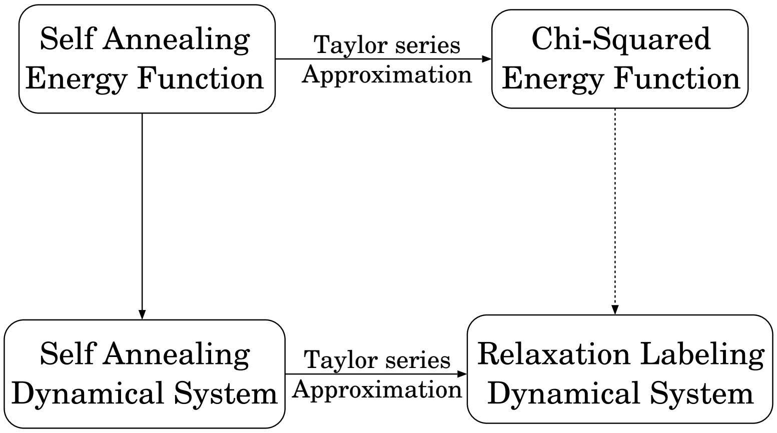
Self Annealing Energy Function
Taylor series Approximation
Chi-Squared Energy Function
Self Annealing Dynamical System
Taylor series Approximation
Relaxation Labeling Dynamical System

Figure 4: From self annealing to relaxation labeling

This new energy function can be minimized w.r.t. v. The fixed points are:

$$\begin{aligned} \frac{\partial E}{\partial v_i} = 0 \quad &\Rightarrow \quad -T_i + \lambda + \frac{v_i - \sigma_i}{\sigma_i} = 0 \\ \frac{\partial E}{\partial \sigma_i} = 0 \quad &\Rightarrow \quad \sigma_i = v_i \end{aligned} \tag{38}$$

which after setting $\sigma = v^{(n)}$ leads to

$$v_i^{(n+1)} = v_i^{(n)} \left[1 + \alpha \left(T_i - \lambda \right) \right] \tag{39}$$

There are many similarities between (39) and (33). Both are multiplicative updating algorithms relying on the derivatives of the energy function. However, the important difference is that the normalization operation in (33) does not correspond to the optimal solution to the Lagrange parameter λ in (39). Solving for λ in (39) by setting $\sum_i v_i = 1$, we get

$$v_i^{(n+1} = v_i^{(n)} \left(1 + \alpha T_i \right) - \alpha \sum_j T_j v_j^{(n)} \tag{40}$$

By introducing the Taylor series approximation at the energy function level and subsequently solving for the update equation, we have obtained a new kind of multiplicative update algorithm, closely related to relaxation labeling. The positivity constraint is not strictly enforced in (40) as in RL and has to be checked at each step. Note that by eschewing the optimal solution to the Lagrange parameter λ in favor of a normalization, we get the RL algorithm for the WTA. The two routes from SA to RL are depicted in Figure 4. A dotted line is used to link the χ-squared energy function to the RL update equation since the normalization used in the latter cannot be derived from the former.

Turning to the problem of symmetry breaking, RL in its basic form is not capable of resolving ties. This is demonstrated in Figure 5 on AP. Just as in SA, self annihilation in RL resolves ties. In Figure 6, the permutation norm $(1 - \frac{\sum_{ai} v_{ai}^2}{N})$ can be reduced to arbitrary small values.

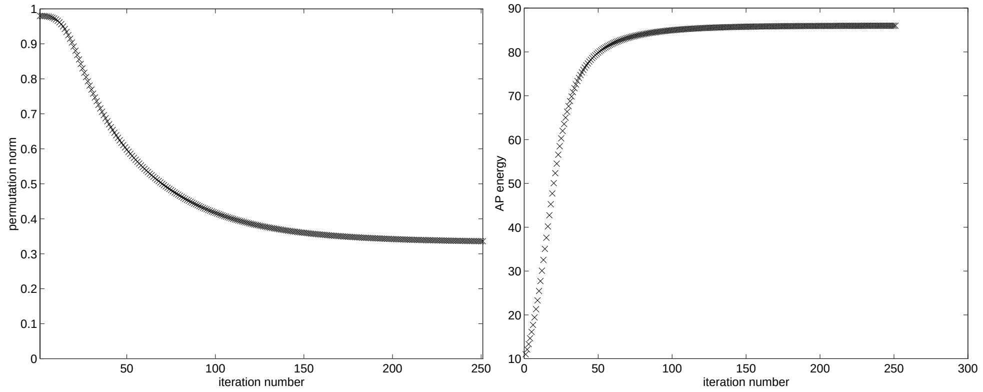

Figure 5: **Relaxation labeling:** 50 node AP with ties. Left: permutation norm. Right: AP energy

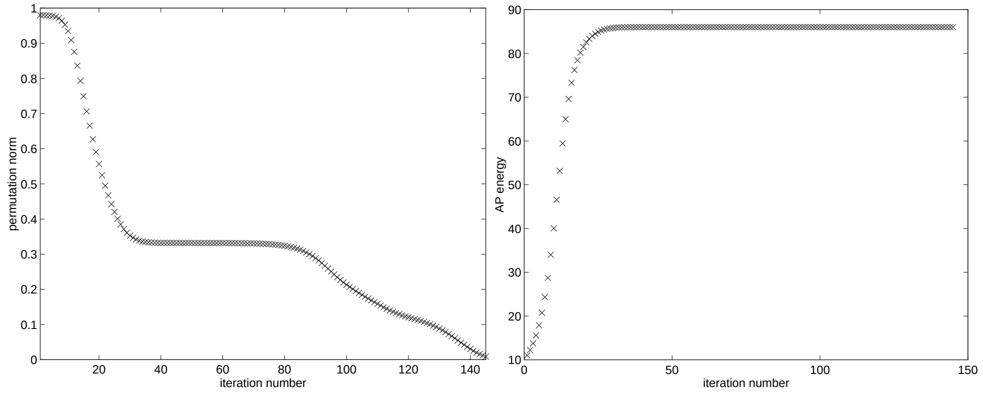

Figure 6: **Relaxation labeling with self annihilation:** 50 node AP with ties. $\delta = 1.1$. Left: permutation norm. Right: AP energy

Comparison at the WTA and AP levels 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 $C_{ai;bj}$ 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 a) the label and the data fields occupy different spaces and b) 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.) Pseudo-code for the QAP RL algorithm is provided below.

Pseudo-code for relaxation labeling QAP

```
Initialize $v_{ai}$ to $\frac{1}{N} + \xi_{ai}$, $\sigma_{ai}$ to $v_{ai}$
Begin A: Do A until integrality condition is met or number of iterations $> I_A$.
    $q_{ai} \leftarrow \sum_{bj} C_{ai;bj} v_{bj}$
    $v_{ai} \leftarrow \sigma_{ai}(1 + \alpha q_{ai})$
    Update $v_{ai}$ by normalizing the columns:
    $v_{ai} \leftarrow \frac{v_{ai}}{\sum_a v_{ai}}$
    $\sigma_{ai} \leftarrow v_{ai}$
End A
```

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 [31, 34]. 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 self and deterministic annealing 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 lack of a temperature parameter in RL, the importance of relaxation at fixed temperature was not recognized. Examining the self annealing 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 is allowed only one iteration.

A remaining difference is the positivity constraint. We have already discussed the relationship between the exponential and the RL term $(1 + \alpha T_i)$ 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 self annealing and RL: (i) The positivity constraint is strictly enforced by the exponential in self annealing 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 self annealing 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.

7 Results

We conducted several hundred experiments comparing the performance of deterministic annealing (DA), relaxation labeling (RL), and self annealing (SA) discrete-time algorithms. The chosen problems were quadratic assignment (QAP) and quadratic winner-take-all (QWTA).

In QAP, we randomly generated benefit matrices $\hat{C}$ (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 $r \overset{\text{def}}{=} I_N - e_N e_N^T / N$ where e_N is the vector of all ones. Generate a matrix R by taking the Kronecker product of r with itself ($R \overset{\text{def}}{=} r \otimes r$). 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 $c \overset{\text{def}}{=} \hat{c} - \lambda_{\min}(R\hat{c}R) I_{N^2} + \epsilon$ (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 $\Delta = 0.0001$, 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

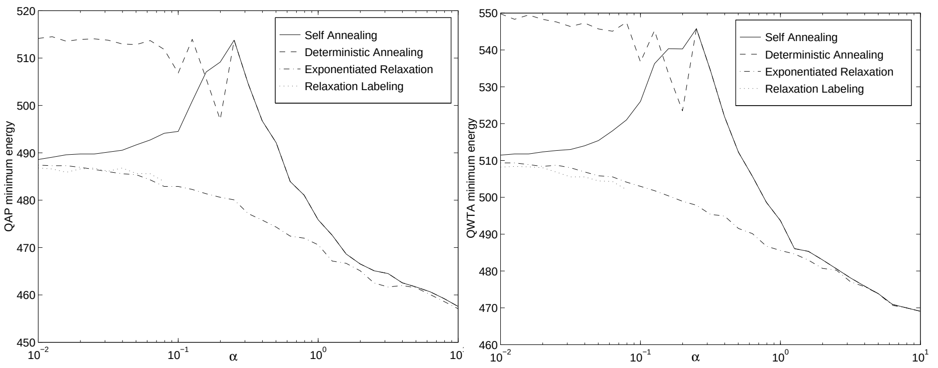

Figure 7: Median of 100 experiments at each value of α. Left: (a) QAP. Right (b) QWTA. The negative of the QAP and QWTA minimum energies is plotted on the ordinate.

criterion chosen was $p_{\mathrm{thr}} = 0.001$ and row dominance [29]. In this way, we ensured that all four algorithms returned permutation matrices. A linear schedule with $\beta_0 = \alpha$ and $\beta_r = \alpha$ 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 7(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 derived analytically for the WTA seems to be valid only after a certain value of α.

Figure 7(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 [11] 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 8(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. 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 also anecdotal.

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 8(b), the contour and "waterfall" plots in Figure 9 indicate the presence of different regimes in SA. The change in

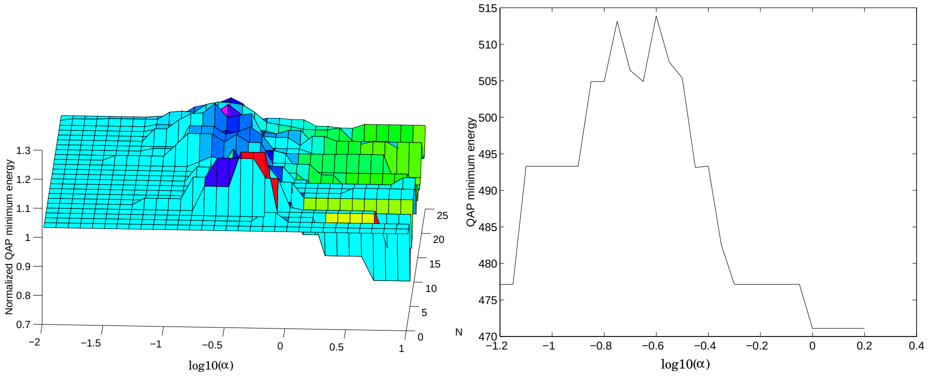

Figure 8: **Self annealing:** Left: (a) Normalized negative QAP minimum energy plot for problem size N varying from 2 to 25 in steps of one. The performance is somewhat invariant to the broad range of α. Right. (b) Negative QAP minimum energy plot in a more finely sampled range of α.

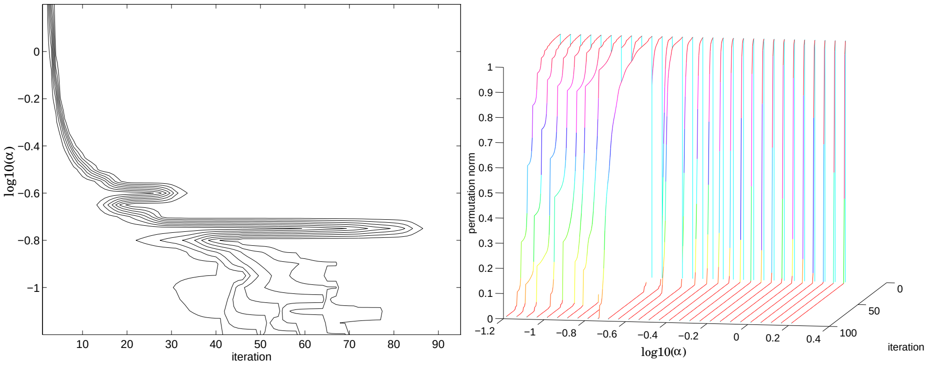

Figure 9: **Self annealing:** Left: A contour plot of the permutation norm versus α. Right: A "waterfall" plot of the permutation norm versus α and the number of iterations. Both plots illustrate the abrupt change in behavior around $\alpha = 0.1$.

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.

8 Discussion

We have for the most part focused on the three way relationships between SA, DA and RL discrete time dynamical systems. One of the reasons for doing so was the ease with which comparison experiments could be conducted. But there is no reason to stop here. Continuous time projected gradient dynamical systems could just as easily have been derived for SA, RL and DA. In fact, continuous time dynamical systems were derived for RL and DA in [3] and in [31, 45] respectively. In a similar vein, SA continuous time projected gradient descent dynamical systems can also be derived. It would be instructive and illuminating to experimentally check the performances of these continuous time counterparts as well as other closely related algorithms such as iterated conditional modes (ICM) [46] and simulated annealing [47, 48] against the performances of the discrete time dynamical systems used in this paper.

9 Conclusion

We have demonstrated that self annealing has the potential to reconcile relaxation labeling and deterministic annealing as applied to matching and labeling problems. Our analysis also suggests that relaxation labeling can itself 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. What this suggests is that there exists a class of hitherto unsuspected self annealing energy functions from which relaxation labeling dynamical systems can be approximately derived. It remains to be seen if some of the other modifications to relaxation labeling like probabilistic relaxation can be related to deterministic annealing.

References

[1] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, NY, 1973.

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

[3] R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 5(3):267–287, May 1983.

[4] J. J. Hopfield and D. Tank. 'Neural' computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[5] G. Parisi. *Statistical Field Theory*. Addison Wesley, Redwood City, CA, 1988.

[6] A. L. Yuille. Generalized deformable models, statistical physics, and matching problems. *Neural Computation*, 2(1):1–24, 1990.

[7] K. Rose, E. Gurewitz, and G. Fox. Constrained clustering as an optimization method. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(8):785–794, 1993.

[8] 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.

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

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

[11] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Syst. Man, Cybern.*, 6(6):420–433, Jun. 1976.

[12] S. Peleg. A new probabilistic relaxation scheme. *IEEE Trans. Patt. Anal. Mach. Intell.*, 2(4):362–369, Jul. 1980.

[13] 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.

[14] E. R. Hancock and J. Kittler. Discrete relaxation. *Pattern Recognition*, 23(7):711–733, 1990.

[15] 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.

[16] M. Pelillo. Learning compatibility coefficients for relaxation labeling processes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 16(9):933–945, Sept. 1994.

[17] B. Kamgar-Parsi and B. Kamgar-Parsi. On problem solving with Hopfield networks. *Biological Cybernetics*, 62:415–423, 1990.

[18] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(1):1–14, Jan. 1997.

[19] 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 9*, pages 620–626. MIT Press, Cambridge, MA, 1997.

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

[21] P. D. Simic. Constrained nets for graph matching and other quadratic assignment problems. *Neural Computation*, 3:268–281, 1991.

[22] 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.

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

[24] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakic, and J. Duncan. A robust point matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 4(1):379–398, 1997.

[25] S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2-D and 3-D point matching: pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.

[26] D. E. Van den Bout and T. K. Miller III. Improving the performance of the Hopfield–Tank neural network through normalization and annealing. *Biological Cybernetics*, 62:129–139, 1989.

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

[28] D. Geiger and A. L. Yuille. A common framework for image segmentation. *Intl. Journal of Computer Vision*, 6(3):227–243, Aug. 1991.

[29] 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.

[30] A. Rangarajan and E. Mjolsness. A Lagrangian relaxation network for graph matching. *IEEE Trans. Neural Networks*, 7(6):1365–1381, 1996.

[31] A. L. Yuille, P. Stolorz, and J. Utans. Statistical physics, mixtures of distributions, and the EM algorithm. *Neural Computation*, 6(2):334–340, March 1994.

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

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

[34] K. Urahama. Gradient projection network: Analog solver for linearly constrained nonlinear programming. *Neural Computation*, 8(5):1061–1074, 1996.

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

[36] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[37] 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.

[38] L. S. Davis. Shape matching using relaxation techniques. *IEEE Trans. Patt. Anal. Mach. Intell.*, 1(1):60–72, Jan. 1979.

[39] L. Kitchen and A. Rosenfeld. Discrete relaxation for matching relational structures. *IEEE Trans. Syst. Man Cybern.*, 9:869–874, Dec. 1979.

[40] S. Ranade and A. Rosenfeld. Point pattern matching by relaxation. *Pattern Recognition*, 12:269–275, 1980.

[41] K. Price. Relaxation matching techniques—a comparison. *IEEE Trans. Patt. Anal. Mach. Intell.*, 7(5):617–623, Sept. 1985.

[42] 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.

[43] M. Pelillo and A. Jagota. Relaxation labeling networks for the maximum clique problem. *Journal of Artificial Neural Networks*, 2(4):313–328, 1995.

[44] J. Kivinen and M. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Journal of Information and Computation*, 132(1):1–64, 1997.

[45] K. Urahama. Mathematical programming formulations for neural combinatorial optimization algorithms. *Journal of Artificial Neural Networks*, 2(4):353–364, 1996.

[46] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, B*, 48(3):259–302, 1986.

[47] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(6):721–741, Nov. 1984.

[48] S. Li, H. Wang, K. Chan, and M. Petrou. Minimization of MRF energy with relaxation labeling. *Journal of Mathematical Imaging and Vision*, 7(2):149–161, 1997.