LU TP 01-16
May, 2001

Deterministic annealing and nonlinear assignment

Henrik Jönsson* and Bo Söderberg†

Complex Systems Division
Dept. of Theoretical Physics
Lund University
Sölvegatan 14
S-223 62 LUND

Submitted to Physical Review E

For combinatorial optimization problems that can be formulated as Ising or Potts spin systems, the Mean Field (MF) approximation yields a versatile and simple ANN heuristic, Deterministic Annealing.
For assignment problems the situation is more complex – the natural analog of the MF approximation lacks the simplicity present in the Potts and Ising cases.
In this article the difficulties associated with this issue are investigated, and the options for solving them discussed. Improvements to existing Potts-based MF-inspired heuristics are suggested, and the possibilities for defining a proper variational approach are scrutinized.

PACS number(s): 05.10.-a, 75.10.Hk

*henrik@thep.lu.se
†bo.soderberg@thep.lu.se

I. INTRODUCTION

Many mathematical methods originating from theoretical physics have found use in completely different contexts, among them the variational approach to the thermodynamics of complicated systems, lying at the basis of e.g. the mean field approximation to spin systems. This has been successfully used in heuristic methods in the context of combinatorial optimization, for problems that allow a simple formulation in terms of Ising or Potts spins. For other kinds of combinatorial optimization problems, in particular assignment problems, a similar approach is more difficult to achieve; the related difficulties is the focus of this paper.

In an instance of a combinatorial optimization problem, a cost function is defined in terms of a set of discrete variables, and the object is to find an optimal state – a particular state of the variables that minimizes the cost function; in other words the ground state, if the cost function is interpreted as a Hamiltonian. In cases where the variables are of a binary nature, such a problem thus amounts to finding the ground state of an Ising spin system (spin glass) with a given Hamiltonian.

For a small problem instance, an exact method can be used to solve it exactly. In addition to more problem-specific methods, Branch-and-Bound [1] provides a generic class of exact methods, where an intelligent (as opposed to exhaustive) tree-search of the phase-space is performed, disregarding parts that can be ruled out beforehand. Another interesting approach is Simulated Annealing [2], where a standard Monte-Carlo method is used to simulate the immersion of the system in a heat bath, starting at a high temperature, which is slowly lowered (annealing) in the course of the simulation. In the limit of very slow annealing, this stochastic method is guaranteed to yield the ground state as $T \to 0$ [3].

For a large system, however, finding the exact ground state can be a very time-consuming task. For a large class of problems (NP-hard), the expected time required scales worse than any polynomial in the system size, and the quest for the exact ground state must be given up. Instead, one has to resort to more or less dedicated heuristics, to meet the more modest goal of finding states with as low a cost as possible.

Problems that can be formulated in terms of Potts or Ising spins admit a versatile heuristic method, *Deterministic Annealing*, based on the iterative solution of the equations associated with the *mean field* (MF) approximation of the system at hand, combined with a slow decrease in temperature. With the MF variables interpreted as neuron activities, the resulting dynamics at each temperature is that of a generalized Hopfield (or connectionist) network [4]. Deterministic (MF) annealing has been successfully applied to a range of problem types, see e.g. [5–8].

The MF approximation is most conveniently derived from a variational approach, where the proper Boltzmann distribution based on the true Hamiltonian is approximated by a factorized distribution, constrained to be the product of individual single-spin distributions, each of which can be parameterized by the corresponding single spin average. The optimal parameters of the approximating distribution minimize an associated free energy.

Thus, the minimization of the cost function in a discrete phase space is replaced by the minimization of an effective cost function in a continuous parameter space, which in suitable coordinates (the spin averages) interpolates between the discrete states of the original phase space. This gives an advantage as compared to a local optimization method confined to the discrete space, due to the possibility of taking shortcuts.

A somewhat different type of optimization problems is given by *assignment* problems, where an optimal matching (assignment) between two sets of objects is desired, as defined by a given cost function. While certain subclasses of assignment problems, like e.g. *linear assignment* where the cost is linear in the assignment matrix, can be solved exactly in polynomial time, the generic assignment problem is a non-polynomial one.

For nonlinear assignment problems, an obvious generalization of the MF-based deterministic annealing approach is lacking, mainly due to the absence of a simple and natural analog of the MF approximation. While a linear cost appears to be the most sensible choice for a variational Ansatz, it does not lead to the simplicity usually associated with the MF approximation. Nevertheless, it is possible to exploit the linear Ansatz to define dedicated deterministic annealing schemes for non-linear assignment, and we will investigate the difficulties and peculiarities involved in connection with this. A major drawback with this approach, however, is that the time required is exponential in the problem size, and so its practical usefulness is limited.

A popular alternative, to avoid the complexity of such an approach, is to tweak MF annealing as defined for Potts systems to make it apply to assignment problems. We will discuss two common methods of this type, Potts-plus-Penalty [9] and SoftAssign [10], point out their strong and weak points, and where appropriate suggest improvements to the existing state of the art.

To illustrate the implementation on a specific problem type, and to gauge the effect of the suggested improvements, a suitable subset of the methods will be applied to a small testbed of simple applications.

The article is structured as follows: In Sec. II, the basic idea of variational methods in general is described. In Sec. III, MF Annealing for a Potts system is derived from a variational MF approximation, and briefly described. Sec. IV contains a general discussion of assignment problems, and defines some notation. The polynomial problem of Linear Assignment is briefly discussed there. In Sec. V, we discuss the definition of proper deterministic annealing methods dedicated to assignment problems. Sec. VI contains a discussion of existing tweaked Potts-based MF approaches, and suggestions for improvements. In Sec. VII we compare some of the suggested methods on a few simple test problems. Finally, Sec. VIII contains our conclusions.

II. MF IN GENERAL – VARIATIONAL APPROACH

The MF approximation, as it is used in MF annealing for binary and Potts systems, is most conveniently derived from a general variational principle.

Given a complicated cost function $H(s)$ of the variables of interest, s, the idea is to approximate its associated Boltzmann distribution $\propto \exp(-H/T)$ (at a fixed artificial temperature T) with one derived from a simpler cost function $H_V(s,\lambda)$ (e.g. a linear one), with a set of free parameters λ (the coefficients in the linear case). The parameter values are then determined by minimization of the associated free energy $F_V(\lambda)$,

$$F_V(\lambda) = \langle H \rangle - TS \equiv -T \log Z + \langle H - H_V \rangle\,, \qquad (1)$$

where $\langle\cdot\rangle$ stands for an expectation value in the approximating distribution, and Z denotes the corresponding partition function $\sum_s \exp(-H_V(s)/T)$. S is the associated entropy, given by $-\sum_s p(s)\log p(s)$, with $p(s)$ the probability of state s, $p(s) \equiv \exp(-H_V(s)/T)/Z$.

The variational free energy is bounded from below by the true free energy, $F = -T\log(Z_0) = -T\log\sum_s \exp(-H(s)/T)$. The condition for an extremum of F_V with respect to parameter variations $\delta\lambda$ is

$$\delta F_V \equiv \langle \delta H_V\,(H - H_V)\rangle_c = 0, \qquad (2)$$

where $\langle ab\rangle_c$ stands for the *connected* expectation value (or cumulant) $\langle ab\rangle - \langle a\rangle\langle b\rangle$. Thus, for each parameter λ_a, we must have

$$\left\langle \frac{\partial H_V}{\partial \lambda_a}(H - H_V)\right\rangle_c = 0. \qquad (3)$$

Although (3) may permit multiple solutions, including saddle-points or local minima, it is commonly used in the search for an optimal set of parameter values.

Note that since the expectation values involve the variational Boltzmann distribution $\propto \exp(-H_V/T)$ and hence depend on the temperature T, so will the optimal parameter values.

A particularly simple special case results when the variational cost function depends **linearly** on the parameters,

$$H_V(s;\lambda) \equiv \sum_a \lambda_a E_a(s). \qquad (4)$$

Then, eq. (3) for an extremum takes the simple form

$$\sum_b \langle E_a E_b\rangle_c \lambda_b = \langle E_a H\rangle_c\,. \qquad (5)$$

This has the form of a matrix equation, $\langle EE\rangle_c \lambda = \langle EH\rangle_c$, and a straightforward strategy for finding a solution is given by iteratively updating the parameters according to

$$\lambda \to \langle EE\rangle_c^{-1}\langle EH\rangle_c\,, \qquad (6)$$

followed by the corresponding updates of the expectation values $\langle EE\rangle_c, \langle EH\rangle_c$, which depend on the parameters via the Boltzmann distribution.

III. MF ANNEALING FOR POTTS SYSTEMS

In order to understand the problems associated with defining a deterministic annealing approach to assignment-based problems, it is instructive to first review how simpler types of systems are treated.

A simple q-state multiple-choice variable (Potts spin) is conveniently represented by a q-dimensional vector $\mathbf{s}$, with the allowed states represented by the q principal vectors $(1,0,0,\ldots)$, $(0,1,0,\ldots)$, etc., with the position of the single nonvanishing component indicating which state is "on". These vectors are linearly independent, and point to the corners of a regular q-simplex.

As a result, any single-spin cost function can be written as a linear function in $\mathbf{s}$, $H(\mathbf{s}) = \mathbf{C}\cdot\mathbf{s}$, where C_a is the cost associated with state a. For a system of N Potts spins, it follows that an arbitrary cost function can be written in a *multilinear* form,

$$H(s) = a + \sum_{i,a} b_{ia}s_{ia} + \frac{1}{2}\sum_{i,a}\sum_{j\neq i,b} c_{ia,jb}s_{ia}s_{jb} + \ldots, \qquad (7)$$

where s_{ia} denotes the a-th component of the i-th spin. Solving the associated optimization problem corresponds to finding the ground state of the system, i.e. the combination of states of the N Potts spins that minimizes $H(s)$.

The MF approximation to such a system results from a variational approach, corresponding to an optimal approximation of the non-linear cost $H(s)$ in terms of a *linear* one,

$$H_V(s) = \sum_i \mathbf{C}_i\cdot\mathbf{s}_i = \sum_{ia} C_{ia}s_{ia}. \qquad (8)$$

The coefficients C_{ia} constitute the parameters, and are to be chosen so as to minimize the variational free energy F_V. It is convenient to express F_V in terms of the spin averages in the variational distribution, the (*mean field spins*) $\mathbf{v}_i$, with components in $[0,1]$ amounting to

$$v_{ia}(C) \equiv \langle s_{ia}\rangle_V = \frac{\exp(-C_{ia}/T)}{\sum_b \exp(-C_{ib}/T)}. \qquad (9)$$

In the MF approximation, v_{ia} corresponds to the probability for spin i to be in state a, consistently with the identity $\sum_a v_{ia} = 1$. The MF spins thus interpolate between the discrete states of the original spins; in terms of them the variational free energy evaluates to

$$F_V(v) = T \sum_{ia} v_{ia} \log v_{ia} + H(v), \tag{10}$$

which can be minimized with respect to the normalized MF spins by adding a Lagrange parameter λ_i for the normalization of each MF spin $\mathbf{v}_i$. The condition for a extremum, equivalent to eq. (3) amounts to

$$dH(v)/dv_{ia} + T(1 + \log v_{ia}) = \lambda_i, \tag{11}$$

which, together with the normalization that fixes the λ_i values, gives the variational coefficients C up to an unimportant constant in terms of v as

$$C_{ia}(v) = \frac{\partial H(v)}{\partial v_{ia}}. \tag{12}$$

Eqs. (9) and (12) define the *MF equations*.

The MF approximation corresponds to neglecting the correlations between the different spins, since the linear variational Ansatz used is the most general factorized distribution $P_V(s) = \prod_i p_i(s_i)$, where the different Potts spins obey independent distributions.

MF annealing corresponds to solving the MF equations iteratively, starting with a high T, where a fixed point with $v_{ia} \approx 1/N$ will dominate, and slowly lowering T. At low enough T, the MF spins will be forced *on shell*, i.e. for $v_{ia} \approx s_{ia} \in 0, 1$, and a suggested solution can be extracted.

IV. ASSIGNMENT PROBLEMS – GENERAL DISCUSSION

A. Notation

When it comes to permutation/assignment problems, we have to distinguish between *single assignment* problems and *multiple assignment* problems, the latter being based on several assignments.

To begin with, we will consider the simpler case of a single assignment, where an optimal matching between two sets of N objects is desired, i.e. for each object i in the first set, an object a in the second set is to be chosen, such that different i are assigned to different a. There are obviously $N!$ ways to accomplish this.

A compact way to describe an assignment is by means of the associated *assignment matrix*, i.e. an $N \times N$ doubly stochastic matrix s with elements in $\{0, 1\}$, such that $s_{ia} = 1$ if i is assigned to a, and 0 otherwise (for a somewhat different encoding of assignment problems as used in deterministic annealing, see [8]). Obviously, we must have precisely one unit element in each row as well as in each column of s, consistently with

$$\sum_a s_{ia} = 1, \; \sum_i s_{ia} = 1. \tag{13}$$

Then, a given single assignment problem can be described in terms of a cost function $H(s)$, which is to be minimized.

Note, however, that, in contrast to e.g. the Potts case, the most general cost function for single assignment (for $N > 2$) is **not** linear in s (see Appendix); in the worst case a polynomial of degree $N - 1$ is needed.

Alternatively, the cost function can be viewed as an explicit function over the permutation group P_N: Each group element g is associated with an individual cost C_g, defining an element of an $N!$-dimensional *cost vector* $\vec{C}$. The relation to the formulation in terms of the assignment matrices s is $C_g = H(s_g)$, where s_g is the particular assignment matrix representing g.

B. Thermodynamics for a single nonlinear assignment

The thermodynamics af a system consisting of a single $N \times N$ assignment with an arbitrary cost function is not difficult to express, when formulated in terms of a cost vector $\vec{C}$ over group space. The assignment then acts as a single $N!$-state Potts spin, that can be described by an $N!$-dimensional vector $\vec{S}$ with precisely one unit component, while the rest is zero: $S_g \in \{0, 1\}$, $\sum_g S_g = 1$.

At an artificial temperature T, the probability of a particular state g amounts to

$$V_g = \langle S_g \rangle = \frac{\exp\left(-C_g/T\right)}{\sum_h \exp\left(-C_h/T\right)}. \tag{14}$$

As $T \to 0$, the distribution gets increasingly concentrated at the state (group element) with the lowest cost.

When viewed this way, the difficulty lies entirely in the huge number of states involved if N is large. In order to compute one component of $\vec{V}$, one has to know the costs for all the $N!$ states. For a generic cost function, the associated computational complexity is non-polynomial in the size N of the system.

Thus, for a generic large assignment problem, one has to make do with some kind of heuristic.

C. Linear assignment

Certain classes of assignment problems can be solved exactly in polynomial time. One such class is linear assignment, where the cost function is constrained to be linear in the assignment matrix s,

$$H(s) = \sum_{ij} c_{ij} s_{ij}, \tag{15}$$

defining an $N \times N$ *cost matrix* c.

This problem corresponds essentially to a linear programming one, and can be solved in polynomial time, using e.g. the so called *Hungarian algorithm* [11,1], based on the fact that the addition of terms to c depending on row or column alone, $c_{ij} \to c_{ij} + a_i + b_j$, is equivalent to adding a constant to the cost function, $H(s) \to H(s) + \sum_i a_i + \sum_j b_j$. This is used to iteratively modify the cost matrix until it takes a form where it has zeros on a set of elements corresponding to the optimal assignment, and non-negative values elsewhere.

Unfortunately, this doesn't help when computing thermal averages at a finite T; this is still a non-polynomial task. Thus, e.g., the expectation value of s_{ij} is given in terms of a matrix M, obtained from c by elementwise exponentiation,

$$M_{ij} = \exp(-c_{ij}/T), \tag{16}$$

as

$$v_{ij} \equiv \langle s_{ij} \rangle = \frac{M_{ij} P_{ij}(M)}{P(M)}. \tag{17}$$

Here, $P(M)$ is the *permanent* [12] of M; it has some similarities to the determinant, being the sum of all the possible products of N elements in M, one in each row and one in each column, but with *no minus signs* in contrast to the case for the determinant. Similarly, $P_{ij}(M)$ is the *subpermanent* of M, obtained by removing row i and column j from M, and computing the permanent of the remaining $(N-1) \times (N-1)$ matrix.

The expression for v_{ij} in eq. (17) can be derived from

$$\langle s_{ij} \rangle = -\frac{T}{Z} \frac{\partial Z}{\partial c_{ij}}, \tag{18}$$

where Z is the partition function,

$$\begin{aligned} Z &= \sum_g \exp\left(-\sum_{ij} c_{ij} s_{ij}(g)/T \right) \\ &= \sum_g \exp\left(-\sum_{ij \in g} c_{ij}/T \right) \\ &= \sum_g \prod_{ij \in g} \exp\left(-c_{ij}/T\right) = \sum_g \prod_{ij \in g} M_{ij} = P(M), \end{aligned} \tag{19}$$

where the restriction $ij \in g$ in the sum over ij means that row i and column j are matched in $s(g)$ (so $s_{ij}(g) = 1$). The derivative of $Z = P(M)$ with respect to M_{ij} yields $P_{ij}(M)$, which completes the proof of eq. (17).

The combination $M_{ij} P_{ij}(M)$, appearing in eq. (17), gives the sum of those terms in the permanent that contain the element M_{ij}. Summing this over i or j yields $P(M)$, ensuring that eq. (17) yields a doubly stochastic matrix v.

Similarly, the expectation value of the product of two elements of s becomes

$$\langle s_{ij} s_{kl} \rangle = \delta_{ik} \delta_{jl} \frac{M_{ij} P_{ij}(M)}{P(M)} + \frac{M_{ij} M_{kl} P_{ik,jl}(M)}{P(M)}, \tag{20}$$

where $P_{ik,jl}(M)$ is a subpermanent obtained as the permanent of the submatrix where rows i,k and columns j,l are removed, if $i \neq k$ and $j \neq l$; else it is zero. Thus $M_{ij} M_{kl} P_{ik,jl}(M)$ sums up the terms in $P(M)$ that contain $M_{ij} M_{kl}$.

As an aside, replacing the permanents by determinants in the expression (17) for v would lead to the combination $D_{ij}(M)/D(M)$, exactly corresponding to the j,i element of the matrix inverse of M. The elementwise product with M would yield a doubly (quasi-)stochastic v, where row and column sums are equal to one, albeit with elements of both signs.

However, while the determinant of a matrix can be calculated in polynomial time, the permanent in general can not, in spite of their similarity – the computational time required to compute a generic $N \times N$ permanent is exponential in N (roughly $\propto 2^N$ using e.g. Ryser's method [13,12]).

V. PROPER VARIATIONAL METHOD FOR A SINGLE NONLINEAR ASSIGNMENT

For a large generic single assignment problem, an exact solution is out of reach, and one has to make do with heuristic methods. One possibility then is to consider a deterministic annealing approach based on approximating the true cost function by a variational one that is simpler.

A. Linear Ansatz for H_V

The most natural Ansatz for the variational cost function H_V is a linear one,

$$H_V(s) = \sum_{ij} c_{ij} s_{ij}, \tag{21}$$

with the coefficients c_{ij} as free parameters.

The equation (3) for a minimum of the variational free energy then yields:

$$\sum_{kl} \langle s_{ij} s_{kl} \rangle_c c_{kl} = \langle s_{ij} H \rangle_c . \tag{22}$$

In analogy with eq. (6), this is a matrix equation, from which c can be formally extracted as

$$c = \langle ss \rangle_c^{-1} \langle sH \rangle_c . \tag{23}$$

Note that the $N^2 \times N^2$ matrix $\langle ss \rangle_c$ is not fully invertible; it always has $2N-1$ zero-modes corresponding to the addition of redundant terms to c depending on row or column index alone. These merely yield row and column factors in the exponentiated matrix M, and are of no importance for expectation values.

If H is a low-order polynomial in s, a solution to eq. (22) can in principle be computed iteratively in analogy to the iterative solution of the Potts MF eqs. (9, 12), by repeating the two steps

1. Calculate the expectation values appearing in eq. (22); they depend on the present c via M and its permanent (and subpermanents), where $M_{ij} = \exp(-\beta c_{ij})$, in analogy to eqs. (17,20).

2. Obtain an updated cost matrix c by means of eq. (23), suitably regularized with respect to the zero-modes of $\langle ss \rangle_c$. [‡]

This can be turned into an annealing approach, by starting with a high T (low β), and decreasing T slightly after every step, until the "MF variables" $v_{ij} \equiv \langle s_{ij} \rangle$ have stabilized sufficiently close to zero or one.

A major drawback of this approach is that it is only feasible if N is not too large, since the computation of expectation values involves the computation of permanents, which requires a time exponential in N.

B. Quadratic H

The simplest non-linear function of s is quadratic, so assume H to be a given quadratic plus linear function in s,

$$H(s) = \frac{1}{2} \sum_{ijkl} A_{ijkl} s_{ij} s_{kl} + \sum_{ij} B_{ij} s_{ij}, \tag{24}$$

involving a symmetric tensor A, $A_{ijkl} = A_{klij}$, which can be assumed to vanish for $i=k$ or $j=l$.

The variational eq. (22) corresponding to a linear $H_V(s)$ becomes

$$\sum_{kl} \langle s_{ij} s_{kl} \rangle_c c_{kl} = \frac{1}{2} \sum_{klmn} \langle s_{ij} (s_{kl} s_{mn}) \rangle_c A_{klmn} + \sum_{kl} \langle s_{ij} s_{kl} \rangle_c \left(\sum_{mn} A_{klmn} \langle s_{mn} \rangle + B_{kl} \right). \tag{25}$$

Denoting the first term on the RHS of eq. (25) by F_{ij}, the effective cost matrix c can be formally extracted as $c = B + A \langle s \rangle + \langle ss \rangle_c^{-1} F$, using a suitably regularized matrix inverse.

C. Group theoretical aspects

It is instructive to view an assignment problem from a group-theoretical point of view, where the relevant group of course is the permutation group of N elements, denoted by P_N.

Like any functions over group space, H and H_V can be expressed in a unique way as linear combinations of the matrix elements of the irreducible representation matrices of P_N (see Appendix for details).

Requiring H_V to be linear in s means that its expansion is constrained to contain only elements from the trivial and the fundamental irreducible representations, **e** and **f**; thus, it can be written in a unique way in the form

$$H_V(g) = A + \sum_{ij} B_{ij} u^f_{ij}(g), \tag{26}$$

where f stands for the fundamental representation. This leads to the probability $V_g \propto \exp(-H_V(g)/T)$, such that $\sum_g V_g = 1$, for a particular group element g. The corresponding version of the variational eq. (3) becomes

$$A + \sum_{kl} B_{kl} \sum_g V_g u^f_{kl}(g) = \sum_g V_g H(g), \tag{27}$$

$$A \sum_g V_g u^f_{ij}(g) + \sum_{kl} B_{kl} \sum_g V_g u^f_{ij}(g) u^f_{kl}(g) = \sum_g V_g u^f_{ij}(g) H(g), \tag{28}$$

corresponding exactly to respectively the trivial and the nontrivial parts of eq. (22). The trivial part A can be eliminated from (27) and inserted into (28), yielding

$$\sum_{kl} \left(\sum_g V_g u^f_{ij}(g) u^f_{kl}(g) - \sum_g V_g u^f_{ij}(g) \sum_h V_h u^f_{kl}(h) \right) B_{kl} = \sum_g V_g u^f_{ij}(g) H(g) - \sum_g V_g u^f_{ij}(g) \sum_h V_h H(h), \tag{29}$$

which is nothing but a disguised version of eq. (22); the sums over g with V_g as a weight correspond to averages.

For the special case (24) of a quadratic $H(s)$, the corresponding $H(g)$ is constrained to include elements from **e**, **f** and two additional representations, **a** and **s** (see Appendix), in its irrep expansion, with dimensions $d_a = (N-1)(N-2)/2$ (if $N > 2$), and $d_s = N(N-3)/2$ (if $N > 3$), respectively.

Although the above analysis illuminates the linear variational approach from a group-theoretical point of view, the resulting formulation is not of an immediate practical interest for large N, since it (at least formally) requires the complete enumeration of the $N!$-dimensional group space.

[‡]In cases of instability, the change in c can be decreased by e.g. a factor of 1/2.

D. Proper variational methods for multiple nonlinear assignment

Here we will briefly discuss the possibilities for treating systems of several distinct assignments in a variational approach.

1. General additive Ansatz

In the case of a generic cost function of several assignments, the most natural choice is to consider a generic *additive* Ansatz for a variational cost function, corresponding to a factorized Boltzmann distribution. In principle, this corresponds to the MF approximation to a system of $N!$-state Potts variables, and can be treated as such. This can be useful, even for large systems (many distinct assignments), as long as the individual assignments are of low dimensionalities.

2. Linear Ansatz

A further simplification results from requiring that the variational cost function not only be additive in the different assignments, but also that the contribution from each assignment be linear.

A possible strategy then is to update the cost matrix for one assignment at a time according to eq. (23), considering the single-assignment averages associated with other assignments as constant, and recomputing the single-assignment averages associated with the updated cost matrix before moving on to update the next assignment.

3. Multilinear cost, linear Ansatz

A particularly simple special case of the above is when the exact cost function is a *multilinear* function of the assignments matrices s. For the case of two assignments, $s^{(1)}$ and $s^{(2)}$, this means a cost function H of the form

$$H(s^{(1)}, s^{(2)}) = \sum_{ij} \sum_{kl} A_{ij,kl} s^{(1)}_{ij} s^{(2)}_{kl} + \sum_{ij} B^{(1)}_{ij} s^{(1)}_{ij} + \sum_{ij} B^{(2)}_{ij} s^{(2)}_{ij}. \tag{30}$$

Then an additive variational cost function automatically will be linear

$$H_V(s^{(1)}, s^{(2)}) = \sum_{ij} C^{(1)}_{ij} s^{(1)}_{ij} + \sum_{ij} C^{(2)}_{ij} s^{(2)}_{ij}. \tag{31}$$

The resulting updates then amount simply to

$$\begin{aligned} C^{(1)}_{ij} &= \sum_{kl} A_{ij,kl} \left\langle s^{(2)}_{kl} \right\rangle + B^{(1)}_{ij}, \\ C^{(2)}_{kl} &= \sum_{ij} A_{ij,kl} \left\langle s^{(1)}_{ij} \right\rangle + B^{(2)}_{kl}, \end{aligned} \tag{32}$$

where variational expectation values are understood, computable in terms of permanents and sub-permanents of the respective cost matrices.

The generalization to several assignments is straightforward.

VI. POTTS-BASED MF HEURISTICS FOR NONLINEAR ASSIGNMENT

Although a proper variational approach as described above appears to be the natural choice for constructing a deterministic annealing approach for assignment problems, a major problem is the computational complexity involved in the required computation of permanents and subpermanents. Indeed, for certain problem classes an instance can be solved exactly in the time it takes to compute a single permanent. This implies that these methods have a rather limited applicability.

Instead, alternative methods based on Potts spins have been used to construct faster deterministic annealing methods for non-linear assignment problems.

A. Potts plus Penalty

One such method is based on viewing the assignment matrix s as an array of N-state Potts spins, one for each row. Then the row condition, $\sum_a s_{ia} = 1$, is automatically fulfilled. One then adds to H a penalty term for breaking of the column normalization, and treats the result using Potts MF annealing, based on the modified cost function

$$H(s) + \frac{\alpha}{2} \sum_a \left(1 - \sum_i s_{ia} \right)^2, \tag{33}$$

with the penalty strength α suitable adjusted. (Of course, one might equally well consider the columns as Potts spins and add penalties for the rows.)

This approach, in what follows referred to as **PPP** for Potts plus Penalty, has been successfully applied to a number of different problems [5]. The problem with a soft penalty is that it involves a delicate tuning of the coefficient α; too small, and improper assignments result, where two rows are mapped to the same column; too large, and the cost is too dominated by the penalty term, with a consequent deterioration in performance.

In PPP, the MF spins are preferably updated in a serial manner, one row at a time. This leads to the most stable MF dynamics, provided H is put in multilinear form. It is easy to see then that the Potts free energy (10) becomes a Lyapunov function of the dynamics at a fixed temperature. This ensures that the MF dynamics is well-behaved in the high-T domain, with the trivial high-T fixed point losing stability in a controlled manner. It also guarantees that PPP turns into a form of local optimization in the low-T limit, ensuring the stability of an optimal assignment.

B. SoftAssign

An obviously ugly feature of the PPP approach is the asymmetry in the treatment of rows and columns of s. This can be cured in a slightly more advanced Potts-inspired method, the *SoftAssign* (or "Double Potts") approach [10], which can be derived as a somewhat *ad hoc* improvement to PPP as follows.

In PPP, the resulting MF average is given by $v_{ij} = a_i M_{ij} b_j$, where M_{ij} is given by $\exp(-(\partial H/\partial v_{ij})/T)$ and the column factor b_j comes from the penalty term, $b_j = \exp(\alpha(1 - \sum_i v_{ij})/T)$. In contrast, the row factor is the usual automatic Potts normalization factor, ensuring the exact normalization of rows.

The idea in SoftAssign is to skip the penalty, and freely choose positive row and column factors so as to force the exact normalization of both rows and columns. This leads to the following problem: Given a matrix M with non-negative elements, find vectors of row and column factors, a and b, such that the result,

$$v_{ij} \equiv a_i M_{ij} b_j, \tag{34}$$

is a doubly stochastic matrix.

This in fact determines v uniquely, which can be seen by defining $x_i = \log a_i$, $y_j = \log b_j$ (a_i, b_j are assumed positive), and noting that the correct x, y minimize the strictly convex function $f(x,y) \equiv \sum_{ij} e^{x_i} M_{ij} e^{y_j} - \sum_i (x_i + y_i)$.

However, the proper row and column factors can not be obtained as simple, closed expressions in the matrix elements of M. Instead, the desired doubly stochastic matrix v is usually obtained by iteratively modifying M, alternatingly normalizing rows and columns until the resulting matrix is sufficiently close to being correctly normalized:

$$\textbf{i)}\ M_{ij} \rightarrow \frac{M_{ij}}{\sum_k M_{ik}},$$

$$\textbf{ii)}\ M_{ij} \rightarrow \frac{M_{ij}}{\sum_k M_{kj}},$$

$$\textbf{iii)}\ \text{Go to } \textbf{i)}.$$

This procedure, which is due to Sinkhorn [14], is guaranteed to yield convergence to a unique doubly stochastic matrix v.

For a nonlinear problem, we can obviously identify the derivatives $\partial H/\partial v_{ij}$ with the elements of an estimated *effective cost matrix*, obtained by linearizing the cost-landscape in the vicinity of the present point. Note that, for a *linear* assignment problem, SoftAssign leads to exactly the same initial M as in eq. (17); but that a different doubly stochastic matrix v is derived from it in eq. (34).

As an aside, the SoftAssign approach can formally be associated with the minimization of an entity reminiscent of a free energy,

$$F(v) = T \sum_{ij} v_{ij} \log v_{ij} + H(v), \tag{35}$$

with v constrained to be doubly stochastic, e.g. by means of adding suitable Lagrange terms (with Lagrange multipliers associated with the row and column factors). Although $F(v)$ is superficially highly analogous to the Potts free energy in eq. (10), SoftAssign does *not* correspond to a proper variational approch to approximating the true $H(s)$, mainly because the first term is not $-T$ times the proper entropy. Nor does v correspond to the expectation value of s in an approximating Boltzmann distribution.

Note that for SoftAssign (unlike the case with a proper variational approach), the resulting dynamics is sensitive to the precise formulation of $H(v)$ as an extrapolation of $H(s)$ to continuous arguments v.

Nevertheless, SoftAssign seems theoretically more appealing than PPP, in treating rows and columns in a symmetric manner, and in guaranteeing a doubly stochastic v, and it has also been successfully applied to various types of problems [6], although the method does have some weak points as will be discussed below.

1. Weak points with SoftAssign

In the SoftAssign approach, the iterative Sinkhorn procedure for normalizing v is problematic at low T. This can be seen by assuming one has reached a stage where the matrix is close to being doubly stochastic:

$$M_{ij} = (1 + \alpha_i) v_{ij} (1 + \beta_j), \tag{36}$$

where v is the desired doubly stochastic matrix, while α_i, β_j are assumed small. To linear order in α, β, one step of row normalization corresponds to $\alpha \rightarrow -v\beta$ in matrix notation, while one step of column normalization yields $\beta \rightarrow -v^\top \alpha$. Together, this gives $\beta \rightarrow v^\top v \beta$.

Hence, the asymptotic rate of convergence is determined by the eigenvalues of the positive-definite matrix $U = v^\top v$. At a high temperature, U will be close to a uniform matrix with $1/N$ everywhere, while at a low T, it will be close to the identity matrix. Consider a simple Ansatz for U: $U_{ij} = (1-a)\delta_{ij} + a/N$, with $0 \leq a \leq 1$. The limit $a \rightarrow 1$ corresponds to high T, while $a \rightarrow 0$ emulates low T. U can also be written as $U = (1-a)\mathbf{1} + aP$, where $\mathbf{1}$ is the identity matrix, and P is the projection matrix onto the uniform vector.

U has two distinct eigenvalues: a single unit eigenvalue with a uniform eigenvector $(1, 1, \ldots, 1)$, and an $(N-1)$-fold degenerate value $1-a$ with eigenvectors in the transverse space. The unit eigenvalue is to be expected, reflecting the irrelevance of shuffling a global factor between a and b.

What is worse is that as $T \to 0$ ($a \to 0$), also the other eigenvalue, $1-a$, tends towards unity. This is not a special feature of this particular Ansatz for v, but a generic phenomenon: As v approaches a proper assignment matrix, $v^\top v$ approaches the identity matrix. This means that the normalization procedure inevitably runs into convergence problems in the limit of low T.

Another drawback as compared to PPP is that in SoftAssign, the elements of the assignment matrix v have to be updated in *synchrony*, and there is no obvious simple way to update it, say, one row at a time. As a result, stability is not guaranteed at low T even for a good solution, unless the cost function H is carefully tuned.

2. Speeding up the iterative normalization

In order to improve convergence for the normalization procedure in SoftAssign at low T, it appears important to initialize the multiplicative row and column vectors a and b carefully, so as to leave as little as possible for the slow iterative procedure to do.

Obviously, to avoid overflow on the computer upon implementation of eq. (16), the effective cost matrix will have to be modified with suitable row and column additions, $c_{ij} \to c_{ij} + \alpha_i + \beta_j$, to ensure that the smallest elements be zero and the rest positive. This can be done, e.g., by first subtracting from the elements in each row the lowest element in that row, and then subtracting from the elements in each column the lowest element in that column.

This measure does not suffice, however, to guarantee a proper starting point. Consider $N = 3$ and the cost matrix $c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$. At zero T this yields $M = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$, and the normalization procedure will be caught in an eternal loop, alternating forever between the two states $\begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 0 & 0 \end{pmatrix}$. In fact, this M is not normalizable with finite row and column factors!

Obviously, it is not enough to ensure at least one zero in each row and each column of c. The zeros must be arranged such that at least one combination of them will correspond to a one-to-one matching between rows and columns. Finding such a modification with otherwise non-negative elements corresponds precisely to solving the associated (effective) linear assignment problem.

Thus, we suggest using e.g. the Hungarian method to transform c to the proper form, in order to guarantee a normalizable M for $T \to 0$.

Even this step will not guarantee a fast convergence. As a second example, consider $N = 2$ and an effective cost matrix $c = \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}$, obviously in the proper form. The corresponding M at $T = 0$ will be $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$. If this is handed to the normalization procedure at a low T, the approach to the final doubly stochastic matrix, $v = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, will be merely harmonic.

The situation is considerably improved, by in addition carefully *balancing* the cost matrix c, to ensure also a maximal number of non-zero elements (while maintaining a sufficient set of zeros), with the smallest of these being as large as possible. This can be done in polynomial time, using a recursive procedure. For the $N = 2$ case above, the balancing step will yield $c = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, giving $M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ at zero T, and convergence is immediate.

Still, there will be cases for $N > 2$ where the procedure will be caught in a slowly converging sequence – this is inevitable, due to the unit eigenvalues at zero T –, but in this way the most obvious traps are avoided.

An additional improvement is possible when using the Hungarian algorithm for preprocessing the cost matrix, by exploiting that it gives a preferred matching. This can be used to modify the normalization process to improve the convergence rate by updating of matched row-column pairs simultaneously. Especially in the low T region, where the matched elements unambiguously define a selected assignment, and the corresponding elements of v will be close to unity and the rest small, this noticeably speeds up the normalization.

Thus, the normalization constraints for a coupled row-column pair i, j of a modified M read

$$
\begin{aligned}
a_i \left(\sum_{k \neq j} M_{ik} + M_{ij} b_j \right) &= 1, \\
b_j \left(\sum_{k \neq i} M_{kj} + a_i M_{ij} \right) &= 1,
\end{aligned}
\tag{37}
$$

and are simultaneously satisfied by $a_i = x/A_i$ and $b_j = x/B_j$, where $A_i = \sum_{k \neq j} M_{ik}$, $B_j = \sum_{k \neq i} M_{kj}$ and

$$
x = \frac{\sqrt{A_i B_j \left(4 M_{ij} + A_i B_j\right)} - A_i B_j}{2 M_{ij}}. \tag{38}
$$

Each matched row-column pair of M in turn is updated in this manner and the process is repeated until the result is close to being a doubly stochastic matrix. We will refer to this normalization scheme as *coupled normalization*.

3. Ensuring a stable dynamics

The second major problem with SoftAssign is the lack of a guarantee for the stability of a solution in the low-T limit; this problem never appears e.g. in a properly handled system of Potts MF spins with serial updating.

To remedy this, we will have to exploit the freedom of adding terms to H which are trivial on shell, but nevertheless affect the MF dynamics. One would at least like to ensure that in the low-T limit, SoftAssign turns into some form of local optimization.

One possibility then is to demand that $H(v)$ be transformed into a *concave* function of v (in the subspace consistent with v being doubly stochastic). This guarantees that the energy in eq. (35) for $T \to 0$ will have a local minimum in the corner corresponding to an optimal assignment.

A crude way to ensure concavity is to add a negative quadratic diagonal term $-(\alpha/2)\sum_{ij} s_{ij}^2$ to H, with a large enough coefficient α. For a quadratic H in particular, concavity also ensures that the SoftAssign free energy (35) becomes a Lyapunov function of the dynamics at a fixed temperature [15]. A disadvantage with this method is that also non-solutions will be stabilized. Empirically, a smaller diagonal addition often suffices to stabilize the dynamics.

A more advanced possibility is to employ problem-specific modifications of H, adding suitable terms to ensure the stability of a good solution. It is therefore of interest to analyze what kinds of generic additions are possible without altering the on-shell costs (or merely adding a constant). We will refer to such terms as being **redundant**.

Linear redundant terms: Based on decomposing the defining representation of the permutation group P_N, given by s, into the direct sum of the trivial 1-dimensional irrep $\mathbf{e}$ and the fundamental $(N-1)$-dimensional one $\mathbf{f}$ (see Appendix), the only possible linear redundant additions are given by combinations of row or column sums of s; in SoftAssign, such additions merely lead to a modification of the initial row or column factors a, b in eq. (34), and have no effect on the resulting doubly stochastic matrix v.

Combinations of quadratic and linear terms: The possibilities here stem from the decomposition of the reducible representation of P_N defined by the symmetric direct product of the defining representation ($\Leftrightarrow \mathbf{e}+\mathbf{f}$) with itself, as discussed in the Appendix. The contributions stemming from the trivial direct product of the $\mathbf{e}$ part with itself or with $\mathbf{f}$ give nothing useful, corresponding to quadratic terms involving row or column sums of s, completely equivalent to the corresponding linear terms with the row or column sums replaced by 1.

The possibly interesting terms come from $\mathbf{f}\times\mathbf{f} = \mathbf{e}+\mathbf{f}+\mathbf{s}+\mathbf{a}$. As discussed in the Appendix, the symmetric part of this consists of a part $\mathbf{a}$, antisymmetric in both row and column indices, yielding no redundant terms, and a part containing $\mathbf{e}+\mathbf{f}+\mathbf{s}$, symmetric in both row and column indices, which yields quadratic terms that vanish on shell, or equal a constant, or a linear combination of the elements of s. In a slighly disguised form, they correspond to the on-shell identities

$$
\begin{aligned}
s_{ij}^2 - s_{ij} &= 0, \\
s_{ij}s_{il} &= 0, \quad \text{for } j \neq l, \\
s_{ij}s_{kj} &= 0, \quad \text{for } i \neq k.
\end{aligned}
\tag{39}
$$

Group theory certifies that these identities suffice to generate all possibly useful redundant additions to $H(s)$, that are at most quadratic in s. Although such additions to H are identically zero on shell ($v \to s$), as additions to $H(v)$ they will alter the properties of the dynamics in SoftAssign, in terms of a modification of the expression for the effective cost matrix $c = \partial H(v)/\partial v$.

Thus, the addition of a term proportional to the square of an element v_{ij} minus the element itself, modifies only the corresponding element of c, by an addition proportional to $2v_{ij} - 1$. Adding the product of two elements of s in the same row i but in different columns j, l affects the corresponding two elements of the effective cost matrix, with an addition to c_{ij} proportional to v_{il}, and vice versa. Analogously, adding to H a term involving the product of two elements in the same column j but in different rows i, k yields the addition to c_{ij} of a term proportional to v_{kj} and vice versa.

4. TSP-specific modifications

In certain quadratic problems, such as the *travelling salesman problem* (TSP), where a set of N *sites* is to be cyclically visited in an optimal order, the cost function has the particular structure

$$
H(s) = \frac{1}{2}\mathrm{Tr}\left(sDs^\top X\right), \tag{40}
$$

with D, X a pair of symmetric $N \times N$ matrices, vanishing on the diagonal.

For TSP, D is the pair-distance matrix, with D_{ab} defining the distance between sites a, b, while X defines the cyclic tour sequence neighborhood, $X_{ij} = \delta_{i,j+1} + \delta_{i,j-1}$, such that H measures the total *tour length*.

Concavity in the subspace consistent with s being doubly stochastic, of the direct product matrix $A = D \times X$, then corresponds to one of D or X being positive-semidefinite, and the other negative-semidefinite, each in the subspace orthogonal to $e = (1, 1, 1 \ldots 1)$. This can be ensured by suitable diagonal additions to D and X separately,

$$
D \to D + \alpha\mathbf{1}, \; X \to X + \gamma\mathbf{1}, \tag{41}
$$

with α and γ of opposite signs. The diagonal additions to D and X implies adding terms to H of the form discussed in the previous subsection. All vanish on shell except for the $\alpha \times \gamma$ term, which evaluates to a simple constant.

For the case of TSP, often D is already negative-definite in the transverse subspace, and a suitable addition to X suffices. X is easily diagonalized by means of a discrete Fourier transform, with the spectrum given by $\lambda_k = 2\cos(2\pi k/N)$, for $k = 1, \ldots, N-1$. Thus, $\gamma = 2$ is required to make the modified X positive-semidefinite.

In practice, however, $\gamma = 1$ will suffice to stabilize the dynamics in most cases; in the low-T limit this is just enough to secure the stability of assignments locally optimal with respect to local changes in the ordering of visited sites.

VII. TESTS ON SIMPLE APPLICATIONS

In order to illustrate the ideas discussed in the previous section, we will here test the various improvements to the SoftAssign algorithm, as applied to a set of small single-assignment problems.

The effects of the improvements to the normalization algorithm at low temperatures are illustrated using the linear assignment problem. For TSP, the use of a problem-specific stabilizing term is compared to employing a negative quadratic term $-(\alpha/2)\sum_{ij} s_{ij}^2$.

The SoftAssign algorithm used is described in Fig. 1.[§] All experiments have been performed on an 800MHz PentiumIII computer running Linux.

- Initiate the elements of v to random values close to $1/N$, and T to a high value.
- Repeat the following (a sweep), until the v matrix has *saturated* (i.e. become close to a (0,1)-matrix):
  - Calculate the effective cost matrix by means of $c_{ij} = \partial H/\partial v_{ij}$, possibly modify it with suitable row/column additions, and let $M_{ij} = exp(-c_{ij}/T)$.
  - Normalize M with the proper row and column factors to yield a doubly stochastic matrix, defining an updated v.
  - Decrease T slightly (typically by a few percent).
- Extract the resulting solution candidate.

FIG. 1. A SoftAssign algorithm

A. Speeding up the iterative normalization

As discussed in section VI B 1, the Sinkhorn normalization of M, eq. (35), runs into trouble when the temperature is low and the corresponding v is close to an on-shell assignment matrix.

To probe the efficiency of each normalization scheme, we use random linear assignment, eq. (15), where the costs $c_{ij} \in [0, 1]$ are uniform random numbers. We investigate the number of iteration steps needed before row and column sums of the modified M matrix are in the range $1 \pm \Delta_{max}$ with Δ_{max} a small number, and measure the time used by the normalization scheme. This is done at a set of decreasing temperatures such that $v_{ij} \approx 1/N, \forall i, j$ for the higher values of T, while v is nearly on shell for the lower T values.

We compare the following schemes described in section VI B 2.

1. Plain Sinkhorn. Preprocess c by first for each row subtracting the smallest element, and then doing the same for each column. Then the Sinkhorn row and column normalization (eq. (35)) is applied on the resulting M.
2. Hungarian+Sinkhorn. Preprocess c using the Hungarian algorithm. Then normalize M using Sinkhorn.
3. Hungarian+Balancing+Sinkhorn. Preprocess c using the Hungarian and the balancing algorithms. Sinkhorn normalization of M.
4. Hungarian+CoupledNorm. Preprocess c as in 2, then apply the coupled row-column normalization described in section VI B 2.
5. Hungarian+Balancing+CoupledNorm. Same preprocessing of c as in 3, then coupled row-column normalization.

Figures 2 and 3 show statistics from 100 linear assignment problems of size $N = 100$. The data is binned for different values of the *saturation*, $\Sigma = (1/N)\sum_{ij} v_{ij}^2$, representing different temperature regions. At high temperatures the saturation is close to $1/N$, while it approaches one in the low temperature region. The annealing is continued until the saturation becomes larger than 0.999, but is also aborted when the number of normalization steps exceeds a maximal value of 20000 iteration steps for three consecutive temperatures (which only happened for the plain Sinkhorn approach as discussed below); these data points are not included when calculating the averages in Figs. 2 and 3.

The results illustrate the efficiency of the different normalization methods used on M, and the effect of preprocessing of the cost matrix c.

[§]For a more thorough description of SoftAssign in general, we refer to [10]

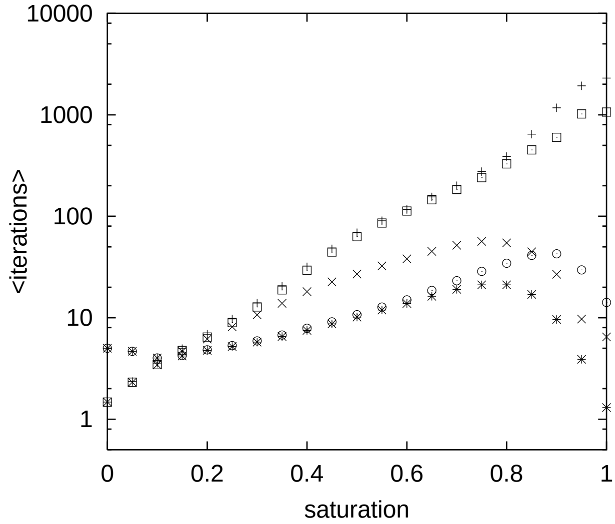

FIG. 2. Number of normalization iterations used versus saturation. The data consist of averages from runs on 100 random linear assignment problems, binned into different values of the saturation. The normalization procedure is continued until all row and column sums are within $1 \pm \Delta_{max}$ with $\Delta_{max} = 0.01$. The plot shows Plain Sinkhorn (+), Hungarian+Sinkhorn (□), Hungarian+CoupledNorm (⊙), Hungarian+Balancing+Sinkhorn (×), and Hungarian+Balancing+CoupledNorm (*).

As can be seen in Fig. 2 the number of iterations needed with plain Sinkhorn grows by several orders of magnitude in the low temperature region, where the saturation approaches 1. What is not revealed in the figure is that the plain Sinkhorn scheme failed to normalize M at low temperatures in *all* of the tested problem instances – because of the limited resolution on the computer, small elements in M become zero where the corresponding elements in v should be one. The Sinkhorn scheme then gets stuck in an eternal loop, failing to produce a doubly stochastic v. The failure occurred at high enough temperatures that the saturation was below our limit of 0.999, and the algorithm had to be aborted as described above.

This problem can be avoided by either interrupting the annealing at an earlier stage, and extracting a solution from the unsaturated v matrix, or by adding additional redundant terms to the Hamiltonian, at the cost of a lower performance. However, to guarantee that an arbitrary cost matrix yields a doubly stochastic v, preprocessing of the cost matrix is essential.

Using a Hungarian preprocessor on the cost matrix ensures that the initial M has element values equal to one on a permutation, and values between zero and one on the other elements. This guarantees that at least the selected permutation survives in the normalizing process, and a doubly stochastic matrix v will always result. Though this sounds appealing, our tests reveal that this does not substantially decrease the number of Sinkhorn iterations as compared to the plain Sinkhorn approach, as seen in Fig. 2.

To avoid the extreme increase of the number of normalization iterations in the low temperature region one can apply balancing of the cost matrix c, or use the coupled normalization approach, both described in section VI B 2. Applying either one (or both) of the methods decreases the number of normalization iterations. This is evident in Fig. 2, especially in the low temperature region where the saturation approaches one.

Applying the Hungarian and balancing methods to the cost matrix c leads to an increased time used to produce an initial M, which is revealed in Fig. 3. The total time for the algorithm is dominated by the time spent on Hungarian and balancing. This time is nevertheless far exceeded by the time required with the plain Sinkhorn approach in the low temperature region.

The Hungarian method is known to scale in time with problem size as $O(N^3)$ [1], and the balancing routine empirically appears to behave similarly. This is to be compared to the time it takes to calculate the effective cost matrix, which e.g. for TSP is also an $O(N^3)$ procedure, while for generic quadratic assignment it scales as $O(N^4)$.

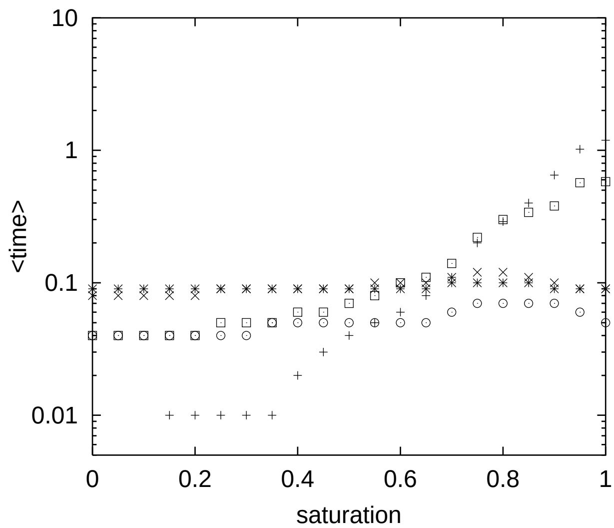

FIG. 3. Time used for normalization, including preprocessing of the cost matrix, versus saturation. The data consists of averages from runs on 100 random linear assignment problems, binned into different values of the saturation. The normalization procedure is continued until all row and column sums are within $1 \pm \Delta_{max}$ with $\Delta_{max} = 0.01$. The plot shows Plain Sinkhorn (+), Hungarian+Sinkhorn (□), Hungarian+CoupledNorm (⊙), Hungarian+Balancing+Sinkhorn (×), and Hungarian+Balancing+CoupledNorm (*). All times are measured in seconds.

B. Stable dynamics in TSP

One of the most studied combinatorial optimization problems is TSP. Deterministic annealing has been applied to it using both PPP and SoftAssign [16,6] and we refer to these articles for a more thorough description of the implementation on TSP. Here we will use TSP as an example where the choice of stabilizing term needed by SoftAssign indeed influences the performance.

The standard assignment-matrix Hamiltonian for TSP is given in equation eq. (40). In addition to this an extra stabilizing term is needed. We have compared the addition of a *generic stabilizer* in the form of a diagonal quadratic term,

$$H_A = -\frac{\alpha}{2} \sum_{ia} s_{ia}^2, \tag{42}$$

as proposed in the literature [6], with a *problem-specific stabilizer* as discussed in section VI B 4($X \to X + \gamma \mathbf{1}$),

$$H_B = \frac{\gamma}{2} \sum_{iab} s_{ia} s_{ib} D_{ab}. \tag{43}$$

Throughout our experiments we have used the values 1.0 for both α and γ. The α value is slightly smaller than the value 1.4 used in [6]. A γ value of 1.0 ensures the stability with respect to local changes of the ordering of visited sites as discussed in section VI B 4, but does not always suffice to produce a proper assignment. When this happens (in about 1% of the tested problems) the algorithm is restarted, initialized with a new v. Typically, one restart is sufficient to find a proper assignment matrix.

We studied random TSP problems where the sites were uniformly generated in the two-dimensional unit square. In Fig. 4 the tour lengths from 500 problems of size 100 are shown.

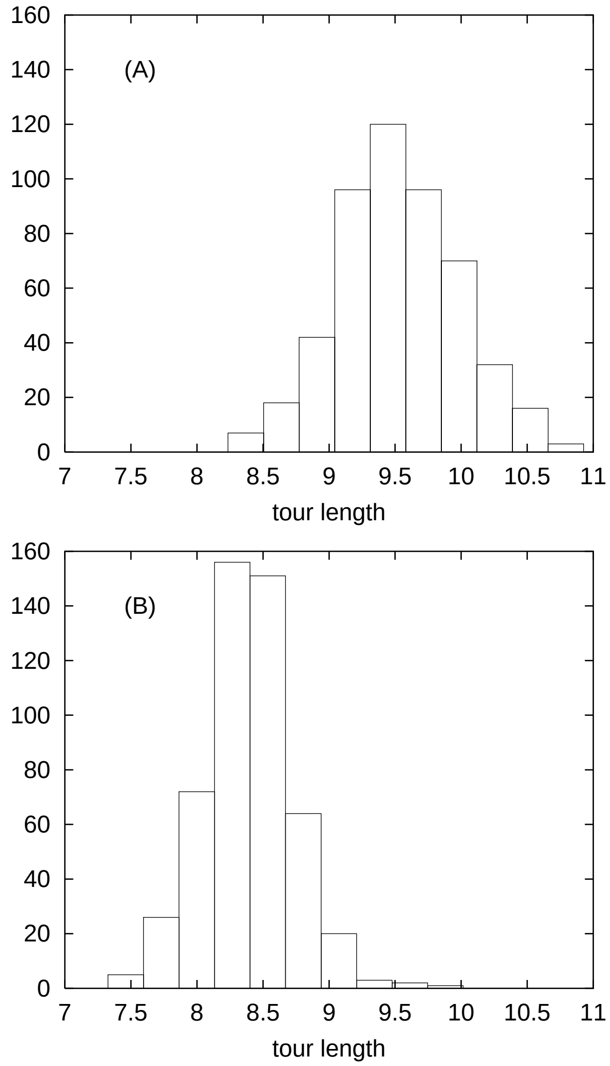

FIG. 4. Tour lengths for 500 random two-dimensional Euclidean TSP problems of size 100, using (**A**) the generic stabilizer (42) with $\alpha = 1.0$, (**B**) the problem-specific stabilizer (43) with $\gamma = 1.0$.

The generic stabilizer (eq. (42)) works by enhancing already large spin elements. Due to this, v saturates faster (towards an assignment matrix) in the course of the annealing. Since this effect is not as pronounced with the problem-specific stabilizer (eq. (43)), equal annealing parameters will not lead to equal time used by the algorithms. Instead, parameters are chosen such that the respective performances are not too far from optimal and the times used are comparable. We have used a slower annealing, $T \to T/1.01$, for the generic stabilizer, and also allowed it to use up to 5 sweeps per temperature if the maximal change in a spin components is larger than 0.01. An even slower annealing would not lead to a considerably better performance, in spite of the increase in time. With the problem-specific stabilizer, we have used the annealing rate $T \to T/1.05$, with one sweep per temperature.

With the generic stabilizer, the average tour length was 9.53 and the average time used 1.53 s. With the problem-specific one, corresponding values where 8.39 and 3.74 s (including possible restarts). Thus, performance-wise, the problem-specific stabilizer is superior, while the increase in time used can be attributed to the slower saturation – this might be avoided by adding a small generic stabilizer term.

VIII. CONCLUSIONS

We have investigated the possibilities of defining a deterministic annealing approach to nonlinear assignment problems, in anology to existing algorithms for Ising and Potts systems.

We have analyzed a proper variational approach, where the problem cost function is approximated by a variational cost, linear in the assignment matrices. For a single assignment problem this allows for an iterative scheme to minimize the variational free energy at a given temperature. Combined with annealing, this can be used as a deterministic annealing algorithm.

As an aside, the generalization to multiple assignment problems is straightforward. Assuming additive linear contributions to the variational cost from the different assignments leads to a mean-field-like approximation with a factorized Boltzmann distribution, and the variational parameters for the individual assignments can be updated in a serial manner.

A major problem with the proper variational approach, however, is that it requires the calculation of permanents, which needs exponential time (in problem size). This implies that considering this as a general heuristic for large nonlinear assignment problems is not feasible.

Abandoning the quest for a proper variational method for nonlinear assignment, we have also studied Potts-based methods as a more promising alternative, although *per se* not tailored for assignment problems. The currently most appealing method of this type is the SoftAssign algorithm, and we have proposed some improvements to it.

The Sinkhorn normalization procedure used in SoftAssign runs into convergence problems at low temperatures. We present arguments why this is unavoidable, and propose proper adjustments of the effective cost matrix to reduce the effect. The application of a Hungarian preprocessing to the effective cost matrix guarantees that the Sinkhorn procedure always produces a doubly stochastic matrix. An additional balancing of the cost matrix decreases the number of iteration needed by the normalization. In addition we devise an alternative normalization procedure which is easily implemented when a Hungarian preprocessing is used. It is superior to the Sinkhorn procedure at low temperatures. We have experimentally confirmed these statements by implementation of SoftAssign on random instances of linear assignment. With other problem ensembles, however, we have experienced varying effects of the improvements, in some cases they appear essential, while in others they are more or less superfluous.

Another problem with the SoftAssign approach is the lack of guarantee for stability in the low temperature region: Solutions may not be stable. This problem can be resolved by adding to the Hamiltonian a stabilizer – a redundant term that affects the dynamics without altering the on-shell cost. We have used arguments from group theory to determine the possible types of redundant additions that are at most quadratic in the spin components. As an example we discuss how such redundant terms can be used for the travelling salesman problem, and propose a TSP-specific stabilizing term different from the generic one normally used with SoftAssign. In numerical experiments we show that this enhances the performance.

IX. ACKNOWLEDGEMENTS

This work was in part supported by the Swedish Foundation for Strategic Research.

---

[1] C. H. Papadimitriou, *Combinatorial Optimization* (Dover Publications, Inc., Mineola, New York, 1998).

[2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).

[3] S. Geman and D. Geman, IEEE Transactions on Pattern Analysis and Machine Intelligence **6**, 721 (1984).

[4] J. J. Hopfield and D. W. Tank, Biological Cybernetics **52**, 141 (1985).

[5] C. Peterson and B. Söderberg, Neural optimization, in *The Handbook of Brain Research and Neural Networks, (2nd edition)*, edited by M. A. Arbib, pp. 617–622, Bradford Books/The MIT Press, Cambridge, Massachusetts, 1998.

[6] S. Gold and A. Rangarajan, Journal of Artificial Neural Networks **2**, 381 (1996).

[7] H. Jönsson and B. Söderberg, Neural Computation (2001), (to appear).

[8] M. Lagerholm, C. Peterson, and B. Söderberg, European Journal of Operations Research **120**, 81 (2000).

[9] C. Peterson and B. Söderberg, International Journal of Neural Systems **1**, 3 (1989).

[10] A. Rangarajan, S. Gold, and E. Mjolsness, Neural Computation **8**, 1041 (1996).

[11] H. W. Kuhn, Naval Research Logistics Quarterly **2**, 83 (1955).

[12] H. Minc, *Permanents* (Addison-Wesley Publishing Company, Reading, Massachusetts, 1978).

[13] H. J. Ryser, *Combinatorial Mathematics* (Mathematical Association of America, Washington DC, 1963).

[14] R. Sinkhorn, Annals of Mathematical Statistics **35**, 876 (1964).
[15] A. Rangarajan, A. Yuille, S. Gold, and E. Mjolsness, A convergence proof for the softassign quadratic assignment algorithm, in *Advances in Neural Information Processing Systems*, edited by M. C. Mozer, M. I. Jordan, and T. Petsche, volume 9, pp. 620–626, The MIT Press, 1997.
[16] C. Peterson, Neural Computation **2**, 261 (1990).

APPENDIX A: BASIC GROUP THEORY FOR P_N

Here we will briefly review elements of the basic group theory for the permutation group of N elements, $G \equiv P_N$.

1. Representations and irreps

G has a finite number of inequivalent irreducible representations, or irreps; the squares of their dimensions sum up to the size V of the group, $V \equiv N!$. If r labels an irrep, let d_r be its dimension, and let the associated $d_r \times d_r$ matrices be denoted $u^r(g)$. We have $\sum_r d_r^2 = V$.

A D-dimensional reducible representation $\{U(g)\}$ can be decomposed into the direct sum of (not necessarily distinct) irreps $\{r_\mu\}$ as

$$U_{ij}(g) = \sum_\mu \sum_{k,l} P^\mu_{ik} P^\mu_{jl} u^{r_\mu}_{kl}(g), \tag{A1}$$

or, in matrix form, $U(g) = \sum_\mu P^\mu u^{r_\mu}(g) P^{\mu\top}$, where P^μ is a (g-independent) $D \times d_{r_\mu}$ matrix that projects out the part of a vector that belongs to the associated irrep $r = r_\mu$. It can be seen as a submatrix of an orthogonal $D \times D$ matrix V, used to similarity transform U to an explicitly blocked form U_B, $U(g) = V U_B(g) V^\top$.

The ortogonality of V implies the following properties for the matrices P^μ:

$$\sum_\mu P^\mu P^{\mu\top} = \mathbf{1}_D, \tag{A2}$$

$$P^{\mu\top} P^\nu = \delta_{\mu\nu} \mathbf{1}_{d_{r_\mu}}, \tag{A3}$$

where $\mathbf{1}_d$ denotes the $d \times d$ identity matrix. Inverting the similarity transform, eq. (A1) is equivalent to

$$\sum_{ij} P^\mu_{ik} P^\nu_{jl} U_{ij}(g) = \delta_{\mu\nu} u^{r_\mu}_{kl}(g), \tag{A4}$$

or, in matrix form, $P^{\mu\top} U(g) P^\nu = \delta_{\mu\nu} u^{r_\mu}(g)$, expressing the similarity transform of U to blocked form, with μ, ν labeling respectively the row and column block.

The assignment matrices s define a particular N-dimensional representation (the defining representation) of P_N. It is *reducible* if $N > 1$, being the direct sum $\mathbf{e} + \mathbf{f}$ of two irreps, where $\mathbf{e}$ is the trivial one-dimensional irrep with $u^{\mathbf{e}} \equiv 1$, while $\mathbf{f}$ is a non-trivial $(N-1)$-dimensional irrep, the *fundamental* representation. For the $\mathbf{e}$ part, e.g., the corresponding $N \times 1$-dimensional projection matrix is given by $P^{\mathbf{e}}_{ik} = 1/\sqrt{N}$.

2. Irrep Expansion

Due to the identity $\sum_r d_r^2 = V$, there are as many distinct matrix elements in the inequivalent irreps as there are elements in the group. As is well known, these elements form a complete orthogonal basis in group space, as expressed by

$$\sum_{g \in G} u^r_{ij}(g) u^s_{kl}(g) = \frac{V}{d_r} \delta_{r,s} \delta_{i,k} \delta_{j,l} (\text{orthogonality}), \tag{A5}$$

$$\sum_r \sum_{i,j=1}^{d_r} d_r u^r_{ij}(g) u^r_{ij}(h) = V \delta_{g,h} (\text{completeness}). \tag{A6}$$

Thus, any function F over the group can be expressed in a unique way as a linear combination of the irrep elements,

$$F(g) = \sum_r \sum_{ij} C^r_{ij} u^r_{ij}(g), \tag{A7}$$

where C are coefficients, and $u^r(g)$ is the orthogonal matrix representing the group element g in the irrep r. Due to the completeness, eq. (A7) can be inverted to yield the coefficients uniquely as

$$C^r_{ij} = \frac{d_r}{N!} \sum_g u^r_{ij}(g) F(g). \tag{A8}$$

a. Linear expressions in s

Eq. (A4) can be interpreted as follows: Independently of the group element g, certain linear combinations of the elements of the matrix $U(g)$ representing g in a reducible representation R, will be identical to the elements of the orthogonal matrix $u^r(g)$ corresponding to an irrep r that appears in R; certain other linear combinations (corresponding to $\mu \neq \nu$) will vanish. Together, these span a complete basis in the space of all possible linear combinations. This can be used to identify the redundant linear or quadratic expressions in the assignment matrix s.

A *linear* function of the assignment matrix s can have non-vanishing coefficients only for $r = \mathbf{e}$ and $r = \mathbf{f}$ in its expansion (A7).[**] Separating the $\mathbf{e}$ and $\mathbf{f}$ parts of s yields

[**] Which shows that the most general cost function is not linear in s.

$$s_{ij} = 1/N + \sum_{kl} P^{\mathbf{f}}_{ik} u^{\mathbf{f}}_{kl} P^{\mathbf{f}}_{jl}. \tag{A9}$$

The different versions of eq. (A4) then yield a set of identities,

$$\sum_{ij} s_{ij}(g) = N, \tag{A10}$$

$$\sum_{ij} s_{ij}(g) P^{\mathbf{f}}_{jl} = 0, \tag{A11}$$

$$\sum_{ij} P^{\mathbf{f}}_{ik} s_{ij}(g) = 0, \tag{A12}$$

$$\sum_{ij} P^{\mathbf{f}}_{ik} s_{ij}(g) P^{\mathbf{f}}_{jl} = u^{\mathbf{f}}_{kl}(g). \tag{A13}$$

The first three of these express in a slightly disguised form the constraints of unit row and column sums in s.

b. Quadratic expressions in s.

A quadratic expression in s means a linear combination of products $U_{ik,jl} \equiv s_{ij}s_{kl}$, defining the direct product representation $(\mathbf{e}+\mathbf{f}) \times (\mathbf{e}+\mathbf{f})$, considering the pair of row indices $\{i,k\}$ as a composite row index, and the pair of column indices $\{j,l\}$ likewise as a composite column index.

It is reducible, and the corresponding version of eq. (A4) reads

$$\sum_{ik,jl} Q^{\mu}_{ik,m} s_{ij}(g) s_{kl}(g) Q^{\nu}_{jl,n} = \delta_{\mu\nu} u^{r_\mu}_{mn}(g). \tag{A14}$$

The non-trivial part comes from $\mathbf{f} \times \mathbf{f}$, which reduces to $\mathbf{e}+\mathbf{f}+\mathbf{s}+\mathbf{a}$ (for $N > 2$), where $\mathbf{s}$ and $\mathbf{a}$ are two new irreps, of dimensionalities $d_{\mathbf{s}} = N(N-3)/2$ and $d_{\mathbf{a}} = (N-1)(N-2)/2$.

Due to the obvious symmetry of U, $U_{ik,jl} \equiv U_{ki,lj}$, it is natural to divide the resulting irreps in two sets, according to the symmetry with respect to swapping the index pair ik in $Q^{\mu}_{ik,m}$. Thus, the *symmetric* part of $\mathbf{f} \times \mathbf{f}$ contains $\mathbf{e}+\mathbf{f}+\mathbf{s}$, while the *antisymmetric* part contains $\mathbf{a}$ alone.

With opposite symmetry type in the row and column parts, the LHS of eq. (A14) vanishes identically. Thus, the interesting parts require μ, ν to correspond to the same type of symmetry. The antisymmetric part contains only the nontrivial part $\mu = \nu = \mathbf{a}$, yielding $u^{\mathbf{a}}$. In the symmetric part, the $\mu = \nu = \mathbf{s}$ part similarly yields $u^{\mathbf{s}}$, while the remaining combinations yield products of elements of s for which the RHS will either vanish ($\mu \neq \nu$), or yield a constant ($\mu = \nu = \mathbf{e}$), or a linear combination of the elements of s ($\mu = \nu = \mathbf{f}$); the LHS then defines candidates for redundant quadratic additions to a cost function.