AVAILABLE AT
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Neural Networks 16 (2003) 1421–1428

Neural Networks

www.elsevier.com/locate/neunet

Comparison of simulated annealing and mean field annealing as applied to the generation of block designs

Pau Bofill[a], Roger Guimerà[b], Carme Torras[c,*]

[a]Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona, Spain
[b]Departament d'Enginyeria Química, Universitat Rovira i Virgili, Autovia de Salou s/n, 43006 Tarragona, Spain
[c]Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Parc Tecnològic de Barcelona-Edifici U, c/Llorens i Artigas 4-6, 08028 Barcelona, Spain

Received 4 July 2001; accepted 17 July 2003

Abstract

This paper describes an experimental comparison between a discrete stochastic optimization procedure (Simulated Annealing, SA) and a continuous deterministic one (Mean Field Annealing), as applied to the generation of Balanced Incomplete Block Designs (BIBDs). A neural cost function for BIBD generation is proposed with connections of arity four, and its continuous counterpart is derived, as required by the mean field formulation. Both strategies are optimized with regard to the critical temperature, and the expected cost to the first solution is used as a performance measure for the comparison. The results show that SA performs slightly better, but the most important observation is that the pattern of difficulty across the 25 problem instances tried is very similar for both strategies, implying that the main factor to success is the energy landscape, rather than the exploration procedure used.

Keywords: Block designs; Neural cost functions; Higher-order connections; Simulated annealing; Mean field annealing; Experimental comparison

1. Introduction

There is nowadays a growing interest in computational models inspired in physical and natural phenomena (quantum computing, DNA computing, evolutionary algorithms, chaotic networks). This, together with the practical successes attained by artificial neural networks, has raised some intriguing theoretical questions on the links between discrete and analog computation. Is there a relationship between the complexity of algorithms in the continuous and discrete settings for solving the same problem? How do approximate solutions affect the complexity?

Computational complexity theory, as developed within Computer Science, has focused on discrete computation, whereas analog computation has mainly been studied using the tools of Theoretical Physics. However, to try to answer the above questions, theories to study both types of computation under a unifying perspective are needed, and these are only beginning to be developed. Meanwhile, several experimental studies have been undertaken to explore the links between discrete and continuous solutions to the same problem. This paper presents one such study.

Block designs are combinatorial objects satisfying a set of integer constraints (Colbourn & Dinitz, 1996; Hall, 1986; Street & Street, 1987). They were introduced in the thirties by statisticians working on experiment planning, and are nowadays being used in many other fields, such as coding theory, network reliability, and cryptography. The most widely used designs are the Balanced Incomplete Block Designs (BIBDs). Their generation is a very challenging NP problem (Corneil & Mathon, 1978) with a wide variety of problem instances, ranging from very easy to very hard ones; some relatively small designs even remaining unsolved (McKay & Radziszowski, 1996).

Its wide variability in size and difficulty makes this problem a very appropriate benchmark to carry out the aforementioned comparative experimental study. Moreover, we have good knowledge of the relative computational difficulties of its different instances, after having explored them thoroughly using systematic search procedures (Meseguer & Torras, 1999, 2001).

* Corresponding author.
E-mail addresses: ctorras@iri.upc.es (C. Torras); pau@ac.upc.es (P. Bofill); rguimera@etseq.urv.es (R. Guimerà).

0893-6080/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2003.07.003

In this paper, we formulate the generation of BIBDs as a combinatorial optimization problem and tackle it using optimizing neural networks. Several such network models have been proposed in the literature leading to good practical results: Hopfield networks (Hopfield & Tank, 1985), Boltzmann machines (Aarts & Korst, 1988), Cauchy machines (Takefuji & Szu, 1989), Gaussian machines (Akiyama, Yamashita, Kajiura, & Aiso, 1989), Mean Field Annealing (MFA) (Peterson & Anderson, 1987; Peterson & Södeberg, 1989) and, more recently, chaotic neural networks (Chen & Aihara, 1995; Kwok & Smith, 1999). Among them, we have chosen two classical models for the comparison, namely Boltzmann machines (which implement Simulated Annealing, SA) and MFA. The former works on a discrete state space, while the latter evolves in a continuous one. In some sense, the latter can be seen as the deterministic version of the former. Both are based on statistical mechanics models and, therefore, they have a strong theoretical basis (Aarts & Korst, 1988; Hertz, Krogh, & Palmer, 1993). In particular, due to the characteristics of the problem tackled, we use networks with higher-arity connections (Sejnowski, 1986), whose dynamics is analysed in Goles and Matamala (1994).

Let us now turn to the performance measures to be used in the comparison. Due to its constraint satisfaction nature, BIBD generation is different from most combinatorial optimization problems where the goal is to optimize the 'quality' of the solution (minimize the energy) with as little computational cost as possible. In our case, only global minima are of interest, that is, solutions corresponding to BIBDs. In such a setting, the *expected number of runs to the first solution* is a measure of the *efficacy* of an optimization strategy (or, reciprocally, of the *difficulty* of a problem). Yet, it does not take into account the resources invested by different algorithms. Thus, for an objective comparison, the *expected cost to the first solution* must be used, as a measure of the *efficiency* of the search. In this work, since the two relaxation strategies under study (SA and MFA) are based on local search, the computational complexity of an iteration (one update of each unit in the network) is the same for both of them. Thus, in order to avoid implementation issues, computational costs are compared directly in terms of the number of invested *iterations*.

2. The problem of BIBD generation

A *Balanced Incomplete Block Design* (v, b, r, k, λ)-BIBD can be defined in terms of its incidence matrix as follows. Let $A \equiv [x_{ij}]$ be a given configuration in the space $\mathscr{A} - v \times b$ of binary configurations with v rows and b columns. Let the state variables $x_{ij} \in \{0, 1\}$ represent the incidence of treatment i in block j of A, and let $r_i = \sum_{j=1}^{b} x_{ij}$ be the number of ones in row i (the replicate number for treatment i), $k_j = \sum_{i=1}^{v} x_{ij}$ the number of ones in column j (the size of block j), and $\lambda_{ik} = \sum_{j=1}^{b} x_{ij}x_{kj}$ the correlation or dot product between rows i and k (the number of times that treatments i and k occur together in a block).

Definition 1. For fixed r, k and λ, with $k < v$ and $\lambda > 0$, we say that A is the incidence matrix of a BIBD with parameters (v, b, r, k, λ) if and only if:

(i) *Uniform columns*: $k_j = k$, $j = 1, \ldots, b$.
(ii) *Balance*: $\lambda_{ik} = \lambda$, $i = 1, \ldots, v-1$, $k = i+1, \ldots, v$.

It is not difficult to show (Street & Street, 1987) that when (i) and (ii) apply it also follows that

(iii) *Uniform rows*: $r_i = r$, $i = 1, \ldots, v$.

In order to be consistent, a set of BIBD parameters has to fulfill the following *admissibility* conditions:

$$kb = rv$$

$$\lambda(v-1) = r(k-1).$$

The first equation adds up to the total number u of ones in the design, while the second equation counts the number of ones co-occurring with those of a given row.

The admissibility of its parameters is a necessary but not sufficient condition for the existence of a block design. The situation is summarized in Mathon and Rosa (1996), that lists all non-trivial admissible parameter sets with $r \leq 41$, together with the currently known bounds on the number of *non-isomorphic* solutions. For some particular parameter sets, it has been established that a design does not exist, and other cases remain unsettled. Some (infinite) families of block designs (designs whose parameters satisfy particular properties) can be constructed analytically, by direct or recursive methods (Hall, 1986, Chapter 15), and the state of the art in computational methods for design generation is described in Gibbons (1996). The smallest unsettled case is (22,33,264) (McKay & Radziszowski, 1996), with $vb = 726$ entries, which proves that exhaustive search is still intractable for designs of this relatively small size. In the general case, as with other combinatorial configurations, the algorithmic generation of block designs is an NP problem (Corneil & Mathon, 1978).

3. A fourth-order network for the generation of BIBDs

The generation of block designs is a constraint satisfaction problem. In order to use optimizing neural networks, we must first reformulate it as a combinatorial optimization problem and then map it onto a standard neural network architecture.

Let $\mathscr{A} - v \times b$ be the set of all $A \equiv [x_{ij}]$ binary configurations with v rows and b columns. We say that $F : \mathscr{A} \rightarrow \mathscr{R}$ is a *cost function* for the generation of block designs if there exists a *lower bound* F^* such that $F(A^*) = F^*$ if and

only if there exists a (v, b, r, k, λ)-BIBD with incidence matrix A^*.

Let

$$N_q = \sum_{i=1}^{v-1} \sum_{k=i+1}^{v} \sum_{j=1}^{b-1} \sum_{l=j+1}^{b} x_{ij}x_{kj}x_{il}x_{kl}$$

be the number of *quadruples* of ones in a given configuration A (combinations of four ones arranged square-wise), and let

$$N_u = \sum_{i=1}^{v} \sum_{j=1}^{b} x_{ij}$$

be the number of ones in A. The *local increment* of these measures with respect to component (α, β) is

$$\Delta^{\alpha\beta}N_q = N_q|_{x_{\alpha\beta}=1} - N_q|_{x_{\alpha\beta}=0} = \sum_{i\neq\alpha}^{v} \sum_{j\neq\beta}^{b} x_{\alpha j}x_{ij}x_{i\beta}$$

$$\Delta^{\alpha\beta}N_u = N_u|_{x_{\alpha\beta}=1} - N_u|_{x_{\alpha\beta}=0} = 1.$$

We finally define the cost function

$$F_{uq} = -\nu N_u + N_q, \tag{1}$$

with local increment

$$\Delta^{\alpha\beta}F_{uq} = -\nu\Delta^{\alpha\beta}N_u + \Delta^{\alpha\beta}N_q.$$

Notice that, from the definition of local increment, the actual increment resulting from *updating* unit (α, β) (i.e. $x_{\alpha\beta} \leftarrow \bar{x}_{\alpha\beta}$) is $(1 - 2x_{\alpha\beta})\Delta^{\alpha\beta}F_{uq}$.

Theorem 1. *Given a set (v, b, r, k, λ) of admissible parameters, the function F_{uq} in Eq. (1) is a cost function for the generation of BIBDs iff*

$$(k-1)(\lambda-1) < \nu < k\lambda.$$

Its global minimum is

$$F^*_{uq} = -\nu u + \binom{v}{2}\binom{\lambda}{2}.$$

As described in detail in Bofill (1997), over all configurations with a given number o of ones, the number $N_q|o$ of quadruples is minimum when columns are maximally uniform and pairs of rows are maximally balanced. The marginal increment $\Delta^*N_q|o$ of this lower bound increases with o, whereas the marginal increment for N_u is 1. Thus, by setting ν between $\Delta^*N_q|u-1$ and $\Delta^*N_q|u$, we guarantee that any configuration minimizing F_{uq} will have exactly the desired number u of ones. The optimal value is obtained by counting the number of ones and quadruples that would be present in a configuration satisfying conditions (i)–(iii) in Definition 1. For the sake of symmetry, in the experiments described later, ν was set to the value in the middle of the allowed range ($\nu = k\lambda - k/2 - \lambda/2 + 1/2$). Other cost functions for BIBD generation that make use of the remaining properties of a block design (ones per row, ones per column and quadruples of zeros) are described in Bofill (1997) and discussed in Section 7.

The above cost function is isomorphic to an *optimizing neural network* by arranging vb neural units in v rows and b columns, and assigning each state variable x_{ij} to the unit with coordinates (i, j). Then, each square-wise combination of four units corresponds to a potential quadruple (a fourth-order connection with connection weight 1), and all individual units have their threshold with weight $-\nu$. Cost function F_{uq} corresponds thus to the energy E of the network and the local increments $\Delta^{\alpha\beta}F_{uq}$ correspond to local fields. In the following, the terms energy and function cost will be used indistinctly.

4. Optimization with simulated annealing

The basic relaxation strategy for optimization networks is Down-Hill search. The decision rule is to accept all energy-decreasing transitions, until an optimum is found or the algorithm converges to a minimum:

$$x_{\alpha\beta} \leftarrow \bar{x}_{\alpha\beta} \qquad \text{iff } (1 - 2x_{\alpha\beta})\Delta^{\alpha\beta}F < 0.$$

The goal of SA is to avoid undesired local minima by means of *thermal noise*. If the Metropolis decision rule is used (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953), the *probability* of accepting a transition at *computational temperature* T is given by

$$P\{x_{\alpha\beta} \leftarrow \bar{x}_{\alpha\beta}\} = \begin{cases} 1, & (1-2x_{\alpha\beta})\Delta^{\alpha\beta}F < 0 \\ e^{-((1-2x_{\alpha\beta})\Delta^{\alpha\beta}F/T)}, & \text{otherwise} \end{cases}.$$

The efficacy of SA depends on a good temperature *schedule*. The goal is to spend most of the relaxation time around the *critical temperature* T_{cri}, where global minima start to be noticeable (i.e. when escaping from deeper minima starts to be significantly harder). For $T \gg T_{cri}$, the system evolves randomly, whereas for $T \ll T_{cri}$ the system 'freezes' in a local minimum. The usual procedure is to start at a sufficiently high initial temperature T_0 and decrease it slowly until some final temperature T_f.

In this work, in order to avoid big perturbations to the thermal equilibrium, rather than making a large temperature update after each iteration, we made smaller temperature updates after each *unit* update, thus leading to a smooth cooling schedule. The actual decrement law that was used (Gutzmann, 1987) is

$$T_k = \sqrt[vb]{\tau}T_{k-1},$$

with k the number of updated *units* and τ the decay constant corresponding to an iteration. The temperature range was set to $T_0/T_f = 2$ and the maximum number of iterations per descent was set to $N_t = 100$, thus yielding $\tau = (T_f/T_0)^{1/N_t}$. If after N_t iterations, no solution had been found, the search was stopped and the system was driven with Down-Hill

search to the nearest local minimum. Yet, in practice, most of the solutions were found well before N_t iterations.

The remaining parameter is the *central* operational temperature $T_c = \sqrt{T_0 T_f}$, which was set experimentally as follows. A training set was defined with seven problems randomly selected among the 25 smallest BIBD settings (see Table 1 in Section 6), and the optimal temperature T_c^{opt} was obtained for each problem by experimental optimization (finding the T_c that minimized the expected cost to the first solution). An experimental law could then be found that relates the optimal temperatures to the parameters of the BIBD, and a 'standard' central temperature setting was thus defined, $T_c^{std} = 0.110 \times (k + \lambda - 1)$, allowing for an a priori estimate of the operational temperature. Results were then generalized over the remaining $(25 - 7)$ problems (the test set), and the optimality of this standardized setting was verified, with quite consistent results.

5. Optimization with mean field annealing

The deterministic counterpart of SA is MFA. According to statistical mechanics, the average value of a unit x_{ij}, that we will denote m_{ij}, is given by

$$m_{ij} = \langle x_{ij} \rangle = \frac{\sum_k x_{ij}^k \exp(-F^k/T)}{Z},$$

where the sum extends over all the possible states k of the unit, F is the energy of the configuration and depends of the state of all the units, Z is the partition function, and T is the computational temperature, as before. The mean field approximation consists in considering that each unit is interacting with an average field $\langle h_{ij} \rangle$ generated by the others and given by

$$\langle h_{ij} \rangle = \frac{\partial F}{\partial x_{ij}}.$$

Thus, the value m_{ij} of a continuous unit is found through iteration by solving the equation

$$m_{ij} = \frac{1}{2}\left[1 + \tanh\left(-\frac{1}{2T}\langle h_{ij} \rangle\right)\right]. \quad (2)$$

The continuous counterpart of the cost function (1) is

$$F_{uq} = -\nu \sum_{i=1}^{v} \sum_{j=1}^{b} m_{ij} + \sum_{i=1}^{v-1} \sum_{k=i+1}^{v} \sum_{j=1}^{b-1} \sum_{l=j+1}^{b} m_{ij} m_{kj} m_{il} m_{kl}$$

and the corresponding mean field felt by a unit in row α and column β is given by

$$\langle h_{\alpha\beta} \rangle_{uq} = -\nu_u + \sum_{i \neq \alpha}^{v} \sum_{j \neq \beta}^{b} m_{\alpha j} m_{ij} m_{i\beta}.$$

Iterating, as we did, at a fixed operational temperature T, the convergence of Eq. (2) shows two different behaviors or *phases*. For T greater than the critical temperature of the system T_{cri}, the final state corresponds to the fixed point where all the units have the same value, m^*, given by the condition

$$m^* = \frac{1}{2}\left[1 + \tanh\left(-\frac{1}{2T}\langle h(m_{ij} = m^*) \rangle\right)\right].$$

Thus, no feasible solutions for the optimization problem are found in this high temperature phase.

On the contrary, when $T < T_{cri}$, the units relax to values close to 1 or 0 (Fig. 1). By actually *freezing* the units to either 0 or 1, the solutions obtained correspond to low energy configurations of the optimization problem.

To characterize the transition between one phase and the other, a *saturation* σ can be defined in the following way (Peterson & Södeberg, 1989):

$$\sigma = \frac{1}{u} \sum_{i,j} m_{ij}^2.$$

For the low temperature phase, σ is close to 1, while for $T > T_{cri}$, it is significantly smaller. Measuring the saturation σ and plotting it against increasing values of the operational temperature T, we observe that, at the critical temperature T_{cri}, either a discontinuity in the plot or a discontinuity in its first derivative is observed, depending on the problem considered (Fig. 2). Since in the high temperature phase, the network does not find feasible solutions, T_{cri} represents an upper bound for the actual operational temperature, and finding the discontinuities in the saturation curve is a useful procedure for estimating T_{cri}. It is worth noting that there is not an equivalent procedure in the discrete case (SA).

Typically, as the operating temperature is closer from below to T_{cri}, it is more likely to find a solution since the probability of the system being trapped in a local minimum is lower. However, for the same reason, descents are usually longer in this case. Thus, these two effects compete and in general there is an optimal temperature T^{opt} for which the expected cost to the first solution is minimum, although

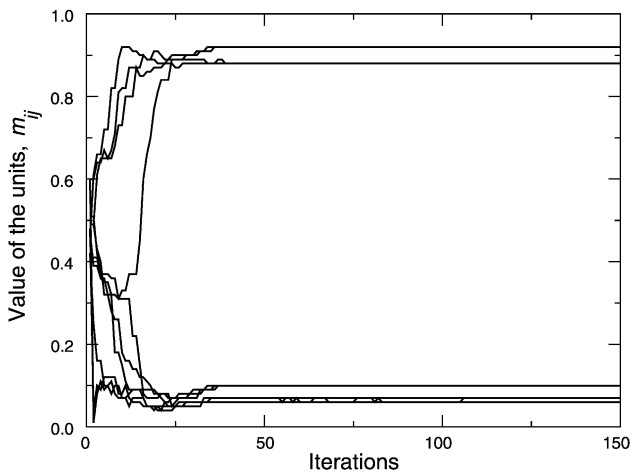

Fig. 1. Evolution of the value m_{ij} of the units for $T = 2.1 < T_{cri}$ on problem (10,18,9,5,4) (problem *d12* in Table 1, Section 6).

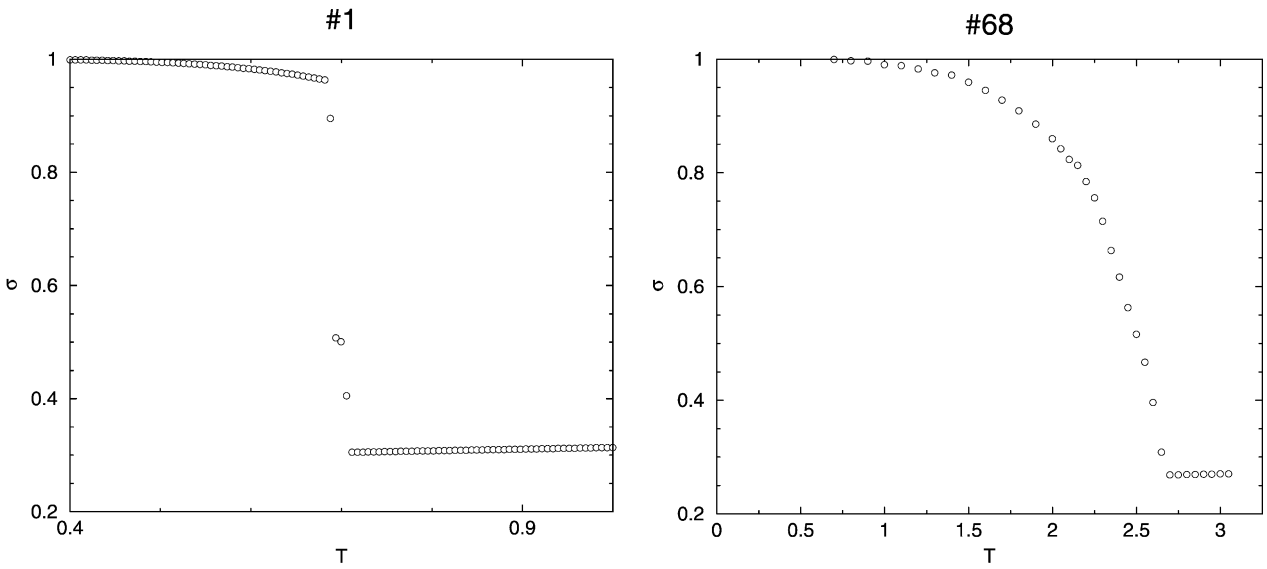

Fig. 2. Two different saturation curves as a function of the operational temperature, for problems (7,7,3,3,1) and (9,60,20,3,5), respectively.

sometimes this minimum is wide instead of being localized at a given T^{opt} (Fig. 3). For the experiments reported in Section 6, where MFA and SA are compared, we used this optimal temperature as the actual operational temperature for each problem.

The second issue that must be taken into account is when to stop a descent. At the end of each iteration, we *freeze* the system by setting the units either to 0 or 1. If the configuration corresponds to the incidence matrix of a BIBD, then we stop the process (a successful descent), otherwise we continue. Thus, we have to provide a mechanism to decide when to stop an unsuccessful descent. As in the case of SA, it would be possible to set a maximum number of iterations. Nevertheless, it is possible to use again the continuous nature of the units to establish a more adequate stopping criterion. When a unit is trapped in a value close to 0 or 1, its value is very unlikely to change. Based on this, we can stop our descent and consider it has failed when all the units are near 0 or 1. Although this may frustrate some successful descents, it also drastically reduces the cost of failed descents, resulting in a general reduction of the expected cost to the first solution. In our experiments, we stop the descents when all the units have values outside the region (0.25,0.75). In addition, it is necessary to set a maximum number of iterations for cases in which a unit is trapped in a metastable value inside this region, but since this does not happen very often, it has little influence on the average expected cost. In our experiments, the maximum number of iterations is set to three times the average cost of successful descents.

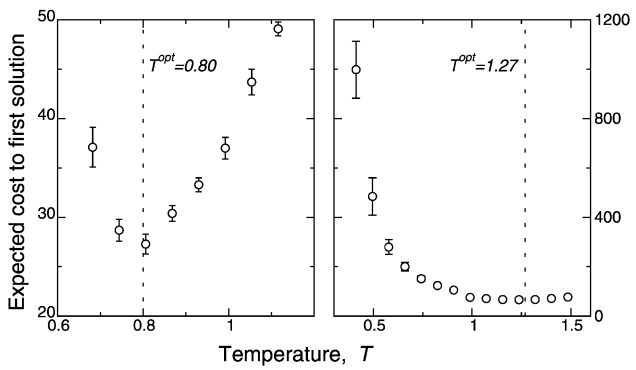

Fig. 3. Cost to the first solution as a function of T for problems (7,14,6,3,2) and (11,11,5,5,2) (problems *d3* and *d7* in Section 6).

6. Experimental comparison

Table 1 shows the experimental results, comparing the expected cost to the first solution for the two strategies. The most important observation is that the pattern of difficulty is similar. Harder problems were hard for both MFA and SA, while easier problems were easy in both cases. The main factor to success is thus the energy landscape of the problem itself, rather than the strategy being used to explore it.

Moreover, the fact that the observed pattern of difficulty is in agreement with the results previously obtained using systematic search procedures (Meseguer & Torras, 1999, 2001) implicitly validates the cost function proposed, which thus appears to reflect faithfully the characteristics of the problem.

Even though the differences in the results were not large, SA performed slightly better, with 13 best marks out of the 25 problems studied, against five for MFA. In six cases, results were within the experimental error and problem *d22* remained unsolved.

Fig. 4 shows a comparison between the optimal temperatures for each strategy. Notice that although

Table 1
Expected cost to the first solution (in iterations) for MFA and SA over the 25 smallest problems in Mathon and Rosa (1996)

#	(v,b,r,k,λ)	MFA		SA	
d2	(6,10,5,3,2)	12.731	(0.092)	*7.9	(0.2)
d4	(9,12,4,3,1)	92.386	(4.002)	*80.0	(1.7)
d8	(7,21,9,3,3)	64.223	(2.684)	61.9	(1.4)
d10	(9,18,8,4,3)	1021.551	(67.714)	*572.4	(37.5)
d15	(9,24,8,3,2)	84.703	(5.217)	*41.1	(0.2)
d18	(6,40,20,3,8)	10095.217	(6527.630)	*1906.8	(175.0)
d21	(16,16,6,6,2)	445.919	(26.674)	421.9	(32.6)
d1	(7,7,3,3,1)	*6.752	(0.085)	7.3	(0.1)
d3	(7,14,6,3,2)	23.525	(0.678)	*13.0	(0.1)
d5	(8,14,7,4,3)	169.875	(16.921)	*112.3	(5.6)
d6	(6,20,10,3,4)	*100.405	(6.815)	122.1	(1.6)
d7	(11,11,5,5,2)	57.599	(0.528)	*48.2	(0.6)
d9	(10,15,6,4,2)	705.085	(87.724)	*233.9	(7.1)
d11	(13,13,4,4,1)	*14.842	(0.173)	22.9	(0.4)
d12	(10,18,9,5,4)	22811.167	(2841.495)	29441.2	(5128.6)
d13	(6,30,15,3,6)	981.193	(207.090)	*453.7	(21.7)
d14	(7,28,12,3,4)	598.844	(65.344)	596.0	(13.5)
d16	(8,28,14,4,6)	15072.170	(11627.418)	9381.1	(1507.9)
d17	(15,15,7,7,3)	*1403.124	(80.516)	3894.1	(512.9)
d19	(11,22,10,5,4)	*124329.667	(480.077)	206995.0	
d20	(7,35,15,3,5)	5688.000	(3699.850)	*840.3	(34.9)
d22	(12,22,11,6,5)	–		–	
d23	(7,42,18,3,6)	13486.000	( )	*3550.3	(372.2)
d24	(10,30,12,4,4)	29540.560	(17623.532)	35200.7	(21245.0)
d25	(10,30,9,3,2)	182.888	(14.390)	*79.9	(2.1)

The first seven rows correspond to the training set for SA, and the remaining 18 to the test set. Entries show mean values and their deviations (within parenthesis) after five descents. For each problem, the entries signaled '*' show significantly best marks. Hyphens signal unsolved problems

the values of the temperatures differ, low optimal temperatures in MFA correspond to low optimal temperatures in SA and, conversely, high optimal temperatures in MFA correspond to high optimal temperatures in SA. Again, this shows that optimal temperatures are intimately related to the energy landscape of the problem.

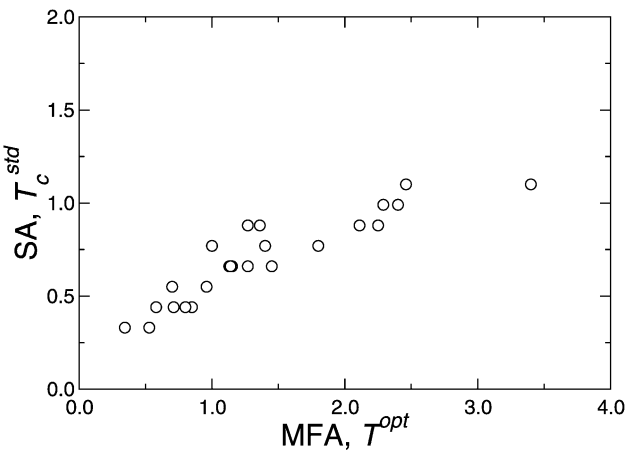

Fig. 4. Optimal temperature for Simulated Annealing against optimal temperature for Mean Field Annealing for the first 25 problems.

7. Modifications to the cost function and their effects on the landscape

Other cost function structures were tried in an attempt to smooth the energy landscape and improve the results. In the first place, redundancy was added by quadratically penalizing configurations with too many ones, rows with too many ones, and columns with two many ones. But the number of local minima actually increased and the results were poorer. Then, a different structure was tried by balancing the quadruples N_q of ones against the quadruples $N_{\bar{q}}$ of zeros. But again the number of local minima increased. In a third attempt, with similar results, a second order cost function was derived by considering the quadratic error between the current number of quadruples and the number of quadruples in an optimal configuration. See Bofill (1997) for details on the three attempts.

Finally, a different approach was tested that consisted of changing the updating rule (and, as a consequence, the definition of the states) rather than the cost function structure. Instead of updating one unit at a time, a quadruple of units were updated simultaneously (four units arranged square-wise) following a $(10/01 \rightarrow 01/10)$ scheme. The transition was accepted whenever the number N_q of quadruples decreased but, by construction, the number of ones per row and column remained always constant. Thus, the state space was restricted to all states with the exact number of ones per row and column. The same scheme can be found in Mathon (1989) and we used it again later in Bofill and Torras (2003). Results with this approach were much better than with the original cost function, but the structure of the system was no longer a neural network.

Our interpretation of these results is the following. There seem to be energy barriers associated with setting the number of ones per row and column. Adding redundancy to these terms—as in the first attempt—simply makes matters worse. Yet, by suppressing all intermediate states and allowing only the states with the right number of ones per row and column—as in the non-neural approach mentioned—most of the barriers disappear. The second attempt is faced with the local minima of both quadruples of ones and quadruples of zeros. And the third attempt is very restrictive in the values of the units, thus generating many local minima.

Thus, without stepping out from the neural networks context in which we have set up our comparison, it turns out that the cost function F_{uq} is the one providing most suitable landscapes (i.e. with the least number of local minima) among the ones tried in our experimentation.

8. Discussion and related work

Since the seminal paper of Kirkpatrick, Gelart, and Vecchi (1983), SA has been applied to a large number of combinatorial optimization problems. In general, SA has

been shown to give good solutions and actually it has been demonstrated that it can find the global optimum if the cooling schedule is slow enough. From a practical point of view, this temperature schedule is unfeasible and, even with faster schedules, the algorithm can be very time consuming.

With the proposal of the MFA algorithm (Peterson & Anderson, 1987, 1988; Peterson & Södeberg, 1989) this shortcoming seemed to be overcome and its application to a number of problems showed that solutions comparable to those obtained with SA could be obtained in much shorter times (Bilbro, Snyder, Granier, & Gault, 1992; Peterson & Anderson, 1987, 1988; Peterson & Södeberg, 1989). Recent works dealing with different problems have arrived to similar conclusions. To cite only a few examples, Herault and Horaud (1993) found MFA to perform 40 times faster than SA in a problem of image recognition, and Aykanat, Bultan, and Haritaoglu (1998) showed that a MFA-based algorithm was able to get good solutions faster than SA.

However, examples in which SA outperforms MFA in terms of quality of the solutions are also abundant in the literature. For example, SA has been shown to yield better results in problems of automated timetabling (Elmohamed, Coddington, & Fox, 1998), in traffic management (Ansari, Arulambalam, & Balasekar, 1996), and in Markov random field modeling (Li, Wang, Chan, & Petrou, 1997).

These examples and many others have suggested that, although SA and MFA are related algorithms, the different nature of the updating process (continuous–deterministic versus discrete–stochastic) may have a significant effect on performance. Clarifying whether this is the case and understanding the origin of the reported performance differences is an important open issue in combinatorial optimization.

From this perspective, the generation of BIBDs provides an excellent benchmark since, within the same family of problems, one finds both easy problems and difficult ones, as well as problems that are more efficiently solved by MFA versus problems for which SA works better. Moreover, in the case of BIBD generation, the 'quality' of the optimization process is a binary variable: either *success* when a solution—global minimum—is found, or *failure* when it is not. Thanks to this, by choosing the expected cost to the first solution as the performance measure, we solve the dilemma of having to consider separately the quality of the solution and the computational cost.

Our work on BIBDs sheds light on two different aspects of the above issue. First, it establishes that in order to achieve *qualitative* better results, it is necessary to change the structure of the energy landscape rather than the optimization algorithm. Second, the detailed results on the benchmark permit studying in which situations one should consider MFA and in which ones one should resort to SA in order to get best *quantitative* results.

9. Conclusions

This paper has presented the formulation of BIBD generation in terms of optimization with a fourth-order neural network, and its continuous counterpart in the framework of mean field theory. A total of 25 problems have been used for the experimental comparison between two optimization strategies, SA and MFA, in terms of the expected cost to the first solution.

Operational temperature and stopping condition were defined differently for each strategy. For SA, a training stage was used over a subset of the problems, and parameters were generalized to the remaining test problems. For MFA, on the other hand, the continuous nature of the units provided a stopping condition and an experimental procedure for determining the critical temperature of the system, based on the saturation versus temperature plot. To use best case results, though, comparisons were done using the optimal temperature for each problem.

In all, SA performed slightly better, but the most significant result was a common *pattern of difficulty*: difficult problems for SA were also difficult for MFA and easy problems for SA were also easy for MFA. There was a similar relationship between the optimal operational temperatures in both cases. This shows that even when the search strategy may have some influence, the main factor determining the difficulty of a problem is the shape of its energy landscape, and deciding which is actually the best strategy depends on the problem considered. This suggests that significant improvements in performance should be oriented to smoothing or simplifying the energy surface, rather than trying to improve the optimization technique itself.

References

Aarts, E. H. L., & Korst, J. H. M. (1988). *Simulated annealing and Boltzmann machines: A stochastic approach to combinatorial optimization and neural computing*. New York: Wiley.

Akiyama, Y., Yamashita, A., Kajiura, M., & Aiso, H. (1989). Combinatorial optimization with Gaussian machines. *Proceedings of the International Joint Conference on Neural Networks*, *1*, 533–540.

Ansari, N., Arulambalam, A., & Balasekar, S. (1996). Traffic management of a satellite communication network using stochastic optimization. *IEEE Transactions on Neural Networks*, *7*(3), 732–744.

Aykanat, C., Bultan, T., & Haritaoglu, I. (1998). A fast neural-network algorithm for VLSI cell placement. *Neural Networks*, *11*(9), 1671–1684.

Bilbro, G. L., Snyder, W. E., Granier, S. J., & Gault, J. W. (1992). Mean field annealing. A formalism for constructing GNC-like algorithms. *IEEE Transactions on Neural Networks*, *3*, 131–138.

Bofill, P (1997). *Optimizing neural networks for the generation of block designs*. PhD Dissertation, Technical report UPC-DAC-1997-76.

Bofill, P., & Torras, C. (2003). MBMUDs: a combinatorial extension of BIBDs showing good optimality behaviour. *Journal of Statistical Planning and Inference*, in press.

Chen, L., & Aihara, K. (1995). Chaotic simulated annealing by a neural network model with transient chaos. *Neural Networks*, *8*(6), 915–930.

Colbourn, C. H., & Dinitz, J. H. (Eds.), (1996). *The CRC handbook of combinatorial designs*. Boca Raton: CRC Press.

Corneil, D. G., & Mathon, R. A. (1978). Algorithmic techniques for the generation and analysis of strongly regular grafs and other combinatorial configurations. *Annals of Discrete Mathematics*, *2*, 1–32.

Elmohamed, M. A. S., Coddington, P., & Fox, G. (1998). A comparison of annealing techniques for academic course scheduling. *Lecture Notes in Computer Science*, *1408*, 92–112.

Gibbons, P. B. (1996). Computational methods in design theory. In: *The CRC handbook of combinatorial designs*, pp. 730–740

Goles, E., & Matamala, M. (1994). Dynamical and complexity results for high order neural networks. *International Journal of Neural Systems*, *5*(3), 241–252.

Gutzmann, K. M. (1987). Combinatorial optimization using a continuous state Boltzmann machine. *Proceedings of the IANN Conference, San Diego*.

Hall, M. (1986). *Combinatorial theory* (2nd ed.). New York: Wiley.

Herault, L., & Horaud, R. (1993). Figure-ground discrimination: a combinatorial optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(9), 899–914.

Hertz, J., Krogh, A., & Palmer, R. G. (1993). *Introduction to the theory of neural computation*. Reading, MA: Addison-Wesley.

Hopfield, J. J., & Tank, D. W. (1985). 'Neural' computation of decisions for optimization problems. *Biological Cybernetics*, *52*, 141–152.

Kirkpatrick, S., Gelart, C. D., Jr, & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680.

Kwok, T., & Smith, K. (1999). A unified framework for chaotic neural network approaches to combinatorial optimization. *IEEE Transactions on Neural Networks*, *10*(4), 978–981.

Li, S. Z., Wang, H., Chan, K. L., & Petrou, M. (1997). Minimization of MRF with relaxation labeling. *Journal of Mathematical Imaging and Vision*, *7*(2), 149–161.

Mathon, R (1989). *Sa2des: A 2-(v, k, λ) design generator, using simulated annealing search (version 1.1d)*, Toronto.

Mathon, R., & Rosa, A. (1996). 2-(v,k,λ) Designs of small order. *CRC Handbook of Combinatorial Design*, pp. 3–41.

McKay, B. D., & Radziszowski, S. P. (1996). Towards deciding the existence of 2-(22,8,4) designs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, *22*, 211–222.

Meseguer, P., & Torras, C. (1999). Solving strategies for highly symmetric CSPs. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99), Stockholm*, 400–405.

Meseguer, P., & Torras, C. (2001). Exploiting symmetries within constraint satisfaction search. *Artificial Intelligence*, *129*(1/2), 133–163. Special issue on 'Heuristic Search'.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations for fast computing machines. *Journal of Chemical Physics*, *21*, 1087–1092.

Peterson, C., & Anderson, J. R. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, *1*(5), 995–1019.

Peterson, C., & Anderson, J. R. (1988). Neural networks and NP-complete optimization problems: a performance study on the graph bisection problem. *Complex Systems*, *2*, 3–22.

Peterson, C., & Södeberg, B. (1989). A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, *1*(1), 3–22.

Sejnowski, T. J. (1986). *Higher-order Boltzmann machines. Proceedings of AIP*, UT, USA: Snowbird.

Street, A. P., & Street, D. J. (1987). *Combinatorics of experimental design*. Oxford: Oxford Science Publications.

Takefuji, Y., & Szu, H. (1989). Design of parallel distributed Cauchy machines. *Proceedings of the International Joint Conference on Neural Networks*, *1*, 529–532.