ELSEVIER

Theoretical Computer Science 215 (1999) 51–68

Theoretical Computer Science

Exploration of NP-hard enumeration problems by simulated annealing – the spectrum values of permanents

Yaghout Nourani*, Bjarne Andresen

Ørsted Laboratory, Niels Bohr Institute, University of Copenhagen, Universitetsparken 5, DK-2100 Copenhagen Ø, Denmark

Received May 1996; revised May 1997
Communicated by M. Nivat

Abstract

The Monte Carlo optimisation technique simulated annealing has been used to sample the possible values of the permanent of fully indecomposable (0,1)-matrices. The basic idea in simulated annealing is that a Markov chain explores the space of feasible solutions, each of which has an associated "energy" (here minus the permanent of the matrix).

The control parameter "temperature" is gradually reduced in order to focus the search on the low-energy region of state space and eventually reaching the minimum. Although the minimum energy per se (largest permanent) is known, the spectrum above the minimum is not, so this importance sampling technique is quite efficient.

Interest in permanents is based on the fact that it is "complete" for the class $\#P$ of enumeration problems, that is as hard as counting the number of accepting computations of any nondeterministic polynomial time Turing machine. The spectrum of possible values for the permanent of a (0,1)-matrix with given topological dimension d is quite interesting but only partially understood. The present work is an experimental study of this spectrum using the handle basis representation of directed graphs and simulated annealing to sample the possible values. The results show that gaps exist in the spectrum of values of the permanent but also show the existence of values for $d=8$ and 9 not hitherto observed. The new values in the spectrum of the permanent for $d=8$ is 72 and for $d=9$ are 119, 132, 136, 138, and 140. Further we found that the density of states grows exponentially with the permanent values. The larger the value of the permanent the larger the number of neighbours and the larger the energy barriers between the permanent values of the neighbours.

Keywords: Importance sampling; Simulated annealing; #P hard enumeration; Permanents; Handle basis representation

*Corresponding author. Present address: ENEA C.R.E. Casaccia, ERG-CNRT C-III 29 Via Anguillarese 301, I-00060 S. Maria di Galeria, Italy. Tel.: (+39)6-3048-4900; fax: (+39)6-3048-6357; e-mail: nourani@fys.ku.dk.

0304-3975/99/$ – see front matter
PII: S0304-3975(97)00130-8

1. Mathematical definitions

The permanent of a matrix was introduced by Cauchy in 1812. At that time, while developing the theory of determinants, he also defined a certain subclass of symmetric functions which later Muir named permanents [12]. Let $A=(a_{ij})$ be an $n\times n$ matrix. The permanent of A, written $Per(A)$, can be calculated in the same manner as its determinant, except that the signs of the summands are all positive:

$$Per(A)=\sum_{\sigma}^{n}\prod_{i=1}^{n}a_{i\sigma(i)}, \tag{1}$$

where σ runs over all permutations of $1,\ldots,n$. At first glance it would appear that this function is easier to compute than the determinant of the same matrix, but that is not true.

All known methods of calculation grow exponentially with the size of the matrix. In fact, the complexity of computing the permanent of an $n\times n$ (0,1)-matrix is NP-hard [6, 8, 17]. Valiant [17] proved that the permanent function is a member of the $\#P$ class. He also proved that the permanent problem is "complete" and that it is as hard as the number of satisfying assignments or number of "solutions" to a perfect matching in a bipartite graph. The direct evaluation of the permanent of an $n\times n$ matrix by formula (1) requires $O((n+1)!)$ multiplications. Ryser's formula for computing the permanent of a matrix involves $(n-1)(2^n-1)$ multiplications. Nijenhuis and Wilf's implementation of Ryser's method reduces the number of multiplications by a factor of two and the number of additions by a factor of $n/2$ [12]. This is the implementation used in the present work. More recently a group at the University of Edinburgh [7] came up with an exponential approximation algorithm for the permanent. But still Nijenhuis and Wilf's implementation of Ryser's method for exact computation is more efficient for general non-negative matrices.

In this paper we consider permanents of fully indecomposable $n\times n$ $(0,1)$-matrices. There is a one-to-one correspondence between digraphs (directed graphs) and $(0,1)$-matrices. Assuming that the vertices of a digraph G are numbered $1,\ldots,n$ in some arbitrary manner, then the adjacency matrix representation of G consists of an $n\times n$ matrix $A=(a_{ij})$ such that

$$a_{ij}=\begin{cases}1 & \text{if there is an edge from } i \text{ to } j,\\ 0 & \text{otherwise.}\end{cases} \tag{2}$$

Conversely, given an $n\times n$ (0,1)-matrix A, one can define a corresponding digraph $G(A)$ as follows: The vertices of $G(A)$ are the rows (or columns) of A and there is a directed edge from vertex i to vertex j iff $a_{ij}\neq 0$. Since we require self-loops in our digraphs [2] there are always ones on the diagonal of the corresponding matrix.

Central to our work is the notion of a handle basis for a finite strongly connected digraph G.

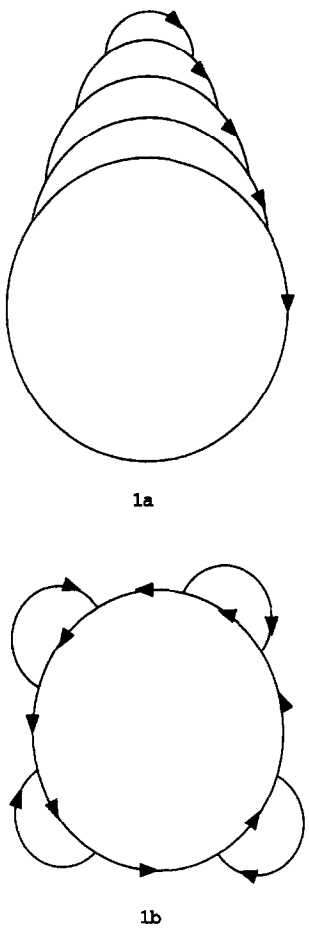

Fig. 1. Two different handle basis representations with $d=5$. The simple cycle is numbered 0 and each of the handles 1,2,3,4 is meeting another handle at its end points. (a) The permanent value for this digraph is 6, the minimum value in the range of permanents for $d=5$. (b) The permanent value for this digraph is 17, the maximum value in the range of permanents for $d=5$.

Definition. A handle basis $B=\{H[i],\ i=0,\ldots,d-1\}$ of a digraph G expresses G as the edge disjoint union of a sequence of subgraphs, where $H[0]$ is a simple cycle, and $H[i]$, $i>0$, is a simple path that meets the union of $H[j]$, $j<i$, exactly in its endpoints [2] (see Fig. 1).

A path $v_0,\ldots,v_n$ of length $n>0$ in the digraph G where v_i are the vertices of the digraph is simple if the vertices traversed are distinct. A simple cycle is a simple path with $v_0=v_n$. Points which are not endpoints are called interior points of the path. In other words, a handle is a path which starts from any previous handle and ends on any previous handle but is otherwise disjoint from the digraph of the previous handles.

It is sufficient to restrict our attention to the simplest configurations in which no two handles attach at the same vertices and which have vertices only at points of attachment of handles. By definition a directed graph admits a handle basis iff it is strongly connected. Donald et al. [2] have shown that if B is a handle basis for G, then every point (vertex) is interior to exactly one handle. In particular, this implies that handles do not cross each other. The number of handles d is the topological dimension of the digraph G. This dimension is also known as the cyclomatic number of G [2, 4, 5].

2. Bounds on the permanent

Minc [12] proved that the permanent of a non-negative $n \times n$ matrix $A = (a_{ij})$ is bounded by

$$\prod_{i=1}^{n} \sum_{t=1}^{i} a'_{it} \leqslant Per(A) \leqslant \prod_{i=1}^{n} \sum_{t=1}^{i} a^*_{it}, \tag{3}$$

where, if $A_i = (a_{i1}, \ldots, a_{in})$ is the ith row of A, then $(a^*_{i1}, \ldots, a^*_{in})$ denotes the n-tuple A_i arranged in non-increasing order and $(a'_{i1}, \ldots, a'_{in})$ denotes the same n-tuple arranged in non-decreasing order. Later Donald et al. [4, 5] proved that

$$Per(A) \leqslant 1 + \min\{\prod(r_i - 1), \prod(c_i - 1)\}, \tag{4}$$

where A is a fully indecomposable $n \times n$ matrix, and c_i and r_i are the column sums and row sums of A, respectively. Foregger [13] has obtained another bound for the permanent of an $n \times n$ matrix A with non-negative entries:

$$Per(A) \leqslant 2^{S(A)-2n} + 1, \tag{5}$$

where $S(A)$ is the sum of all the entries of A.

Since we are considering a matrix as well as its digraph representation, we show that the sum of all the ones in a (0,1)-matrix can be computed knowing just the number of vertices and handles of the digraph.

The number of ones in a $(0, 1)$*-matrix representation of a handle basis of a digraph* G *is*

$$S(A) = 2n + d - 1, \tag{6}$$

where d *is the number of handles and* n *is the number of vertices in the digraph.*

To prove this, suppose that digraph G has $n > 1$ vertices (see Fig. 1(b)). By definition each vertex has a self loop which means that each vertex is connected to itself; this represents n ones in the matrix. The digraph G is strongly connected, therefore at least one edge leads from each vertex to another vertex; this represents another n ones in the matrix. Also digraph G has d handles, each of which starts and ends on one of the previous handles, except the simple cycle. Therefore there are $d - 1$ vertices which

are connected to more than one vertex, implying another $d-1$ ones in the matrix. So in total there are $S(A)=2n+d-1$ ones in the matrix.

Our purpose below is to examine the possible values of the permanent of a d-dimensional strongly connected digraph. Let A be an $n\times n$ (0,1)-matrix with unit diagonal. The topological dimension d of the digraph $G(A)$ is then $S(A)-2n+1$, where $S(A)$ denotes the sum of entries in A [12]. Foregger's bound Eq. (5) can thus be rewritten as [2]

$$\mathrm{Max}[Per(A)]=2^{d-1}+1. \tag{7}$$

A related lower bound is [5]

$$\mathrm{Min}[Per(A)]=d+1. \tag{8}$$

Graphically the value in Eq. (7) is achieved for a digraph $G(A)$ in which all handles start and end on handle number zero $H[0]$ and are oriented in the same direction (see Fig. 1(a)). The value in Eq. (8) is achieved for a digraph $G(A)$ in which each handle $H[n]$ starts and ends on handle $H[n-1]$ in the same direction (see Fig. 1(b)).

3. Simulated annealing

Many problems which arise in practice are concerned with finding a best configuration. These kinds of problems are global optimisation problems. One can draw an analogy between annealing of solids and combinatorial optimisation problems. In statistical mechanics the ground state or meta-stable states of a system are preferentially populated at low temperature. Although low temperature is not a sufficient condition for finding the ground state, for many systems experiments show that by careful annealing and staying long enough at each temperature one may actually obtain the ground state [10].

Combinatorial optimisation is often applied to problems which are central to the disciplines of computer science and engineering, e.g. the travelling salesman problem, the hill climbing problem, and VLSI layout. Research in this area aims at developing efficient techniques for finding minimum or maximum values of a function called the cost function. Blind random search and iterative improvement can be used and do give acceptable approximations to the global minimum if only limited computer time is available. In order to improve the performance of these algorithms we need to escape from local minima and generate statistically independent configurations. In addition to comparing the cost of the current configuration with the cost of a newly generated one, an acceptance function can be evaluated on the basis of the two costs and a random number. If a true value is returned by the acceptance function, the newly generated configuration becomes the current configuration. Since the acceptance function returns true with a nonzero probability even for moves which increase the cost, this type of algorithm is called probabilistic hill climbing. Simulated annealing algorithms represent one class of such hill climbing algorithms.

3.1. The Metropolis algorithm

Annealing is the process of slowly cooling a physical system in order to obtain configurations with minimum energy. The cooling rate of annealing is usually very slow in order to reduce the number of defects in the structure of the system and thus to minimise the potential energy stored in the molecular configuration.

Metropolis et al. [11] introduced a statistical algorithm for simulating the state of a physical system at a temperature T. The Metropolis algorithm can be used to generate sequences of configurations of a combinatorial optimisation problem in which the cost function plays the role of energy and the control parameter is represented by the temperature. Let $S=\{s\}$ represent the state space. For the problem considered in this work, S is the set of all strongly connected d-dimensional digraphs. Let

$$E(s)=-Per(A(s)) \tag{9}$$

be the cost function and T the temperature. We assume that for each state s in S there exists a set $N(s)$, with $N(s)\subset S$, known as the move class or the set of neighbours of the state s. A "meta-graph" structure on a state is thus defined by $N(s)$, $s\in S$. Obviously, the move classes $N(s)$ must eventually connect the entire state space S. Otherwise, separate regions would not be able to equilibrate.

At each step of the Metropolis algorithm a state $s'\in N(s)$ is selected at random from the neighbourhood of the current state s. If the energy $E(s')$ is lower than the energy $E(s)$, the acceptance function returns a true value. In this case the move is accepted, and the new state becomes the current state. If the energy of the new state is higher than the energy of the current state, the acceptance probability for s' to become the next state is

$$P_{\text{accept}}=\exp\left(\frac{-\Delta E}{T}\right), \tag{10}$$

where $\Delta E=E(s')-E(s)$. A pseudo-random number is generated with a uniform distribution on [0,1]. If the random number is less than the acceptance probability, the move is accepted, otherwise it is rejected and the random walk remains in the old state s for another time step [15, 18]. The basic Metropolis algorithm is shown in pseudo-C in Fig. 2.

Simulated annealing uses the Metropolis algorithm with a gradually decreasing temperature. Initially the control parameter T is given a very high value which will assure acceptance of essentially any proposed state. This is then lowered in each step until a given time or temperature is reached. Therefore a combinatorial optimisation problem can be solved as outlined in pseudo-C in Fig. 3.

Simulated annealing probabilistically transforms each current state to one of its neighbours. This random walk can be described by a Markov chain. Markov chains are discrete stochastic processes in time and state with the property that the next state does not depend on the states that have preceded the current state [9]. For a Markov chain

```
Metropolis (T)
float T;
{
initialize:

        GetNewState(CurrentState, NewState);
        ΔE = E(NewState) - E(CurrentState);
        if (ΔE < 0.0)
        CurrentState = NewState;
        else {
        if (Random < (exp(-ΔE/T)))
        CurrentState = NewState;
        }
}
```

Fig. 2. Description of the Metropolis algorithm in pseudo-C.

```
SimulatedAnnealing ()
{
initialize:

        CurrentState = GenerateState;
        T = SomeConstant;
        stop = SomeOtherConstant;
        while ( T > stop ) {
        Metropolis(T);
        T = decrease(T);
        }
}
```

Fig. 3. Description of the simulated annealing algorithm in pseudo-C.

on n states the transition probabilities form an $n \times n$ matrix $P = (p_{ss'})$

$$p_{ss'} = \begin{cases} \dfrac{1}{|N(s)|} & \text{if } s' \in N(s), \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $|N(s)|$ is the number of neighbours of s. In addition

$$\forall s = 1, \ldots, n \quad \sum_{s'=1}^{n} p_{ss'} = 1 \tag{12}$$

which means that P is a stochastic matrix.

3.2. The move class

As part of our Monte Carlo approach we will introduce a random walk on the space of strongly connected d-dimensional digraphs. This in turn requires introducing a digraph structure on this space. In each digraph handles are numbered from 0 to $d-1$, where handle number 0 represents the simple cycle. As mentioned before our interest is in strongly connected digraphs. Since we are considering digraph structures,

each handle has a direction. The vertex which starts a handle is called its tail and the vertex which ends a handle is called its head. We employ two ways of moving a handle: one is reversing its direction, the other is pushing its head or tail which means that its point of attachment moves in either direction, randomly chosen, on the handle with lower handle value.

We require that all moves preserve strong connectivity and define our move class to be any digraph obtained by one of the following operations:

(1) For any handle number zero through $d-1$, reverse the direction of the handle.

(2) For any handle number one through $d-1$, either reverse the direction of the handle or push its head past the next confluence point (in either direction), or push its tail past the next confluence point (in either direction). In either of the two latter cases the handle which makes the move should move over handles with lower handle number, otherwise the move will not be accepted. This move class satisfies the following theorem:

Theorem. *In a handle basis B of a digraph G, the above move class preserves strong connectivity.*

Proof. Let $B=\{H[i],\ i=0,\ldots,d-1\}$ be the handle basis of a strongly connected digraph G. Let $H[n]$ be the handle to be moved. There exist two sub-digraphs $A=\{H[j],\ j=0,\ldots,n-1\}$ and $C=\{H[j],\ j=n+1,\ldots,d-1\}$ in which handle $H[n]$ will move over A. The move can be of two types by definition. If the move is of type 1, then reversing the direction of $H[n]$ will have no effect on its strong connectivity property since the head and the tail of $H[n]$ are interior vertices of A. If the move is of type 2, meaning moving the head or the tail of $H[n]$ over handles with lower numbers in A, for the same reason as for a type 1 move, the tail of a moved handle can be reached from the head through the lower handles and the head directly from the tail though itself. Further, C is either connected to A through $H[n]$ or directly connected to A. In the former case $H[n]$ does not lose its strong connectivity property, therefore C does not lose its strong connectivity either. In the latter case, nothing changes on sub-digraph C, therefore it remains strongly connected.

In a previous move class [14] we had to check for strong connectivity after each attempted move. The difference between the two move classes is in step 2 where previously any randomly picked handle could move over any other handle, and this might result in broken connectivity. With the new move class we do not have to check for strong connectivity of the digraph after each move since the strong connectivity is preserved by this move class. Fig. 4 shows the number of moves visiting a particular energy during one run. It can be seen that the new move class visits states of large permanent value (low energy) more frequently than the previous move class does.

3.3. Neighbourhood of the global minimum

In order to investigate the neighbourhoods of minima we took the last state at the end of each annealing process as the starting point for two thousand attempted

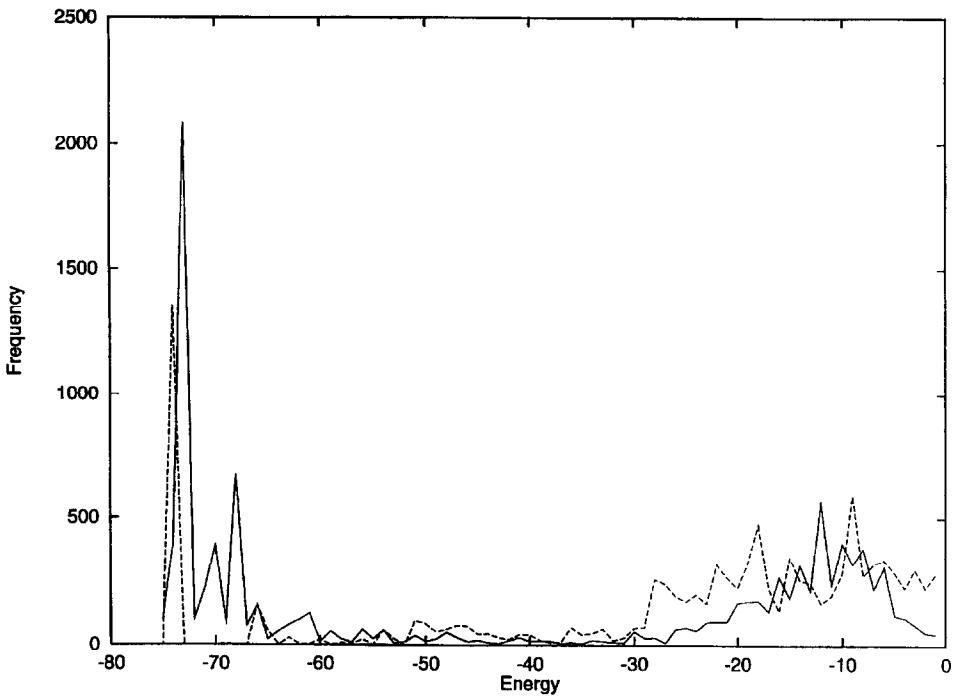

Fig. 4. The number of attempted moves to reach different energies (equal to minus the permanents) during annealing runs using the new (solid line) and old (dashed line) move classes, $d = 8$. The new move class reaches the lower energies faster than the old move class.

moves. These two thousand neighbours, chosen at random, were then compared with an exhaustive search of the neighbourhood of the state. The results show that the two searches, statistical and exhaustive, are in good agreement with each other. We have also looked at the last two thousand attempted moves before reaching the minimum state. Fig. 5 shows a comparison of these different sets of data. It is obvious that the states visited before reaching the minimum state are more diverse than the neighbours of the minimum states. This shows that the program did not reach the minimum state long before the program stopped, indicating that the annealing schedule was adequate and did not waste time sitting in the minimum state, which in this case is the global minimum.

3.4. Temperature schedule

Choosing the annealing schedule $T(t)$, including the starting and stopping temperatures, is a major issue in simulated annealing. Kirkpatrick et al. [10] suggested an exponentially decreasing cooling rate, $T_n = (T_1/T_0)^n T_0$, where $T_1/T_0 = \alpha$ is the variable rate parameter of the calculation. α will usually be close to unity, and the closer α is to unity the slower and more careful is the calculation. Sometimes a number of Metropolis steps are taken at each temperature. A key element in this cooling schedule is the choice of α. Cooling too fast will result in a system frozen into a meta-stable

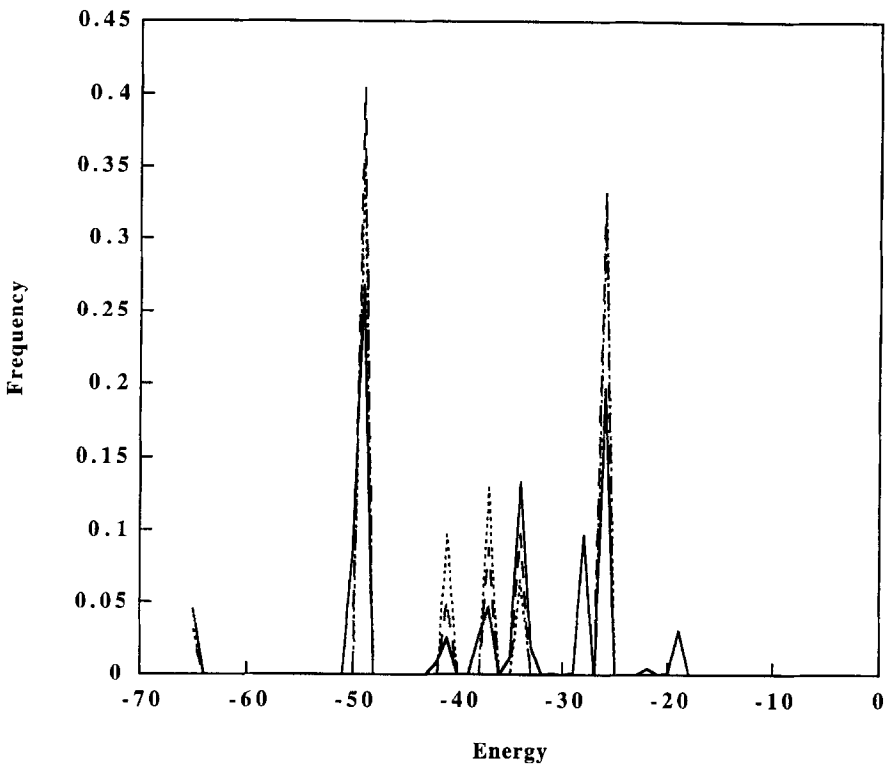

Fig. 5. 2000 attempted moves in the neighbourhood of the ground state using exhaustive (long dashed line) and random search (dashed line). Also the last 2000 attempted moves before reaching the ground state are shown (solid line). Good agreement between random and exhaustive search is seen. The states being visited before reaching the ground state are more diverse than the neighbours of the ground state, implying that the annealing schedule was adequate.

state far from the ground state. Cooling too slowly will waste computer time. Similar arguments hold for the choice of initial and final temperatures.

This exponential cooling schedule has been used in a large number of papers due to its simplicity, including this study. However a more efficient annealing is achieved by using an adaptive schedule based on finite-time thermodynamics [1].

Here we have chosen the starting temperature $T_0 = 2 \times \mathrm{Max}[Per]$, such that it is sufficiently high that the system is able to pass all energy barriers and get out of local minima. We obtained the best results with the rate parameters $\alpha = 0.9995$ and the stopping temperature $T_{\mathrm{stop}} = 0.5$.

4. Permanent values

The spectrum of possible values for the permanent of a $(0,1)$ fully irreducible matrix is not yet fully understood. There has not been much work on the missing values (gaps)

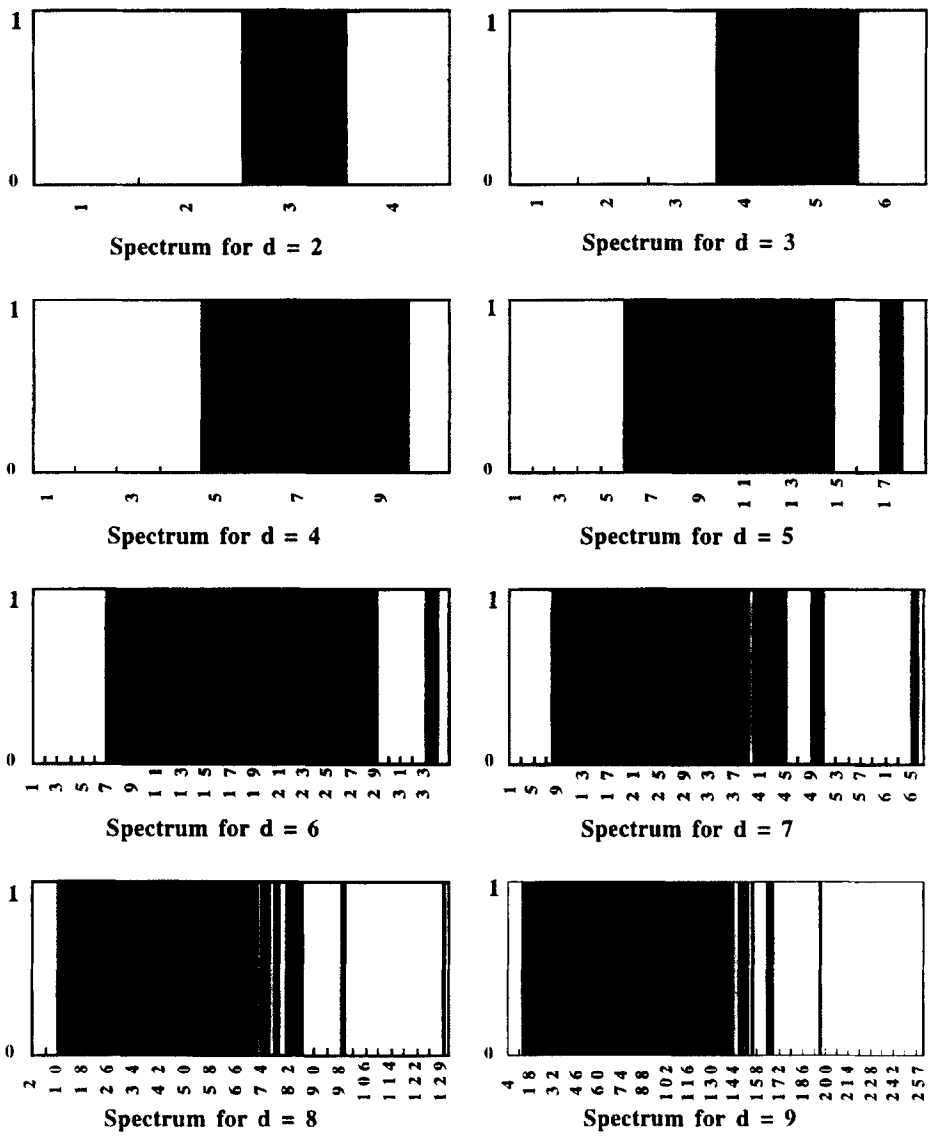

Fig. 6. Spectra of values of the permanent for $d = 2, \ldots, 9$. One represents permanent values which have been seen and zero represents gaps.

in the range of possible permanents. In our search we have found just a handful of papers. Some work came out of a group in San Diego State University [2–5].

In this work we have searched for some of the missing permanent values. Using graph theory enabled us to examine thousands of different permutations of the matrix while calculating the permanent values of these matrices. Fig. 6 shows the range of permanent values seen in this study for different topological dimensions d. Simulated annealing enabled us to focus our search in the range of permanents where value(s) are

missing. While simulated annealing is usually used to search for global minima, for the problem stated in this work the global minimum (maximum value of the permanent) is already known. However, most missing values are close to the global minimum. Therefore, after using different temperature schedules, we have found the range of temperatures where most of the permanent values around the gaps have been seen and thus used temperatures in that range for doing a very careful annealing with several hundred attempted moves in each time step. In our search during the simulation we have found 1 new value of the permanent of a 14 by 14 matrix, $d=8$ namely 72 and 5 new values of the permanent of a 16 by 16 $(0,1)$-matrix, $d=9$, which previously were believed not to exist. These new values are 119, 132, 136, 138, and 140 (See Fig. 6).

We have found one unpublished paper by Donald et al. [3] in which the authors transform the permanent problem into another problem, counting the Number of Independent Sets (NIS) for Handle Order Graphs (hog). There they prove the existence of gaps for $d \geqslant 9$. As acknowledged in that paper, this relationship between the gaps and NIS is not clear, since some bounds are not sharp. The new range, with the newly found permanent values, still does not satisfy this theorem, since gaps exist which are not covered by the formula in their theorem. We have done a very careful search to find these missing values, and we have found some, but there are still values in the spectra of the permanents of $(0,1)$ fully irreducible matrices which we cannot be sure are real gaps. For the small digraphs we will have smaller matrix representations and therefore an exhaustive search can confirm the existence of gaps.

5. Thermodynamic portraits

We used the technique introduced in [1] to estimate the thermodynamic properties of the system. From the data collected during the annealing process at infinite temperature we were thus able to estimate the density of states $P(E)$, the mean energy $E(T)$, the heat capacity $C(T)$, and the relaxation time $\varepsilon(T)$.

As mentioned in Section 3.1, the random walk during the annealing process can be described by a Markov chain. The transition probabilities from any state i to any other state j is

$$p_{ij} = Pr(X_{t+1} = j \mid X_t = i) \tag{13}$$

Assuming that state i has $|N(i)|$ neighbours, the transition probability matrix at infinite temperature is defined by Eq. (11). The matrix P is a stochastic matrix since each of its rows sums to one. Then for a stochastic matrix P and an occupation probability vector m satisfying microscopic reversibility

$$P_{ij} m_i = P_{ji} m_j \tag{14}$$

The stationary distribution of a Markov chain is this unique (normalised) vector m for which

$$Pm = m \tag{15}$$

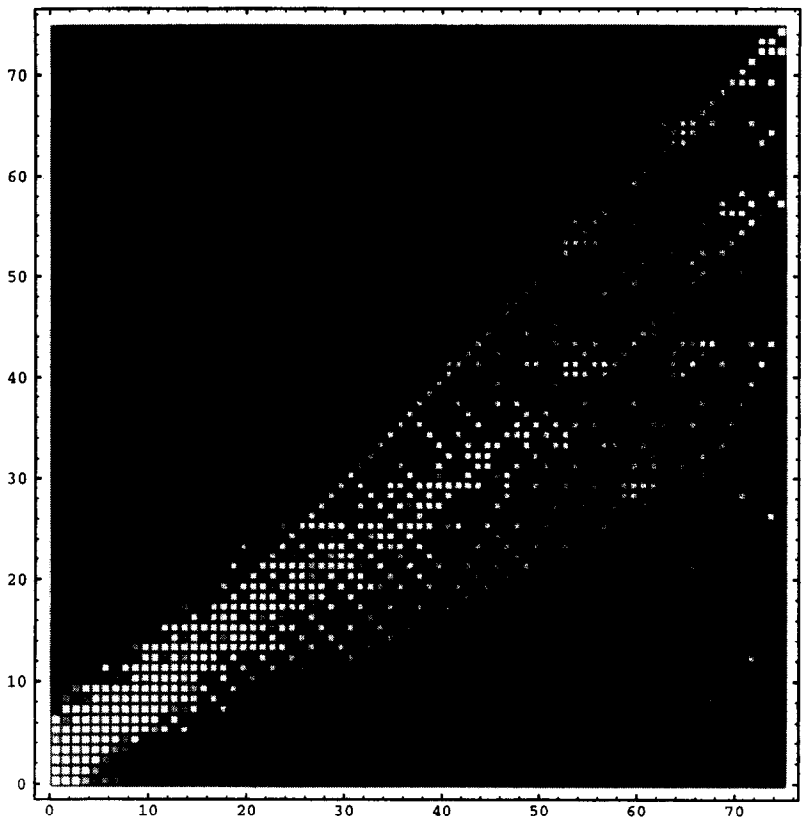

Fig. 7. Density plot of the 75×75 P-matrix linking energies for $d = 8$.

Most thermodynamic properties of a system can be deduced from its transition probability matrix P, the elements of which are the infinite temperature transition probabilities between different states [1].

In principle these states should be the individual micro-states of the system. However, their number is usually extremely large, making the matrix P totally unmanageable. Instead one may, to a good approximation, work with energy levels instead, i.e. lumping all states of a given energy into a single level. This reduces the size of the problem by orders of magnitude.

The transition probabilities can be estimated during the Metropolis algorithm by keeping track of *attempted* moves in a matrix Q where $Q_{ij} =$ number of attempted moves from i to j. A good estimate of the transition probability matrix is then [1]

$$P_{ij} = \frac{Q_{ij}}{\sum_k Q_{ik}} \tag{16}$$

Fig. 7 shows a density plot of the 75×75 P matrix linking all permanent values for $d = 8$. Having obtained the P matrix, we can compute the stationary distribution m,

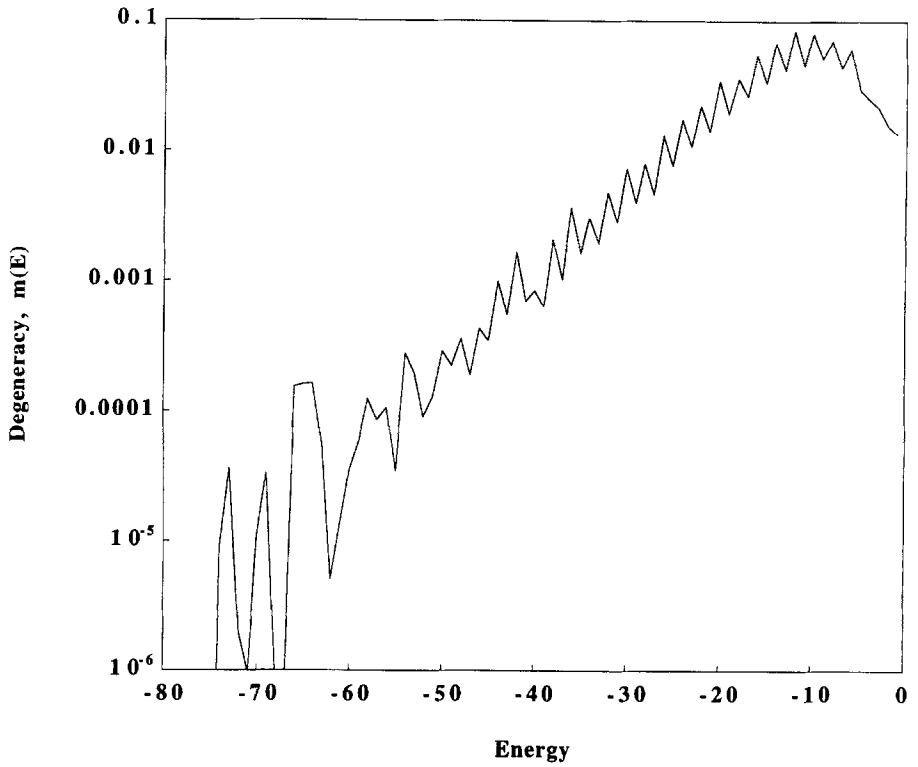

Fig. 8. Density of states $m(E)$ as a function of state energy for $d = 8$. Note that the degeneracy of states is growing exponentially with energy.

which is also the degeneracy of the energy levels (i.e. number of microscopic states in that energy level). From there the partition function

$$Z(T) = \sum_j m_j e^{-E_j/T} \tag{17}$$

can be computed and one can estimate all thermodynamic properties, e.g. average energy $E(T)$, heat capacity $C(T)$, and relaxation time $\varepsilon(T)$ in the usual statistical mechanical manner.

We have chosen graphs with 5, 6, 7, and 8 handles. The ranges of the energies (equal to minus the permanents) of the matrix representations are: $[-17,-6]$, $[-33,-7]$, $[-65,-8]$, and $[-129,-9]$, respectively. In each of these ranges there exist gap(s) in the energy range. We closed up these gap(s) artificially in order to speed up computation of the equilibrium values of the system, Fig. 6. Since the Metropolis acceptance Eq. (10) for uphill moves is reduced by the factor $\exp(-\Delta E/T)$, removing large voids will reduce ΔE and thus increase P_{accept}. Consequently the ranges change to: $[-10,-1]$, $[-21,-1]$, $[-38,-1]$, and $[-75,-1]$, respectively. As can be seen in some cases, the size of the matrix in reduced by almost 40% and as a result it speeds up the computa-

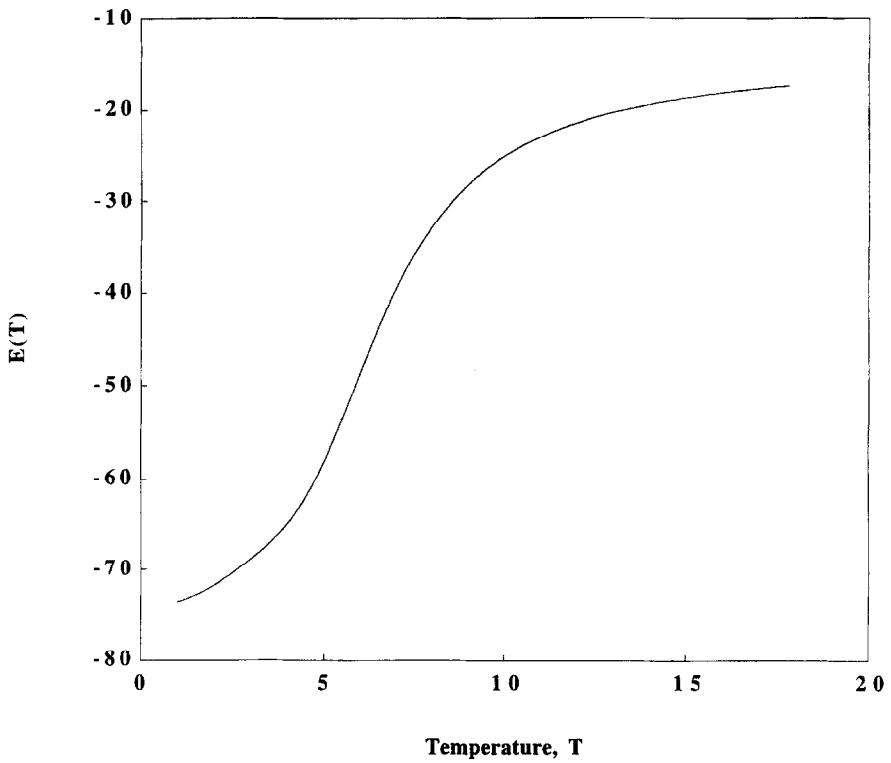

Fig. 9. Average energy $E(T)$ as a function of heat bath temperature during the annealing run for $d = 8$.

tion of the permanent value considerably. Figs. 8–11 show the density of states $m(E)$, average energy $E(T)$, heat capacity $C(T)$, and relaxation time $\varepsilon(T)$, respectively. The data was obtained using the Metropolis algorithm with 5×10^6 steps taken during the annealing process for $d = 8$.

As shown in Fig. 8, the degeneracy of these graphs is growing exponentially with energy. Keeping track of the graphs encountered along the annealing process, we observed that the higher the energy the larger the variety of the graphs with the same energy. This may explain the growth of the degeneracy. Throughout the simulation we have also noticed that the lower the energy the more handles are attached to the simple cycle. For example, at the global minimum all handles are on the simple cycle. Different graphs with the same minimum value have all their handles on the simple cycle. The only difference between these graphs is the direction of the handles. But when the handles are moved and are placed on top of each other, one finds a larger variety of graphs with the same energy. Another observation is that the higher the energy the smaller the number of neighbours with different energy and the smaller the energy barriers between the neighbours.

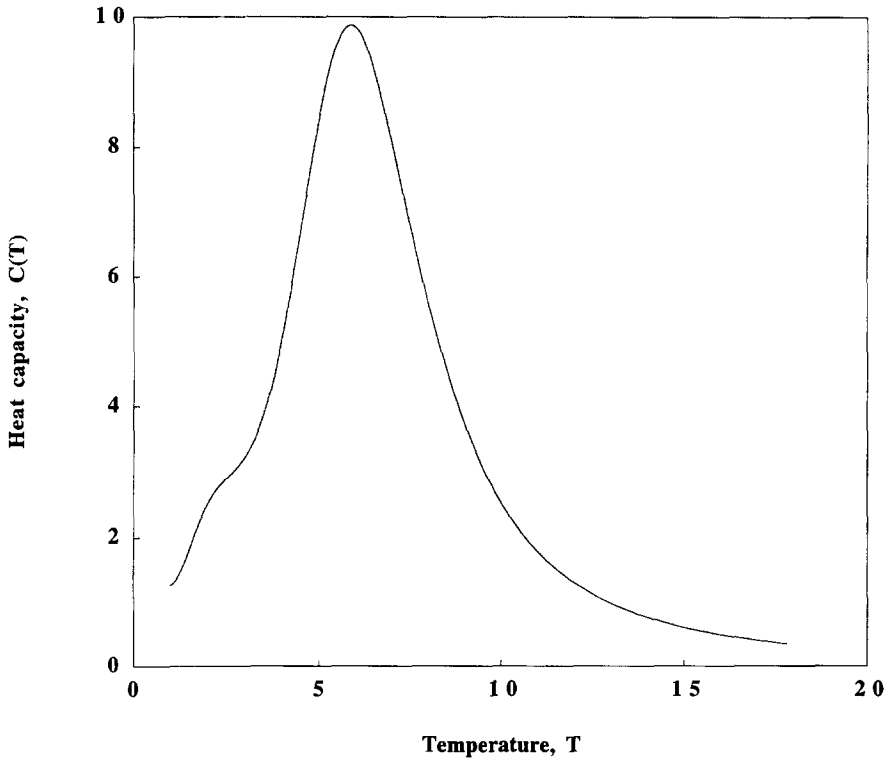

Fig. 10. Heat capacity $C(T)$ as a function of heat bath temperature during the annealing run for $d = 8$.

6. Applications

Permanents are connected to problems in combinatorics and in applied problems of enumerative nature. Especially permanents of doubly stochastic matrices have applications in probability, combinatorics, and statistical mechanics. Some of the applications of permanents are in graph theory and $\#P$ complete enumeration problem [7, 16].

Caianiello [12] used permanents in connection with renormalization problems in quantum field theory. He used permanents and Hafnians to express in algebraic form perturbation expansions of field theory for describing Boson propagators in the same way as determinants and Pfaffians appear in the expressions related to Fermion propagators. In the areas of physics and chemistry where statistical methods are used to study the phenomena that are outcomes of the combined action of a very large number of items permanents are also useful. Such problems in order-disorder statistics, in solid-state chemistry, and in physical chemistry can be formulated as enumeration problems involving lattices [12].

The use of the simulated annealing algorithm in the present study has been rather unconventional since we did not seek the minimum energy state—that was known at the outset—but rather used simulated annealing to perform a random search for new

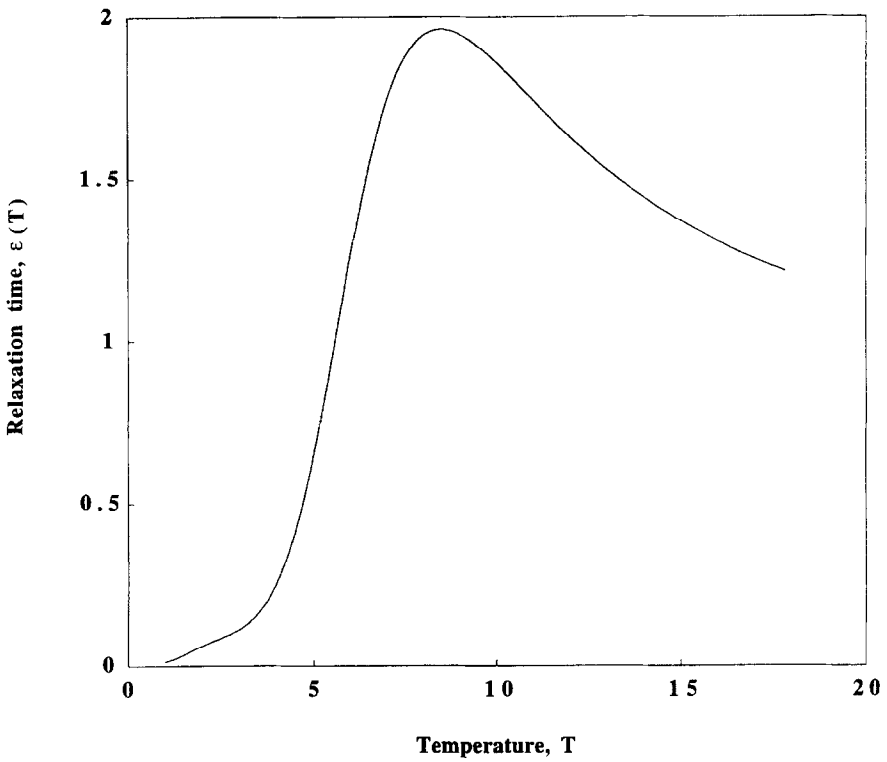

Fig. 11. Relaxation time $\varepsilon(T)$ as a function of heat bath temperature during the annealing run for $d=8$.

permanent values strongly weighted towards the low energy region. A similar biased search may be useful in other situations as well.

Acknowledgements

YN wishes to express her sincere gratitude to Prof. P. Salamon for his generous guidance during his visit to the Ørsted Laboratory. YN also would like to thank the Ørsted Laboratory, University of Copenhagen for their support and hospitality. This work has been partially supported by EU contract no. CHRX-CT92-0007.

References

[1] B. Andresen, K.H. Hoffmann, K. Mosegaard, J. Nulton, J.M. Pedersen, P. Salamon, On lumped models for thermodynamic properties of simulated annealing problems, J. Phys. France 49 (1988) 1485–1492.
[2] J. Donald, J. Elwin, On the structure of the strong orientation of a graph, SIAM J. Discrete Math. 9 (1) (1993) 30–43.
[3] J. Donald, J. Elwin, R. Hager, The full handle support theorem, independent sets, and bounds on the permanent, 1993, draft copy.

[4] J. Donald, J. Elwin, R. Hager, P. Salamon, A graph theoretic upper bound on the permanent of a nonnegative integer matrix, I, II, Linear Algebra Appl. 61 (1984) 187–198.
[5] J. Donald, J. Elwin, R. Hager, P. Salamon, Handle bases and bounds on the number of subgraphs, J. Combin. Theory, Ser. B, 42 (1987) 1–13.
[6] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, 1979.
[7] M. Jerrum, U. Vazirani, A mildly exponential approximation algorithm for the permanent, Proc. 33rd IEEE Conf. on Foundations of Computer Science, IEEE Computer Soc. Press, Silver Spring, MD, 1992, pp. 320–326.
[8] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.
[9] R.G. Kemeny, J.L. Snell, Finite Markov Chains, Van Nostrand, Princeton, NJ, 1960.
[10] S. Kirkpatrick, C.D. Gelatt, Jr., M.P. Vecchi, Optimization by Simulated Annealing, Science 220 (1983) 671–680.
[11] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculation by fast computing machines, J. Chem. Phys. 21 (1953) 1087–1092.
[12] H. Minc, Permanents, Encyclopedia of Mathematics and its Applications, vol. 6, section on Linear algebra, Addison-Wesley, Reading, MA, 1978.
[13] A. Nijenhuis, H. Wilf, Combinatorial Algorithms, Academic Press, 1975.
[14] Y. Nourani, Exploring the spectrum values of the permanents by simulated annealing, C.J. Colbourn, E.S. Mahmoodian (Eds.), Combinatorics Advances, Kluwer Academic Publishers, Dordrecht, 1995, pp. 255–265.
[15] R.H.J.M. Otten, L.P.P. Van Ginneken, The Annealing Algorithm, Kluwer Academic Publishers, Dordrecht, 1989.
[16] A. Sinclair, Algorithms for Random Generation and Counting: A Markov Chain Approach, Birkhauser, Boston, 1993.
[17] L.G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (1978) 189–201.
[18] P.J. Van Laarhoven, E.H.L. Aarts, Simulated Annealing and Applications, Reidel, Dordrecht, 1987.