Philips J. Res. ***40,*** *193-226, 1985* *R 1109*

STATISTICAL COOLING: A GENERAL APPROACH TO COMBINATORIAL OPTIMIZATION PROBLEMS

by E. H. L. AARTS and P. J. M. VAN LAARHOVEN

Philips Research Laboratories, 5600 JA Eindhoven, The Netherlands

Abstract

Statistical cooling is a new combinatorial optimization technique based on Monte-Carlo iterative improvement. The method originates from the analogy between the annealing of a solid as described by the theory of statistical physics and the optimization of a system with many degrees of freedom. In the present paper we present a general theoretical framework for the description of the statistical cooling algorithm based on concepts from the theory of Markov chains. A cooling schedule is presented by which near-optimal results can be obtained within polynomial time. The performance of the algorithm is discussed by means of a special class of traveling salesman problems.

Math. Rev.: 05,49,60,82.

1. Introduction

During the last decade much effort has been invested in the subject of combinatorial optimization[1]). The present-day interest in this field originates, to a large extent, from the need to solve the many combinatorial optimization problems related to computer science and VLSI design[2-4]). Examples include the traveling salesman problem, one- and two-dimensional cell arrangement, macro-placement, global routing and logic minimization. The study of combinatorial optimization focusses on finding techniques to optimize a function of a finite, possibly large, number of variables. This so-called cost function is a quantitative measure of the 'goodness' (with respect to some critical quantities) of the system that is to be optimized and it depends on details of the internal configurations of the system.

Many combinatorial optimization problems have been proven intractable and belong to the class of NP-complete problems, i.e. no algorithm is known that gives an exact solution to the problem within polynomial time (the computation time increases at lest exponentially with the complexity of the problem). If, furthermore, such an algorithm were found, it could be transformed to solve all the problems of the corresponding class in polynomial time. Garey

and Johnson extensively studied this type of combinatorial optimization problems and provided a useful classification of the various problems [2]).

As a result of their intractability, problems with a large complexity cannot be solved exactly within a realistic amount of computation time. Less time-consuming algorithms can be constructed by applying heuristic methods striving for near-optimal solutions. A vast number of different heuristic algorithms has been reported in the literature [1-3,5,6]). The heuristics applied in these algorithms is in most cases strongly problem dependent. This is considered as a major drawback since it prohibits flexible implementation and general application.

Statistical cooling is a new heuristic optimization technique that is generally applicable, easy to implement and its optimization results can be as good as desired. The algorithm is based on Monte-Carlo methods and in a way it may be considered as a special form of iterative improvement [6]) (see also secs 2 and 3). The technique originates from computer simulation methods used in condensed matter physics and was first employed successfully for combinatorial optimization problems by Kirkpatrick et al. [7]). It, ever since, has found great acceptance and has been applied to a number of problems in various disciplines [8]). The work of Kirkpatrick is strongly based on the analogy between the annealing of a solid and the optimization of a system with many independent variables. Solids are annealed by raising the temperature to a (maximal) value for which the particles randomly arrange in the liquid phase, followed by cooling to force the particles into the low energy states of a regular lattice. At high temperatures all possible states can be reached, although low energy states are occupied with a larger probability; lowering the temperature decreases the number of accessible states and the system finally will be frozen into its ground state, provided the maximum temperature is high enough and the cooling is sufficiently slow.

In combinatorial optimization a similar situation occurs: the system may occur in many different configurations. Any configuration has a cost that is given by the value of the cost function for that particular configuration. Similar to the simulation of the annealing of solids, one can statistically model the evolution of the system that has to be optimized into a state that corresponds with the minimum value of the cost function.

So far the literature provides a number of papers that deal predominantly with applications of the statistical cooling algorithm to various combinatorial optimization problems [7-18]) and in most cases the description of the algorithm starts off from the physical analogy formulated by Kirkpatrick et al. [7]). We felt that a more formal theoretical description and analysis of the algorithm would be desirable.

The intention of the present paper is to provide a theoretical formalism and an analysis of the convergence of the statistical cooling algorithm. The formulation starts off with an introductory section on iterative improvement (sec. 2). Section 3 presents a mathematical formalism for the statistical cooling algorithm based on the theory of Markov chains [19]). The convergence of the algorithm is discussed in sec. 4. In this section a number of formal concepts and additional functions are introduced that are useful for the understanding of the convergence behaviour of the algorithm. In sec. 5 we present an analysis and a discussion of the results obtained in sec. 4 by applying the statistical cooling algorithm to a special class of traveling salesman problems. In sec. 6 the convergence of the algorithm is re-examined. Here we introduce a number of heuristic approximations by which faster convergence of the algorithm is achieved. In this section we, furthermore, present a new cooling schedule for the statistical cooling algorithm with emphasis on feasibility and generality. The results of sec. 6 are illustrated and discussed in sec. 7 by applying the algorithm with the new cooling schedule to the traveling salesman problems of sec. 5. A summary of the algorithm is presented in sec. 8. The paper ends with some conclusions and remarks.

2. Iterative improvement

The problem of finding near-optimal solutions for a given combinatorial optimization problem is solved by optimizing heuristically the corresponding cost function. The cost function is a function of many variables and represents a quantitative measure of the 'goodness', with respect to certain criteria, of any internal configuration (or state) of the system that has to be optimized.

A system configuration is represented by a state vector $\boldsymbol{r} \in \mathscr{R}$ whose components uniquely determine the given configuration. The configuration space $\mathscr{R}$ is given by the finite set of all possible system configurations. We, furthermore, introduce the set $I_{\mathscr{R}} = \{1, 2, \ldots, |\mathscr{R}|\}$ as the set of state labels of the system configurations contained in $\mathscr{R}$. The cost function $C(\boldsymbol{r}_i)$, with C: $\mathscr{R} \rightarrow \mathbb{R}$ and $i \in I_{\mathscr{R}}$, assigns a real number to each state. Here, the cost function is defined in such a way that the lower the value of $C(\boldsymbol{r}_i)$ the better is the configuration represented by the corresponding state vector. The solution of the optimization problem is given by a configuration that corresponds with a state vector obtained by minimizing the cost function.

Iterative improvement [6]) is the best known heuristic optimization method for combinatorial optimization problems. The algorithm starts off with a given state, i, with state vector $\boldsymbol{r}_i$ and initial cost $C(\boldsymbol{r}_i)$. Next, by rearranging the system a new configuration, j, is generated with corresponding state vector $\boldsymbol{r}_j$ and cost $C(\boldsymbol{r}_j)$. The difference in cost, given by

$$\Delta C_{ij} = C(r_j) - C(r_i), \tag{1}$$

determines whether the new configuration is accepted or not, i.e. for negative values of ΔC_{ij} the cost decreases and the new configuration is accepted, whereas for positive values the cost increases and the new configuration is rejected. This procedure is then repeated until no further improvement is achieved.

The inherent limitation of the iterative improvement algorithm, when employing relatively simple rearrangements, is that it usually gets stuck in a local minimum: To avoid this problem some tricks must be applied to assure that the final solution is close to a global minimum. If the algorithm gets stuck one may resort to more complicated rearrangements [6]). The process of choosing the appropriate complex rearrangements is strongly determined by the nature of the problem and requires expert knowledge. Many combinatorial optimization problems in the field of computer science and VLSI design, however, require fast and flexible implementation. Optimization methods that incorporate complex system rearrangements and problem dependent manipulations do not meet these requirements.

3. Statistical cooling

A more general and flexible heuristic optimization method is given by the statistical cooling technique, also called simulated annealing [7]) or Monte-Carlo annealing [10]), which can be regarded as a Monte-Carlo iterative improvement method. This method accepts in a limited way deteriorations of the system, i.e. configurations that correspond with an increase in the cost function.

Similar to the iterative improvement algorithm the statistical cooling algorithm starts off with a given initial configuration and generates a sequence of new system configurations by rearranging the system. New configurations are accepted according to an acceptation criterion which allows also for deteriorations of the system. Initially the acceptation criterion is taken such that system deteriorations are accepted with large probabilities. As the optimization process proceeds the acceptation criterion is modified in such a way that the probability for accepting system deteriorations becomes smaller and smaller, and at the end of the optimization process this probability approaches zero. In this way the optimization process is prevented from getting stuck in local minima, thus enabling it to arrive at a global minimum, employing only simple rearrangements of the system. The decrement in the probability for accepting system deteriorations is governed by a so-called cooling control parameter. It is possible to decrease this probability during the optimization process after each system rearrangement. Here we use an approach in which the probability in between decrements is kept constant for a number of system rearrange-

ments. The advantage of this approach is that the optimization process can be described in terms of a finite set of homogeneous Markov chains [19]) for which the limit behaviour is more easy to describe than for inhomogeneous Markov chains [19]). This results in a more transparent description of the statistical cooling algorithm.

A Markov chain is a sequence of states for which the probability of occurrence is determined by the transition probability from state i to state j defined as

$$T_{ij}(\beta) = B_{ij}(\beta)\, P_{ij} \qquad \text{if } i \neq j \tag{2}$$

and

$$T_{ij}(\beta) = 1 - \sum_{\substack{k=1 \\ k \neq i}}^{|\mathcal{R}|} B_{ik}(\beta)\, P_{ik} \qquad \text{if } i = j, \tag{3}$$

where $T_{ij}(\beta)$ denotes the transition probability ($T: \mathcal{R} \times \mathcal{R} \to [0,1] \subset \mathbb{R}$), P_{ij} the perturbation probability ($P: \mathcal{R} \times \mathcal{R} \to [0,1] \subset \mathbb{R}$) to generate state j from state i, $B_{ij}(\beta)$ the acceptation probability ($B: \mathcal{R} \times \mathcal{R} \to (0,1] \subset \mathbb{R}$) to accept state j if the system appears in state i, $|\mathcal{R}|$ the size of the configuration space (total number of states) and $\beta \in \mathbb{R}^+$ the cooling control parameter. Note that the probabilities T_{ij}, P_{ij} and B_{ij} are conditional probabilities, i.e. $T_{ij}(\beta) = T(j \mid i; \beta)$, $P_{ij} = P(j \mid i)$ and $B_{ij}(\beta) = B(\text{acc. } j \mid i, j; \beta)$. Note that T is a stochastic matrix, i.e. $\forall i \in I_{\mathcal{R}}: \sum_{j=1}^{|\mathcal{R}|} T_{ij}(\beta) = 1$.

For each state i a configuration sub-space $\mathcal{R}_i$ exists whose elements are given by the set of configurations that can be reached by exactly one system rearrangement starting from state i, i.e.

$$\mathcal{R}_i = \{r_j \in \mathcal{R} \mid P_{ij} \neq 0\}. \tag{4}$$

Perturbation of the system is usually done by simple random system rearrangements. Consequently the perturbation probabilities P_{ij} are uniformly distributed over the configuration sub-spaces, i.e.

$$P_{ij} = \begin{cases} |\mathcal{R}_i|^{-1} & \text{if } j \in I_{\mathcal{R}_i} \\ 0 & \text{if } j \notin I_{\mathcal{R}_i}. \end{cases} \tag{5}$$

Next we assume that any two states i and j with $i, j \in I_{\mathcal{R}}$ are connected by a finite sequence of perturbations with non-negative probability, i.e.

$$\forall i, j \in I_{\mathcal{R}},\ \exists K_{ij} \in \mathbb{N},\ \{k_1, k_2, \ldots, k_{K_{ij}}\} \subset I_{\mathcal{R}}:$$
$$k_1 = i,\ k_{K_{ij}} = j \quad \text{and} \quad P_{k_l k_{l+1}} \neq 0,\ \forall l \in \{1, \ldots, K_{ij} - 1\}. \tag{6}$$

This ensures connectivity of the configuration space and it implies, furthermore, that T is a primitive matrix. Indeed, by taking $K = \max_{i,j \in I_{\mathcal{R}}} K_{ij}$ we obtain $T^K > 0$.

We, furthermore, impose the following conditions

(i) $\forall\, i, j \in I_{\mathcal{R}} : P_{ij} = P_{ji}$ (symmetry), (7)

(ii) $\forall\, i, j \in I_{\mathcal{R}} : B_{ij}(\beta) = 1$ if $\Delta C_{ij} \leqslant 0$, (8)

(iii) $\forall\, i, j \in I_{\mathcal{R}} : B_{ij}(\beta) \in (0, 1) \subset \mathbb{R}$ if $\Delta C_{ij} > 0$, (9)

with special conditions

$$\lim_{\beta \to \infty} B_{ij}(\beta) = 1 \qquad (\Delta C_{ij} > 0) \tag{9a}$$

and

$$\lim_{\beta \downarrow 0} B_{ij}(\beta) = 0, \qquad (\Delta C_{ij} > 0), \tag{9b}$$

(iv) $\forall\, i, j, k \in I_{\mathcal{R}}$ with $C(r_k) \geqslant C(r_j) \geqslant C(r_i)$:

$$B_{ik}(\beta) = B_{ij}(\beta)\, B_{jk}(\beta). \tag{10}$$

Condition (iii) contains the kernel of the statistical cooling method, i.e. deteriorations of the system ($\Delta C_{ij} > 0$) are accepted with non-zero probability.

A Markov chain is formed by starting off with a given initial configuration and applying the transition mechanism governed by eq. (2) a number of times. Such a Markov proces has an important property: there exists a unique equilibrium vector $q(\beta) \in [0, 1]^{|\mathcal{R}|}$ with [20,21])

$$\forall\, i \in I_{\mathcal{R}} : \lim_{k \to \infty} \boldsymbol{e}_i^T\, T^k(\beta) = \boldsymbol{q}^T(\beta), \tag{11}$$

where $T(\beta)$ denotes the transition matrix whose components are given by eq. (2) and $\boldsymbol{e}_i \in [0, 1]^{|\mathcal{R}|}$ a unit vector whose components are given by $e_{ij} = \delta_{ij}$ (δ denotes the Kronecker-delta).

Initially the system is in the configuration i. The corresponding relative frequency distribution is given by $\boldsymbol{e}_i$, i.e. configuration i occurs with probability one whereas all other configurations occur with probability zero. Next the transition mechanism is applied k times and as $k \to \infty$ the probability to meet the system in a given final configuration becomes independent from the initial configuration, i.e. stationarity is achieved.

The component $q_i(\beta)$ denotes the relative frequency of state i for the stationary distribution of the system at a given value for the cooling control parameter β. In other words, $q_i(\beta)$ denotes the probability that the system occurs in state i when stationarity is achieved.

The equilibrium vector $\boldsymbol{q}(\beta)$ is the left eigenvector of $T(\beta)$ with eigenvalue 1. Its components are given by

$$q_i(\beta) = q_o(\beta)\, B_{i_o i}(\beta), \tag{12}$$

where i denotes the state label of a configuration corresponding to the optimum value of the cost function and $q_o(\beta)$ the normalization constant given by

$$q_o^{-1}(\beta) = \sum_{i=1}^{|\mathscr{R}|} B_{i_o i}(\beta). \tag{13}$$

Note that $\sum_{i=1}^{|\mathscr{R}|} q_i(\beta) = 1$.

The existence of an equilibrium vector $\boldsymbol{q}(\beta)$ can be easily proven by showing that the $q_i(\beta)$ given by eqs (12) and (13), satisfy the detailed balance equations[20])

$$\forall\, i, j \in I_{\mathscr{R}} : q_i(\beta)\, T_{ij}(\beta) = q_j(\beta)\, T_{ji}(\beta). \tag{14}$$

Indeed, this follows from repeated application of conditions (i), (ii) and (iv) given by eqs (7), (8) and (10), respectively.

From eqs (9a) and (9b) it follows that

$$\lim_{\beta \to \infty} q_i(\beta) = |\mathscr{R}|^{-1} \tag{15}$$

and

$$\lim_{\beta \downarrow 0} q_i(\beta) = \begin{cases} 0 & \text{if } i \neq i_o \in I_{\mathscr{R}_o} \\ |\mathscr{R}_o|^{-1} & \text{if } i = i_o \in I_{\mathscr{R}_o}, \end{cases} \tag{16}$$

with

$$\mathscr{R}_o = \{\boldsymbol{r}_i \in \mathscr{R} \,|\, C(\boldsymbol{r}_i) = C(\boldsymbol{r}_{i_o})\} \tag{17}$$

denoting the set of optimal configurations.

For large values of the cooling control parameter ($\beta \to \infty$) all states occur with equal probability, whereas for $\beta \downarrow 0$ the system will occur in an optimal configuration with a probability equal to one.

We now elaborate in more detail on the conditions on the function $B_{ij}(\beta)$ as given by eqs (7)-(10). When we have two pairs of configurations (i_1, j_1) and (i_2, j_2) with $C(\boldsymbol{r}_{i_1}) = C(\boldsymbol{r}_{i_2})$ and $C(\boldsymbol{r}_{j_1}) = C(\boldsymbol{r}_{j_2})$ it follows from repeated application of conditions (ii) (eq. (8)) and (iv) (eq. (10)) that $B_{i_1 j_1}(\beta) = B_{i_2 j_2}(\beta)$. Hence, for a fixed value of β the function $B_{ij}(\beta)$ depends only on $C(\boldsymbol{r}_i)$ and $C(\boldsymbol{r}_j)$. Factorization of the function $B_{ij}(\beta)$ and recalling that $\forall\, i, j \in I_{\mathscr{R}}$: $B_{ij}(\beta) \in (0, 1)$ (condition (iii)) results into

$$\forall\, \Delta C_{ij} > 0 : B_{ij}(\beta) = \exp(-g(C(\boldsymbol{r}_i), C(\boldsymbol{r}_j))\, \alpha(\beta)) \tag{18}$$

with $g: \mathbb{R}^2 \to \mathbb{R}^+$ a two-variable function and $\alpha: \mathbb{R}^+ \to \mathbb{R}^+$ a function of one real variable. Condition (iv) can be rewritten in terms of the function g as

$$\forall\, i, j, k \in I_{\mathcal{R}} \qquad \text{with } C(r_k) \geqslant C(r_j) \geqslant C(r_i):$$
$$g(C(r_i), C(r_k)) = g(C(r_i), C(r_j)) + g(C(r_j), C(r_k)). \tag{19}$$

Assuming that the function g satisfies the condition

$$\forall\, x, y, z \in [a, b] \subset \mathbb{R} \qquad \text{with } x \geqslant y \geqslant z:$$
$$g(x, z) = g(x, y) + g(y, z), \tag{20}$$

it can be shown that any (smooth) function g satisfying this condition takes the form

$$g(x, y) = h(y) - h(x) \quad \forall\, x, y \in [a, b]. \tag{21}$$

Indeed we may take $h(x) = g(a, x)$ $(\forall\, x \in [a, b])$.

The condition of eq. (20) is stronger than the condition of eq. (19) when a and b are chosen such that $\forall\, i \in I_{\mathcal{R}}: a \leqslant C(r_i) \leqslant b$. Nevertheless, since the configuration space is usually very large and the $C(r_i)$ may be assumed to be reasonably uniformly distributed over the interval $[a, b]$, the condition of eq. (20) does not seem to be too restrictive. Consequently we have

$$\forall\, i, j \in I_{\mathcal{R}} \qquad \text{and} \qquad \Delta C_{ij} > 0:$$
$$g(C(r_i), C(r_j)) = h(C(r_j)) - h(C(r_i)), \tag{22}$$

with $h: \mathbb{R} \to \mathbb{R}^+$ a monotonically increasing function. There are many functions that satisfy the condition of eq. (22). Here we choose $h(C(r_i)) = C(r_i)$.

For the function $\alpha(\beta)$ we have

$$\lim_{\beta \to \infty} \alpha(\beta) = 0 \tag{23}$$

and

$$\lim_{\beta \downarrow 0} \alpha(\beta) = \infty. \tag{24}$$

There are many functions that satisfy these conditions. A simple one is given by

$$\alpha(\beta) = \beta^{-1}. \tag{25}$$

With eqs (8), (22) and (25) the final expression for the acceptation criterion becomes

$$B_{ij}(\beta) = \begin{cases} \exp\left(-\dfrac{\Delta C_{ij}}{\beta}\right) & \text{if } \Delta C_{ij} > 0 \\ 1 & \text{if } \Delta C_{ij} \leqslant 0 \end{cases} \tag{26}$$

and consequently the relative frequencies for the stationary distribution are given by

$$q_i(\beta) = q_o(\beta) \exp\left(\frac{-\Delta C_{i_o i}}{\beta}\right) \tag{27}$$

with

$$q_o^{-1}(\beta) = \sum_{i=1}^{|\mathscr{R}|} \exp\left(\frac{-\Delta C_{i_o i}}{\beta}\right). \tag{28}$$

The expression given by eq. (26) is the standard acceptation criterion for statistical cooling and is equivalent to the Metropolis criterion[22]). It was first introduced in combinatorial optimization by Kirkpatrick et al.[7]) (and independently by Cerny[13])) and they used it on the basis of the analogy between computer simulations in condensed matter physics as first described by Metropolis et al.[22]) and the optimization of a multi-variable system. In this analogy ΔC_{ij} corresponds with the energy difference between two many-particle states of the physical system and β corresponds with the product kT, where T denotes the physical temperature of the system and k the Boltzmann constant. The stationary distribution of eq. (27) is known as the Boltzmann distribution[23]).

As we shall see in subsec. 4.4 the special form of the stationary distribution of eq. (27) enables the calculation of additional functions from which information may be deduced on the convergence of the statistical cooling algorithm.

From eq. (26) it is evident that also deteriorations of the system are accepted ($\Delta C_{ij} > 0$). However, if these deteriorations are large or the value of β is small, the probability that they will be accepted is small.

In practice one obtains a Markov chain by repeatedly generating a new configuration from an old one and applying for each new configuration the acceptance criterion of eq. (26) for a fixed value of β. Optimization is performed by starting this chain-generation process for a given value of β and repeating it for decreasing values of β until β approaches 0. Acceptance of configurations for which $\Delta C_{ij} > 0$ is determined by means of a random number generator, i.e. the value of $\exp(-\Delta C_{ij}/\beta)$ is compared with a random number between 0 and 1; if this number is smaller than the value of the exponential function the new configuration is accepted, otherwise, it is rejected.

It should be noted that carrying out the optimization process with $\beta = 0$ reduces the statistical cooling algorithm to a simple iterative improvement algorithm since the acceptation criterion of eq. (26), then, only allows improvements of the system ($\Delta C_{ij} \leqslant 0$), whereas all configurations corresponding with system deteriorations ($\Delta C_{ij} > 0$) are rejected.

4. Convergence of the algorithm

Given the formal description of the previous section one is left with the problem to determine the values of the parameters that govern the convergence of the statistical cooling algorithm, i.e. to establish a cooling schedule. The parameters determining such a schedule are

(i) the length of the Markov chains,
(ii) the start value of the cooling control parameter,
(iii) the decrease of the cooling control parameter and
(iv) a stop criterion to terminate the algorithm.

In the subsequent subsections we discuss some formal aspects related to the convergence of the statistical cooling algorithm. From these results values of the parameters of the cooling schedule can be determined that lead to proper convergence of the statistical cooling algorithm towards near-optimal results (see also sec. 5). Most of these values, however, cause the statistical cooling algorithm to require large computation efforts (some values even lead to exponential-time execution). By introducing some approximations to the formal convergence description, other values for the cooling schedule parameters can be derived leading to polynomial-time execution of the algorithm. Such a cooling schedule is discussed in sec. 6.

4.1. Length of the Markov chains

The equilibrium behaviour of Markov processes is described formally by the theory of finite matrices and Markov chains. A substantial review of this theory is given by Seneta [21]). Here we present some important results which can be deduced from this theory. In the course of this we follow Seneta's book.

For a given Markov chain with transition matrix $T(\beta)$ and length M, stationarity is achieved if $\boldsymbol{e}_i^T T^M(\beta)$ is close to the equilibrium vector $\boldsymbol{q}^T(\beta)$. Combining the Perron-Frobenius theorem (Seneta [21]) theorem 1.1) with Seneta [21]) theorem 1.2 and recalling that T is primitive it can be straightforwardly shown that for $M \to \infty$

$$\forall\, i \in I_{\mathcal{R}}: \left\| \boldsymbol{e}_i^T T^M(\beta) - \boldsymbol{q}^T(\beta) \right\| = O(M^s |\lambda_2(\beta)|^M), \tag{29}$$

where $\lambda_2(\beta)$ $(0 < |\lambda_2(\beta)| < 1)$ denotes the second largest eigenvalue of $T(\beta)$ with multiplicity m_2 and $s = m_2 - 1$. From eq. (29) it is apparent that convergence towards a stationary distribution is completely determined by the second largest eigenvalue of the transition matrix. Determination of this eigenvalue as a function of the cooling control parameter β is difficult and very time consuming. For the statistical cooling algorithm this procedure is, therefore, considered to be impractical since fast convergence is an important requirement.

Next, by using Seneta[21]), theorem 4.16, it follows that as $M \to \infty$

$$\forall\, i \in I_{\mathscr{R}}\colon \| e_i^T\, T^M(\beta) - q^T(\beta) \| < 2(1 - \gamma^t(\beta))^{M/t-1}, \tag{30}$$

with t the number of distinct incidence matrices corresponding to $T(\beta)$, and $\gamma(\beta) \in (0, \min_{i,j}{}^{+} T_{ij}^M(\beta)) \subset \mathbb{R}$. According to Isaacson and Madsen[24]) one may take $t = |\mathscr{R}|^2 - 3|\mathscr{R}| + 3$ and imposing $2(1 - \gamma^t(\beta))^{M/t-1} < \varepsilon$ then results in the following condition for stationarity

$$M \gtrsim K(|\mathscr{R}|^2 - 3|\mathscr{R}| + 3), \tag{31}$$

with $K = \ln\varepsilon / \ln(1 - \gamma^t(\beta))$ and $\varepsilon \in \mathbb{R}^+$ some small positive number. For many combinatorial optimization problems $|\mathscr{R}|$ is exponential in the number of system variables. Using chain lengths according to eq. (31), therefore, results in an exponential-time algorithm. In secs 5 and 6 the chain length is re-examined and there we present a cooling schedule that uses chain lengths resulting in polynomial-time execution of the algorithm.

4.2. *Start value of the cooling control parameter*

The start value β_o of the cooling control parameter has to be large enough to ensure that initially all configurations occur with equal probabilities (see eq. (15)). This requirement is necessary (but not sufficient) to prevent the optimization process from getting stuck in a local minimum. If β_o is taken too low with respect to positive ΔC_{ij} values that correspond with transitions that allow the system to escape from the local minimum, these transitions will probably not be accepted and consequently the optimization process gets trapped in the local minimum. An adequate heuristic expression for β_o is given by

$$\beta_o = \varrho \max_{i,j \in I_{\mathscr{R}}} \Delta C_{ij}, \tag{32}$$

where $\varrho \gg 1$ (e.g. $\varrho = 10$). This start value clearly guarantees access to all possible states with sufficiently high probability.

4.3. *Decrement of the cooling control parameter*

The decrement of the cooling control parameter determines the behaviour of the optimization process in the limit $\beta \downarrow 0$. So far the literature, to our knowledge, does not provide a basic formalism from which unique values for the decrement can be determined. Experimental observations reveal that the decrement of the cooling control parameter is not very critical as long as the Markov chains are long enough to allow the system to arrive at a distribution that is close to the stationary distribution. Fast convergence may be achieved by requiring an exponential decrease of the probability for accepting a transi-

tion with a given value of ΔC_{ij} ($\Delta C_{ij} > 0$). This results in a stepwise decrease of the cooling control parameter by multiplying it each step by a constant factor. Experimental values for this factor lie in the range from 0.80 to 0.99. These values are in agreement with the literature[7,11,12,17]).

If the length of the Markov chains is reduced the decrease of the cooling control parameter becomes more critical. For these cases the literature provides some spare results. Lundy and Mees[25]) present results for inhomogeneous Markov chains ($M = 1$). Experimentally it turns out that in this case the convergence ($\beta \downarrow 0$) is much too slow and that no good optimization results are obtained within reasonable amounts of CPU-time. Otten and van Ginneken[26]) come up with an expression for the decrement which is purely based on experimental observations. This restricts its validity to the class of problems they have been investigating. In sec. 6 we shall derive an expression for the decrease of the cooling control parameter which yields good optimization results and is generally applicable.

4.4. *Termination of the algorithm*

Quantitative information on the progress of the optimization process can be obtained from the entropy which is a natural measure of the order of the system. High entropy values correspond with chaos; low values correspond with order. A formal definition of the entropy S is given by the expression[23])

$$S(\beta) = -\sum_{i=1}^{|\mathcal{R}|} q_i(\beta) \ln q_i(\beta). \tag{33}$$

By using the explicit form of the $q_i(\beta)$ of eq. (27) the following properties can be derived

$$S_\infty = \lim_{\beta \to \infty} S(\beta) = \ln |\mathcal{R}| \tag{34}$$

and

$$S_0 = \lim_{\beta \downarrow 0} S(\beta) = \ln |\mathcal{R}_o|. \tag{35}$$

We, furthermore, introduce the expectations of the cost function, $\langle C(\beta) \rangle$ and of its square, $\langle C^2(\beta) \rangle$, defined as

$$\langle C(\beta) \rangle = \sum_{i=1}^{|\mathcal{R}|} q_i(\beta)\, C(r_i) \tag{36}$$

and

$$\langle C^2(\beta) \rangle = \sum_{i=1}^{|\mathcal{R}|} q_i(\beta)\, C^2(r_i). \tag{37}$$

The variance $\sigma^2(\beta)$ then may be written as

$$\sigma^2(\beta) = \langle C^2(\beta)\rangle - \langle C(\beta)\rangle^2. \tag{38}$$

With these additional functions some useful relations can be derived, e.g.

$$\frac{\mathrm{d}\langle C(\beta)\rangle}{\mathrm{d}\beta} = \beta \frac{\mathrm{d}S(\beta)}{\mathrm{d}\beta} \tag{39}$$

and

$$\frac{\mathrm{d}S(\beta)}{\mathrm{d}\beta} = \frac{\sigma^2(\beta)}{\beta^3}, \tag{40}$$

and consequently the entropy is given by

$$S(\beta) = S(\beta_1) - \int_{\beta}^{\beta_1} \frac{\sigma^2(\beta')}{\beta'^3}\,\mathrm{d}\beta'. \tag{41}$$

For $\beta_1 \gg 0$ the entropy can be approximated by the expression

$$S(\beta_1) = \ln|\mathscr{R}| - \frac{\sigma^2(\beta_1)}{2\beta_1^2}. \tag{42}$$

Indeed, eq. (42) can be derived from eq. (33) by writing the exponential form of eq. (27) as a power series up to the second power and rearranging the various terms.

It should be noted that the expressions of eqs (34)-(42) result from the special form of the relative frequencies for the stationary distribution given by eq. (27). Similar expressions are obtained in the theory of statistical mechanics for the expectations of the energy and its square, the spread of the energy and the physical entropy of a many-particle system in thermal equilibrium [23]).

Comparing the current value of the entropy $S(\beta)$ and the value of the entropy for the optimum configuration S_o during the optimization process yields information on the amount by which the current system configurations deviate from the optimum configuration. This information is contained in the quantity

$$\frac{S(\beta) - S_o}{S_\infty - S_o}. \tag{43}$$

To calculate this quantity the expectations $\langle C(\beta)\rangle$ and $\langle C^2(\beta)\rangle$ have to be estimated (see eqs (36) and (37)). Here, we need two additional functions defined as the mean of the cost function $\bar{C}(\beta)$ and its square $\overline{C^2}(\beta)$, averaged over n samples of a random test, i.e.

$$\bar{C}(\beta) = \sum_{i=1}^{n} \frac{C(r_i)}{n} \tag{44}$$

and

$$\overline{C^2}(\beta) = \sum_{i=1}^{n} \frac{C^2(r_i)}{n} . \tag{45}$$

Note that $\bar{C}(\beta)$ and $\overline{C^2}(\beta)$ depend on the cooling control parameter β. This dependency stems from the fact that the calculation of the averages is restricted to the states belonging to the Markov chain generated for that particular value of β.

If the functions $C(r_i)$ and $C^2(r_i)$ suitably behave the central limit theorem for Markov chains [27]) states that for sufficiently large values of n the quantities $(\langle C(\beta)\rangle - \bar{C}(\beta))$ and $(\langle C^2(\beta)\rangle - \overline{C^2}(\beta))$ are normally distributed with mean 0 and variances proportional to $\mathrm{E}\{(\langle C(\beta)\rangle - \bar{C}(\beta))^2\}$ and $\mathrm{E}\{(\langle C^2(\beta)\rangle - \overline{C^2}(\beta))^2\}$, respectively, where $\mathrm{E}\{\ldots\}$ denotes the expectation for the quantity in the curly braces for the stochastic process in question. Thus, if n becomes large the expectations $\langle C(\beta)\rangle$ and $\langle C^2(\beta)\rangle$ may be approximated by the average values $\bar{C}(\beta)$ and $\overline{C^2}(\beta)$ and with this the entropy $S(\beta)$ can be calculated numerically from eq. (41). In practice the number of samples is taken to be equal to the length of the Markov chains.

With the results presented in this section it is possible to determine values of the parameters that determine the cooling schedule of the statistical cooling algorithm. The next section presents results on the application of the algorithm to a special class of traveling salesman problems. The parameters are discussed on the basis of these results.

5. Examples

The traveling salesman problem [28,29]) is presumably the best known combinatorial optimization problem and many problems in the field of computer science and VLSI design are closely related to it. In the traveling salesman problem it is required to construct the shortest tour of a prescribed list of N cities, where each city is called at exactly once. In our type of problem the tour is taken to be closed, i.e. begin and end of the tour coincide. The state vector for a given configuration is given by $r_i = (r_{i_1}, r_{i_2}, \ldots, r_{i_{N+1}})$, where $r_{i_{N+1}} = r_{i_1}$ and $r_{i_j} \in [1, N] \subset \mathbb{N}$ is the city number corresponding with the j-th sequence number in the i-th tour. The cost function is defined as the average distance between two cities and is given by

$$C(r_i) = \sum_{l=1}^{N} \frac{d(r_{i_l}, r_{i_{l+1}})}{N} , \tag{46}$$

where $d(r_{i_l}, r_{i_{l+1}})$ denotes the distance between two subsequent cities with sequence numbers l and $l+1$.

System rearrangements are carried out by randomly selecting two sequence numbers $(n, m : n > m)$ and reversing the order in which the cities in between these two sequence numbers are traversed, i.e.

$$r_{j_l} = r_{i_l} \qquad \text{if } 1 \leqslant l \leqslant m \quad \text{or} \quad n \leqslant l \leqslant N, \tag{47a}$$

$$r_{j_l} = r_{i_{n+m-l}} \qquad \text{if } m+1 \leqslant l \leqslant n-1. \tag{47b}$$

The difference in the cost function is calculated 'incrementally', i.e. ΔC_{ij} is calculated from the differences of the local rearrangement, by using the expression

$$\Delta C_{ij} = \frac{1}{N}\{d(r_{i_m}, r_{i_{n-1}}) + d(r_{i_{m+1}}, r_{i_n}) - d(r_{i_m}, r_{i_{m+1}}) - d(r_{i_{n-1}}, r_{i_n})\}, \tag{48a}$$

$$= \frac{1}{N}\{d(r_{j_m}, r_{j_{m+1}}) + d(r_{j_{n-1}}, r_{j_n}) - d(r_{j_m}, r_{j_{n-1}}) - d(r_{j_{m+1}}, r_{j_n})\}. \tag{48b}$$

In the present paper we use a special class of problems in which the cities are positioned on the vertices of a square grid such that all vertices are occupied.

Here, we take for the $d(r_{i_l}, r_{i_{l+1}})$ the Manhattan distance defined as

$$d(r_{i_l}, r_{i_{l+1}}) = |x_{i_{l+1}} - x_{i_l}| + |y_{i_{l+1}} - y_{i_l}|, \tag{49}$$

with (x_k, y_k) the coordinates of the k-th city in a two-dimensional Euclidian space. Using Manhattan distances speeds up the distance computations (compared to Euclidean distances).

For the type of problems introduced above optimal configurations can be easily determined by hand and the corresponding value of the cost function is simply given by the characteristic grid size G in the case of a grid with an even by even number of cities and by $(N - 1 + \sqrt{2})\,G/N$ for a grid with an odd by odd number of cities. We assume that $G = 1$. The fact that the optimum value of the cost function is known beforehand is a very powerful aid in the analysis of the statistical cooling algorithm.

The statistical cooling algorithm for these examples is implemented in PASCAL on a VAX 11/780 computer system.

Fig. 1 shows the configuration of the system in the different stages of the optimization process. The optimization process is carried out with a chain length $M = 20\,000$, start value of the cooling control parameter $\beta_o = 1.7$ and a decrease in the cooling control parameter given by a constant factor $\xi = 0.95$ (see subsec. 4.3). The values of M and β_o are calculated according to eqs (31) and (32). The initial configuration (fig. 1*a*) corresponds with a tour in which the 100 cities are traversed in a random sequence. The deviation from the op-

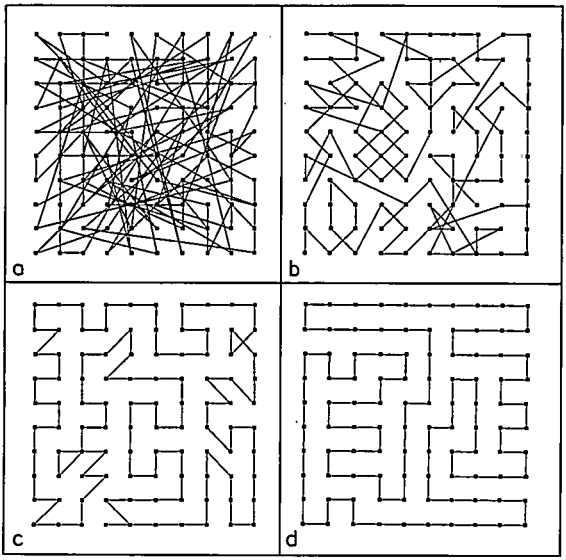

Fig. 1. Evolution of the system configuration of a 100-city traveling salesman problem while applying the statistical cooling algorithm. The configurations are obtained after generation of 0, 68, 92 and 123 chains, respectively. The initial configuration corresponds with a tour that is given by a random sequence of the 100 cities and consequently the tour looks very chaotic (*a*) (high entropy values). During the optimization process the average distance between two subsequent cities in the tour is minimized and the tour becomes less chaotic, (*b*) and (*c*) (the entropy decreases). Finally the average distance between two subsequent cities in the tour becomes minimal and an optimal configuration is reached. The tour shows a highly regular pattern (*d*) (the entropy becomes minimal).

timum is very large: 367% (here the deviation of a given configuration i from the optimum is defined as $(\Delta C_{i_o i}/C(r_{i_o})) \times 100\%$. As the optimization proceeds the deviation from the optimum becomes smaller: 81.5% after 68 chains (fig. 1*b*) and 7.8% after 92 chains (fig. 1*c*), and finally an optimal configuration is found; 0% deviation after 123 chains (fig. 1*d*).

Fig. 2 shows the minimum chain length required to arrive at an optimal configuration as a function of the number of cities, together with the minimum chain length required to arrive at a near-optimal configuration which deviates less than 2% from an optimal configuration. These minimum chain lengths are determined with a greedy algorithm. This algorithm starts off with a given small chain length M and carries out the optimization process until a configuration is obtained that has not changed over 10 subsequent chains (we then say it is a local minimum). If this configuration does not meet the criterion specified for the final configuration (0% or 2% deviation, respectively) the entire optimization process is repeated with a chain length M' given by $M' = 1.2 \times M$

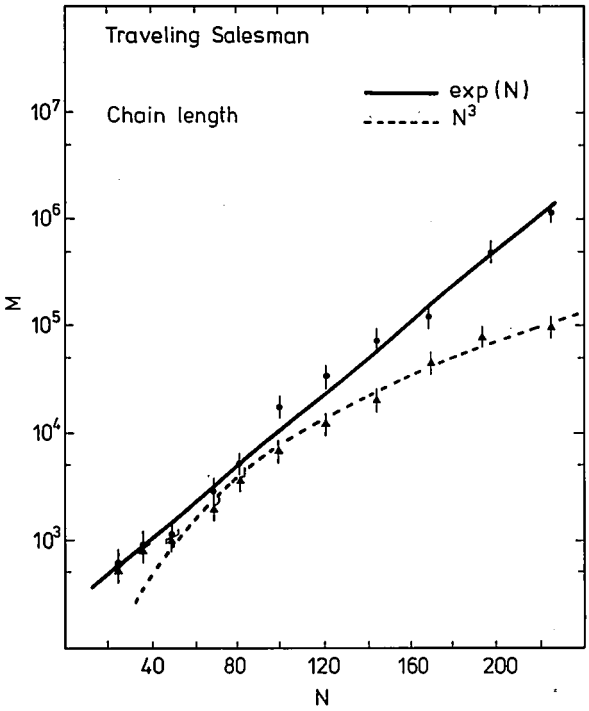

Fig. 2. Minimum chain length M required to reach an optimal configuration (•) and to reach a final configuration deviating less than 2% from the optimum (▲) given as a function of the number of cities N in the traveling salesman problem ($\delta = 0.1$). The curves correspond with an exponential function (solid line) and a cubic function (dashed line), respectively (see also fig. 3).

(the error bars in figs 2 and 3 are deduced from the uncertainty resulting from this factor). This process then is repeated until the criterion is met.

The corresponding minimum CPU-times are shown in fig. 3. The curves in figs 2 and 3 correspond with an exponential function (solid curves) and a cubic function (dashed lines), respectively. The exponential behaviour of the chain length and the corresponding CPU-time for the optimization process for which the final result was required to be an optimum is in agreement with the result of eq. (31) by which an exponential-time behaviour was predicted and the fact that the traveling salesman problem as such is NP-hard. The observation that near-optimal results can be obtained (2% deviation from the optimum) in cubic time indicates that the algorithm is very powerful. Similar results are obtained with traditional algorithms when applying more complex rearrangements as was shown by Kernighan and Lin [28,29]). The CPU time for these algorithms, however, becomes prohibitive if the number of cities becomes large (> 300).

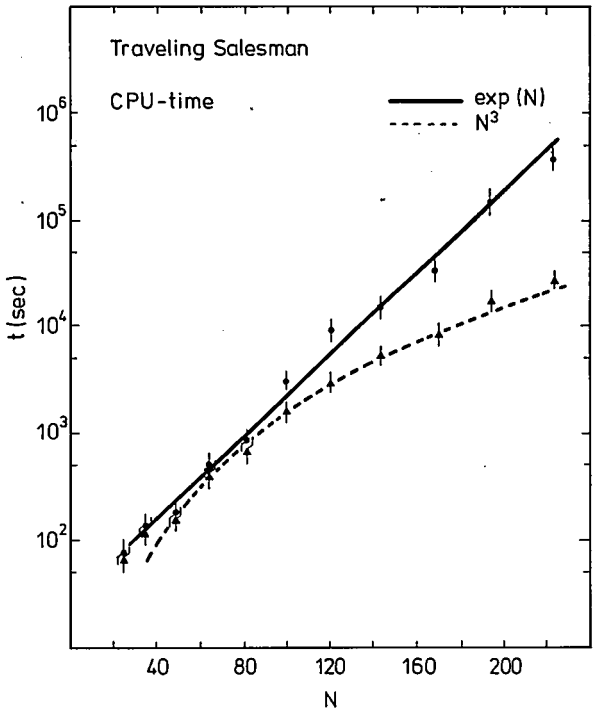

Fig. 3. Minimum CPU time t required to reach an optimal configuration (•) and to reach a final configuration deviating less than 2% from the optimum (▲) given as a function of the number of cities N in the traveling salesman problem ($\varepsilon = 0.001$). The curves correspond with an exponential function (solid line) and a cubic function (dashed line), respectively (see also fig. 2).

Determination of the minimum chain length required for obtaining specified final optimization results by the greedy algorithm described above is very time consuming and impractical. The most elegant and practical implementation of the statistical cooling algorithm is given by a polynomial-time algorithm that obtains near-optimal results with a fixed cooling schedule. The algorithm then has to determine the parameters of the cooling schedule by monitoring the evolution of the system that is to be optimized. In sec. 6 such a cooling schedule is presented.

Results of calculations of the entropy according to eq. (41) (not presented here) show that only for extremely large random tests (long Markov chains) the value for the $S(\beta)$ can be calculated sufficiently accurate, i.e. $S(\beta) \rightarrow S_o$ as $\beta \downarrow 0$, see also eq. (35) (for a 100-city traveling salesman problem a chainlength $M \simeq 100\,000$ was needed). For smaller random tests the calculated values for $S(\beta)$ as $\beta \downarrow 0$ show large fluctuations around S_o. These fluctuations are due to

the fact that the integral of eq. (41) is evaluated numerically and since β appears in the denominator of the integrand as a third power, the value of $\sigma^2(\beta)$, which is approximated by $\overline{C^2}(\beta) - \bar{C}(\beta)^2$ (see eqs (39), (44) and (45)), has to be determined very accurately as $\beta \downarrow 0$. Consequently the random test and thus the chain length has to be large if the entropy is to be calculated accurately (central limit theorem, sec. 4).

Calculations of the entropy from the derivative of the expectations of the cost function according to

$$S(\beta) = S(\beta_1) - \int_{\beta}^{\beta_1} \frac{\mathrm{d}\langle C(\beta')\rangle}{\beta' \mathrm{d}\beta'} \mathrm{d}\beta' \tag{50}$$

yield similar problems with respect to the accuracy.

From this it may be concluded that the stop criterion based on eq. (43) is solely reliable in those cases where the chain length is very large. In their work Otten and van Ginneken[26]) present a stop criterion that is similar to the one given in eq. (43) (they actually evaluate $S(\beta)$ according to eq. (50)). Since they use a small chain length (equal to the number of system elements) it is surprising, on the basis of the discussion presented above, that their stop criterion provides a reliable mains for termination of the algorithm.

Near-optimal results may be obtained by the statistical cooling algorithm using considerably smaller chain lengths than required for accurate calculation of the stop criterion of eq. (43). This stop criterion, therefore, results in needless consumption of CPU time before it can be reliably decided to terminate the algorithm. This is illustrated by the 100-city traveling salesman problem shown in fig. 1, for which reliable values for the entropy could only be determined for chain lengths $M \gtrsim 100\,000$, whereas optimal results are already obtained for $M = 20\,000$.

In subsec. 6.3 a more practical stop criterion is discussed.

6. Cooling schedule and control

In this section we present a new cooling schedule for the statistical cooling algorithm. The emphasis of the schedule is on feasibility and generality, i.e. the parameters of the cooling schedule are determined in such a way that (i) fast convergence of the algorithm is obtained towards near-optimal results, and (ii) the quality of the optimization results is not dependent on the size of the problem.

6.1. Start value of the cooling control parameter

For many combinatorial optimization problems it is difficult to determine the maximum difference in the cost function for any two internal configura-

tions of the system. Consequently it is hard to estimate the start value of the cooling control parameter β_o according to eq. (32). As pointed out in subsec. 4.2 the value of β_o should be high enough to have a sufficiently large probability for reaching all possible system configurations. This corresponds with an acceptation ratio χ that is close to 1. Here χ is defined as the ratio between the number of system rearrangements accepted by the algorithm and the total number of rearrangements that is generated for a given value of β.

The value of β_o can be obtained by monitoring the evolution of the system during a number of rearrangements before the actual optimization process starts, and adjusting the value of the cooling control parameter in such a way that a constant value of the acceptation ratio is maintained. The value of β_o is then given by the final value of β obtained by updating $\beta\, m_o$ times according to the expression

$$\beta = \overline{\Delta C}^{(-)} \left(\ln \frac{m_2}{m_2 \chi - (1 - \chi)\, m_1} \right)^{-1}, \tag{51}$$

with m_1 and m_2 the numbers of rearrangements with $\Delta C_{ij} \leqslant 0$ and $\Delta C_{ij} > 0$ ($m_1 + m_2 = m_o$), respectively, and $\overline{\Delta C}^{(-)}$ the average value of those ΔC_{ij} values for which $\Delta C_{ij} > 0$. During these m_o system rearrangements the acceptance criterion of eq. (26) is applied.

Determination of the start value of the cooling control parameter in the way described above can only be done reliably in those cases where the values of the cost function for the different system configurations are sufficiently uniformly distributed. If the set of system configurations corresponds with values of the cost function distributed over a number of distinct intervals whose mutual distances are large compared to their size, then determination of the start value of the cooling control parameter according to eq. (51) results in most cases in a β_o value which is too small and, therefore, will cause the optimization process to get stuck in a local optimum. In this case one is forced to determine β_o according to eq. (32).

6.2. *Chain length and cooling control decrease*

The experimental observations presented in sec. 5 showed that optimal configurations can only be obtained with the statistical cooling algorithm in exponential time. The experimental results also showed that a considerable gain in CPU time is achieved for the optimization process by allowing the final result to deviate from the optimum only by a small amount: a deviation of the final result by 2% from the optimum reduces the CPU effort from exponential time to cubic time. The major problem, then, is to determine a chain length and decrease in cooling control parameter for which near-optimal results are ob-

tained whose deviation from the optimum is independent of the size of the problem. In this section we derive a formalism that results into values of the parameters that meet these requirements.

The starting point of the derivation is given by the assumption that for each Markov chain generated during the optimization process the distribution of the relative frequencies of the states should be close to the stationary distribution (given by eqs (27) and (28)). In that case we shall say that the process is in quasi equilibrium.

This can be achieved by starting the optimization process at a high value of the cooling control parameter (for high values of β quasi equilibrium is quickly achieved, see eq. (15)). Next, the decrement of β is carried out in such a way that the stationary distributions for two succeeding values of β are close to each other. In this way, after decreasing β, the distribution of the relative frequencies of the various states will rapidly approach the new stationary distribution, so that the lengths of the successive chains can be kept small.

To achieve quasi equilibrium, we therefore impose

$$\| q(\beta) - q(\beta') \| < \varepsilon, \tag{52}$$

which is equivalent to imposing

$$\forall\, i \in I_{\mathcal{R}} : \frac{1}{1+\delta} < \frac{q_i(\beta)}{q_i(\beta')} < 1 + \delta \tag{53}$$

with ε, $\delta \in \mathbb{R}^+$ small positive numbers.

Theorem

If the following condition is satisfied (for $\beta' < \beta$)

$$\forall\, i \in I_{\mathcal{R}} : \frac{\exp\left(\frac{-\Delta C_{i_o i}}{\beta}\right)}{\exp\left(\frac{-\Delta C_{i_o i}}{\beta'}\right)} < 1 + \delta \tag{54}$$

then the condition of eq. (53) is satisfied.

Proof

From eq. (54) we deduce, using $\beta' < \beta$

$$\sum_{i=1}^{|\mathcal{R}|} \exp\left(\frac{-\Delta C_{i_o i}}{\beta'}\right) < \sum_{i=1}^{|\mathcal{R}|} \exp\left(\frac{-\Delta C_{i_o i}}{\beta}\right) < (1+\delta) \sum_{i=1}^{|\mathcal{R}|} \exp\left(\frac{-\Delta C_{i_o i}}{\beta'}\right), \tag{55}$$

whence

$$q_o(\beta') > q_o(\beta) > \frac{1}{1+\delta}\, q_o(\beta'). \tag{56}$$

From eqs (54) and (56) it follows that

$$\frac{q_i(\beta)}{q_i(\beta')} = \frac{q_o(\beta)\exp\left(\frac{-\Delta C_{i_o i}}{\beta}\right)}{q_o(\beta')\exp\left(\frac{-\Delta C_{i_o i}}{\beta'}\right)} < 1+\delta \tag{57a}$$

and

$$\frac{q_i(\beta)}{q_i(\beta')} > \frac{1}{1+\delta}. \tag{57b}$$

■

To satisfy eq. (53), we therefore now impose eq. (54), which is equivalent to

$$\forall\, i \in I_{\mathcal{R}} : \beta' > \frac{\beta}{1+\frac{\ln(1+\delta)\beta}{\Delta C_{i_o i}}}. \tag{58}$$

Making a slight simplification in neglecting the states that will most probably not occur in a Markov chain for that particular value of β we obtain

$$\forall\, i \in I_{\mathcal{R}_\beta} : \beta' > \frac{\beta}{1+\frac{\ln(1+\delta)\beta}{\Delta C_{i_o i}}}, \tag{59}$$

where $I_{\mathcal{R}_\beta}$ denotes the set of states belonging to the Markov chain at β.

If we assume that the values of the cost function are normally distributed for a given value of β, then the $\Delta C_{i_o i}$ are normally distributed with mean $\mu(\beta) = \bar{C}(\beta) - C(r_{i_o})$ and variance $\sigma^2(\beta)$, where the average $\bar{C}(\beta)$ and the variance $\sigma^2(\beta)$ are given by eqs (43) and (39) respectively and $C(r_{i_o})$ is the optimum value of the cost function. Hence

$$\Pr\{\Delta C_{i_o i} \leqslant \mu(\beta) + 3\sigma(\beta)\} \simeq 0.99. \tag{60}$$

We therefore replace the condition of eq. (59) by the following condition (which is, according to eq. (60), a stronger condition for 99% of the $\Delta C_{i_o i}$)

$$\beta' > \frac{\beta}{1+\frac{\ln(1+\delta)\beta}{\mu(\beta)+3\sigma(\beta)}}, \tag{61}$$

from which we obtain the following expression for β'

$$\beta' = \beta\left\{1+\frac{\ln(1+\delta)\beta}{\mu(\beta)+3\sigma(\beta)}\right\}^{-1}. \tag{62}$$

For most optimization problems, however, $C(r_{i_o})$ (and hence $\mu(\beta)$) is not known and, therefore, we replace $\mu(\beta) + 3\sigma(\beta)$ by $3\sigma(\beta)$ to obtain the following final expression for the decrement

$$\beta' = \beta \left\{ 1 + \frac{\ln(1+\delta)\beta}{3\sigma(\beta)} \right\}^{-1}. \tag{63}$$

The rather crude approximation of $\mu(\beta) + 3\sigma(\beta)$ results in larger steps in β. In first order this effect can be taken care of by the choice of δ (see also sec. 7).

If quasi equilibrium is maintained during the optimization process the chain length can be kept small. However, it is apparent that for larger problems re-establishment of stationarity will take longer than for smaller problems. The system, furthermore, should have the possibility to investigate all configurations belonging to the configuration sub-space of a given configuration. A good value for the chain length, therefore, may be given by the maximum of the size of the configuration sub-spaces $\mathcal{R}_i$, i.e. (see eq. (4))

$$M = \max_{i \in I_{\mathcal{R}}} |\mathcal{R}_i| . \tag{64}$$

6.3. *Termination of the algorithm*

In most practical cases the chain length given by eq. (64) is too small to calculate the entropy of the system sufficiently accurate as $\beta \downarrow 0$ (subsec. 4.4). The stop criterion based on eq. (43), therefore, cannot be used in our case.

Here we use a stop criterion that is based on the decrease during the optimization process of the running average of the cost function. Fig. 4 shows a typical example of the behaviour of the average value of the cost function, calculated according to eq. (44), as a function of the cooling control parameter. The dashed curve is obtained by smoothing the experimental curve. This is done by calculating a running average over a number of Markov chains. From the dashed curve it is possible to estimate correctly the average difference in cost function $\overline{\Delta C}(\beta) = \overline{C}_s(\beta) - C(r_{i_o})$ (with $\overline{C}_s(\beta)$ the smoothed value of $\bar{C}(\beta)$) as β becomes small. If $\beta \ll 1$ we have

$$\overline{\Delta C}(\beta) \simeq \beta \frac{\mathrm{d}\overline{C}_s(\beta)}{\mathrm{d}\beta} . \tag{65}$$

It is apparent that the strongly fluctuating values of $\bar{C}(\beta)$ do not allow proper determination of the derivative $\mathrm{d}\bar{C}(\beta)/\mathrm{d}\beta$. Therefore, the smoothed values are used.

If the difference $\overline{\Delta C}(\beta)$ is small compared to the value of $\bar{C}(\beta_o)$ the optimization process is close to a near optimum. The algorithm is then terminated if

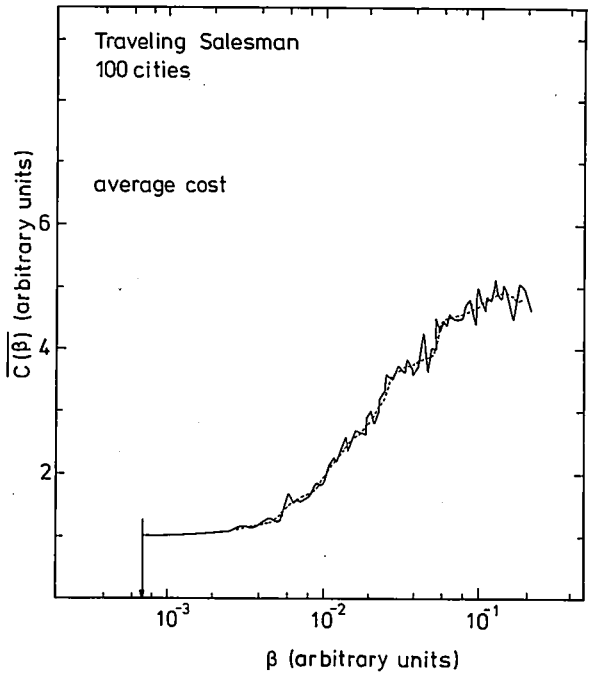

Fig. 4. Average value of the cost function, $\bar{C}(\beta)$, as a function of the cooling control parameter β for a 100-city traveling salesman problem. The dashed curve is obtained by smoothing the experimental data (see text).

the following stop criterion Z holds

$$Z: \frac{d\overline{C}_s(\beta)}{d\beta} \frac{\beta}{\bar{C}(\beta_o)} < \varepsilon, \tag{66}$$

with $\varepsilon \in \mathbb{R}$ a small positive number.

With the cooling schedule presented above the execution time of the statistical cooling algorithm is proportional to

$$\max_{i \in I_{\mathcal{R}}} |\mathcal{R}_i| \cdot \ln|\mathcal{R}|, \tag{67}$$

where the term $\max_{i \in I_{\mathcal{R}}} |\mathcal{R}_i|$ originates from the length of the Markov chains (eq. 64)) and the term $\ln|\mathcal{R}|$ is an upperbound for the number of steps in the cooling control parameter (see appendix). For most combinatorial optimization problems the maximum of the size of the configuration sub-spaces is polynomial in the system complexity. Consequently our statistical cooling algorithm is of polynomial time.

7. Examples revisited

The present section shows experimental results on the analysis of the statistical cooling algorithm governed by the cooling schedule described in the previous section. Here we use again the special class of traveling salesman problems introduced in sec. 5. For these problems the length of the Markov chains is according to eq. (64) given by $M = N(N-1)/2$ with N the number of cities.

Fig. 5 shows the deviation from the optimum during the optimization process as a function of the stop criterion of eq. (66) for a 225-city traveling salesman problem ($\delta = 0.1$). From the figure it is apparent that the stop criterion allows sufficiently accurate selection of near-optimal configurations.

Fig. 6 shows the deviation from the optimum for the final optimization result as a function of the distance parameter δ of eq. (63) for a 225-city traveling salesman problem ($\varepsilon = 0.001$). The figure indicates that the final result of the optimization process can be accurately guided by the value of δ.

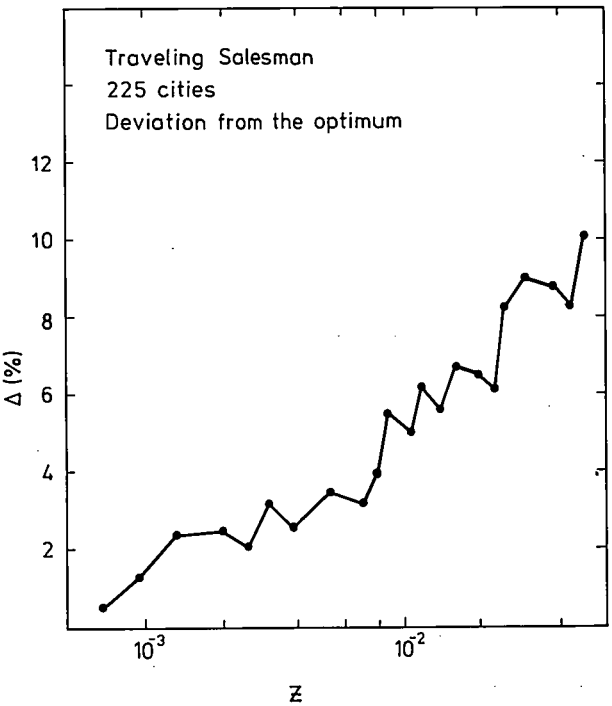

Fig. 5. Deviation, Δ, from the optimum during the optimization process for a 225-city traveling salesman problem as a function of the stop criterion Z (see text). The line is drawn to guide the eye.

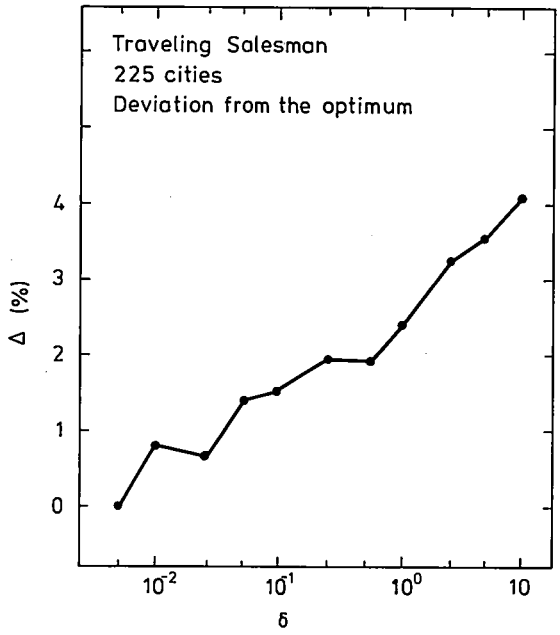

Fig. 6. Deviation, Δ, from the optimum of the final configuration for a 225-city traveling salesman problem as a function of the distance parameter δ (see text). The line is drawn to guide the eye.

Small δ-values (< 0.005) yield optimal results. As the value of δ increases the steps between the subsequent values of the cooling control parameter become larger (the distance between the equilibrium vectors becomes larger) and consequently the deviation from the optimum of the final result increases.

Fig. 7 shows a comparison of the deviations from the optimum of the final results of various optimization processes as a function of the number of cities in the traveling salesman problem. The experimental results indicated with (*b*), (*c*) and (*d*) are obtained with the statistical cooling algorithm, whereas result (*a*) is obtained with an iterative improvement algorithm (sec. 2). For the iterative improvement algorithm we used the statistical cooling algorithm with $\beta = 0$ (see sec. 3). The figure shows that iterative improvement yields results that deviate from the optimum by a considerable amount (7%) (fig. 7*a*) and that the results of the statistical cooling algorithm are much better (fig. 7*b-d*).

The deviation from the optimum of the final results of the statistical cooling algorithm applying the cooling schedule of the previous section is more or less constant over the observed range of cities (fig. 7*c*). The horizontal line repre-

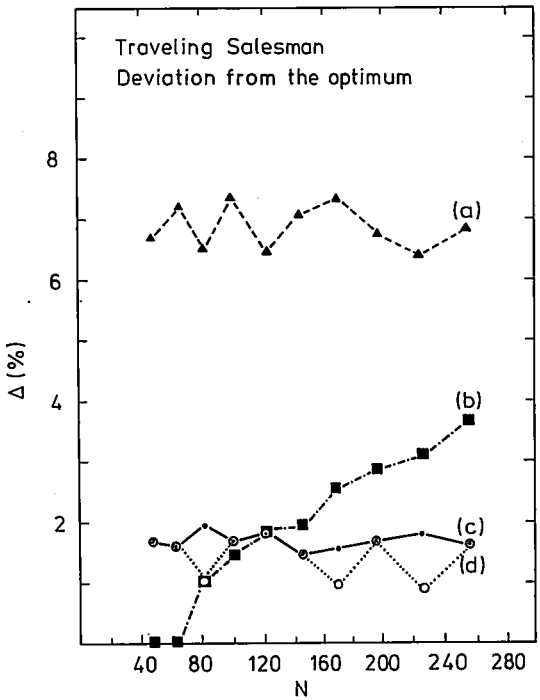

Fig. 7. Deviation, Δ, from the optimum of the final result as a function of the number of cities, N, in the traveling salesman problem. Results are shown for iterative improvement (▲, (*a*)) and for statistical cooling with chain lengths given by $M = 50N$ (■, (*b*)) and $M = N(N-1)/2$ (● o, (*c*) and (*d*)) where (●, (*c*)) corresponds with the deviation of the final result and (o, (*d*)) with the smallest deviation ever reached during the optimization process. The horizontal line represents the average value of the results of (*c*). The lines between the data points are drawn to guide the eye.

sents the average deviation of the final results. This indicates that the deviation of the final result has become independent of the size of the optimization problem. This is an important feature which is completely due to the choice of the parameters in the cooling schedule. If the cooling schedule is changed this problem-size independency may vanish. For instance if the chain length is replaced by $M = 50N$ one observes that the deviation from the optimum of the final result increases as the problem size increases (fig. 7*b*).

An interesting feature of the statistical cooling algorithm is given by the observation that in some cases there is an intermediate result with a lower value of the cost function than the final optimization result (fig. 7*d*). The extent to which this effect takes place in the statistical cooling algorithm has not been studied yet. It provides, however, an interesting subject and should be investigated in more detail.

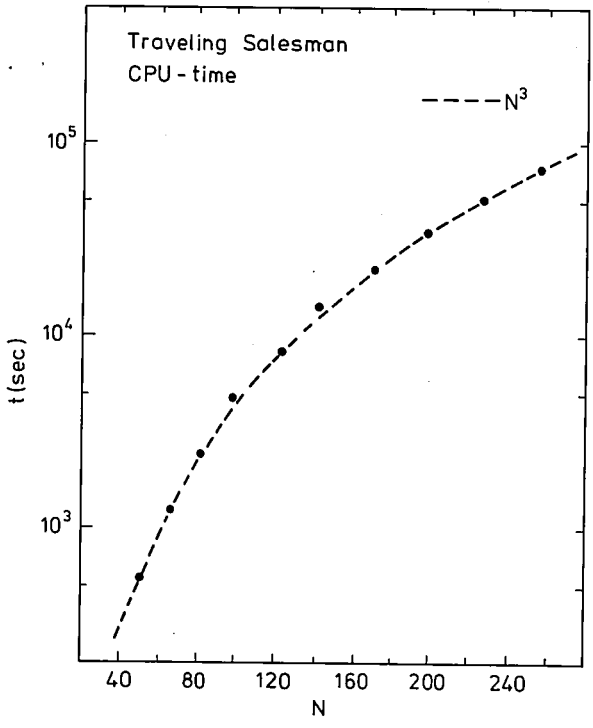

Fig. 8. Total CPU time t for the statistical cooling algorithm employing the cooling schedule presented in sec. 6 as a function of the number of cities, N, in the traveling salesman problem. The curve represents a cubic function in N.

Fig. 8 shows the CPU time required by the statistical cooling algorithm governed by the cooling schedule of sec. 6 as a function of the number of cities in the traveling salesman problem. The dashed curve represents a function that is proportional to a third power of the number of cities and the figure shows that this curve fits the data points very well. From the appendix we obtain that the order of the computation effort for this problem has a worst-case upperbound given by $N^3 \ln N$, the chain length contributing a factor N^2 and the configuration space a factor $\ln N^N$. The time behaviour observed here is similar to the one observed in fig. 3 for the case where the final result was allowed to deviate 2% from the optimum (note that the cooling schedules used in both cases are completely different). This clearly indicates that the algorithm governed by the cooling schedule of sec. 6 meets the requirements formulated in that section, i.e. the values of the parameters of the cooling schedule

are determined by the algorithm during the optimization process in such a way that near-optimal results are obtained in polynomial time.

The polynomial time behaviour of the statistical cooling algorithm is predicted by the results of the appendix. The observation, however, that the decrement in the cooling control parameter contributes only linearly to the total CPU effort (the chain length contributes a factor N^2) indicates that the present cooling schedule is very powerful.

From fig. 8 it is apparent that large problems require large computation times. Reduction of the computation time may be achieved by equipping the statistical cooling algorithm with more powerful system rearrangements, which are, as pointed out before, in most cases strongly problem dependent, or dividing up the problem in a number of smaller less time consuming problems. Bonomi and Lutton obtained in this way near-optimal results for a 10 000-city traveling salesman problem [12]). Here, one should take into account that no results have been published (to our knowledge) on solutions to traveling salesman problems with more than ~300 cities obtained with traditional algorithms.

There are, however, problems for which the computation effort cannot be reduced easily. The macro-placement problem [10,14,18,26]) provides an optimization problem for which the calculation of the difference in cost function for a given system rearrangement is a time consuming manipulation which inherently causes the optimization process to require large computation times. Furthermore, $|\mathscr{R}_i|$ is very large.

8. General application of the statistical cooling algorithm

Statistical cooling is applicable to many problems in combinatorial optimization and the corresponding algorithm is easy to implement. Application of the algorithm can be divided into three distinct stages:

(i) formulation of a concise description of the system,
(ii) formulation of the system rearrangements,
(iii) carrying out the optimization along the lines of a cooling scheme.

Next, we elaborate on these items in more detail.

ad (i) A concise description of the system should contain a state vector representation of the system and an expression for the cost function in terms of the state vector. The cost function has to be defined in such a way that it represents appropriately the quantity that is to be optimized. Both the state vector and the cost function should be given by simple expressions that are easy to manipulate.

ad (ii) Transformation of one system configuration to another involves four steps: firstly, the system has to be perturbed, secondly, the difference

in cost function has to be calculated, thirdly, decisions have to be made whether or not a new configuration is accepted and finally the system has to be updated if the new configuration is accepted. Perturbation of the system is usually done by generating at random new configurations applying only relatively simple rearrangements (swoppings, inversions, translations, rotations). Calculation of the difference in cost function has to be very easy since in practice this is the most time consuming part of the algorithm. The decision for accepting a new configuration is made by applying the criteria of eq. (26). Updating the system again should be a fast manipulation.

ad (iii) Carrying out the optimization process along the lines of a cooling schedule is equivalent to generating the stationary distribution of eq. (27) and taking the limit $\beta \downarrow 0$ of the distribution (eq. (16)). The optimization parameters that determine the schedule are the length of the Markov chains, the initial value and the decrement of the cooling control parameter and the stop criterion. Proper values for these parameters, ensuring fast convergence towards near-optimal results are presented in sec. 6.

The complete algorithm can be schematically depicted in pseudo PASCAL as

```
PROCEDURE STATISTICAL COOLING
begin
    INITIALIZE(M, β_o, C̄(β_o));
    β := β_o;
    repeat
        for i := 1 to M do
        begin
            PERTURB(state i → state j, ΔC_ij);
            if ΔC_ij ⩽ 0 then accept else
                if exp(−ΔC_ij/β) > random[0, 1) then accept;
            if accept then UPDATE(state j);
        end;
        CALCULATE (σ(β), dC̄_s(β)/dβ);
        β = β/(1 + ln(1 + δ) β/3σ(β));
    until dC̄_s(β)/dβ · β/C̄(β_o) < ε
end.
```

Here the function of the various procedures (written in upper case) is indicated by their names.

We consider it feasible to implement the statistical cooling algorithm as some kind of library routine, such that only proper expressions for the state representation and cost function are required as an input from the user, followed by a completely automized optimization. This, however, requires additional investigations which are presently carried out at the Philips Research Laboratories.

9. Summary and conclusions

In this paper we presented a general theoretical framework for the description of the statistical cooling algorithm. The algorithm is formulated as a Markov process and its convergence is analysed within this scope. A cooling schedule is presented by which near-optimal results may be obtained for *NP*-hard combinatorial optimization problems in polynomial time. The performance of the statistical cooling algorithm is analysed by means of a special class of traveling salesman problems. The algorithm turns out to be a powerful tool for the optimization of the cost functions involved in this type of problems. Final results are obtained in cubic time deviating less than 2% from the optimum.

The advantages of the statistical cooling algorithm can be formulated as follows:

(i) the algorithm is generally applicable,
(ii) the algorithm is easy to implement,
(iii) the final results can be as good as desired and
(iv) the *quality* of the final results is problem-size independent.

The major disadvantage is

(i) some applications may require large computation efforts.

Concludingly, the algorithm may serve as a powerful tool in the field of combinatorial optimization.

The analysis of the statistical cooling algorithm presented in the present paper does not deal with the question whether or not statistical cooling is fundamentally different from an algorithm based on repeated iterative improvement. For instance, carrying out iterative improvement 300 times in succession, starting at different initial configurations and keeping up the best result for all runs, yields for a 100-city traveling salesman problem a final result that deviates 2.2% from the optimum. This result was obtained with the statistical cooling algorithm using $\beta = 0$ in a total CPU time of $5 \cdot 10^3$ sec. Both the final result and the computation effort are about equal to the ones obtained by applying the statistical cooling algorithm to the same problem. This is an intri-

guing result which raises the question whether statistical cooling is just a form of guided iterative improvement or a technique that is fundamentally different from iterative improvement.

Acknowledgements

The authors are greatly indebted to J. W. Walkier for his technical assistance during the implementation of the various algorithms and to drs. G. F. M. Beenker, H. M. J. Boots and A. J. E. M. Janssen for many useful discussions.

Appendix

Theorem

Let the decrement of the cooling control parameter leading from β_i to β_{i+1} be given by (eq. (63))

$$\beta_{i+1} = \frac{\beta_i}{1 + \alpha(\beta_i)\,\beta_i} \tag{A1}$$

with $\alpha(\beta_i) = \ln(1 + \delta)/3\sigma(\beta_i)$ and let the decrement be terminated at a final value of the cooling control parameter β_n for which the stop criterion of eq. (66) is satisfied, then the total number of steps in the cooling control parameter is bounded by $\propto \ln|\mathscr{R}|$.

Proof

To prove this theorem we proceed in two stages, (i) and (ii). Firstly, the total number of steps is expressed in terms of β_n. Secondly, β_n is expressed in terms of the system complexity $|\mathscr{R}|$. The final proof is then given by combining the intermediate results.

(i) Eq. (A1) can be written as

$$\gamma_{i+1} = \frac{\gamma_i}{1 + \delta(\beta_i)\,\gamma_i} \tag{A2}$$

with

$$\gamma_i = \frac{\beta_i}{\beta_o}, \quad \delta(\beta_i) = \alpha(\beta_i)\,\beta_o, \qquad i = 0, 1; \ldots . \tag{A3}$$

Let

$$\alpha = \min_i \alpha(\beta_i), \quad \delta = \alpha\,\beta_o. \tag{A4}$$

By using $\gamma_o = 1$ we can prove by induction that $\gamma_i \leqslant \varepsilon_i$ $(i = 0, 1, \ldots)$, where ε_i is generated by the recursive formula

$$\varepsilon_{i+1} = \frac{\varepsilon_i}{1 + \delta\,\varepsilon_i}, \quad \varepsilon_o = 1. \tag{A5}$$

This recursion is solved by

$$\varepsilon_i = \frac{1}{1 + i\delta}, \qquad i = 0, 1, \ldots . \tag{A6}$$

Hence

$$\gamma_i \leqslant \frac{1}{1 + i\delta}, \qquad i = 0, 1, \ldots \tag{A7}$$

and

$$\beta_i \leqslant \frac{\beta_o}{1 + i\alpha\beta_o}, \qquad i = 0, 1, \ldots \tag{A8}$$

from which we obtain for the total number of steps

$$n \leqslant \frac{\beta_o - \beta_n}{\alpha\beta_o\beta_n} < \frac{1}{\alpha\beta_n} . \tag{A9}$$

(ii) The final value of the cooling control parameter β_n is determined by the stop criterion of eq. (60), i.e.

$$\exists\, \varepsilon' \in (0, \varepsilon] \subset \mathbb{R} : \varepsilon' = \frac{\beta_n}{\bar{C}(\beta_o)} \left(\frac{\mathrm{d}\bar{C}_s(\beta)}{\mathrm{d}\beta} \right)_{\beta=\beta_n} . \tag{A10}$$

Replacing $\mathrm{d}\bar{C}_s(\beta)/\mathrm{d}\beta$ by $\mathrm{d}\langle C(\beta)\rangle/\mathrm{d}\beta$ and using the relation of eq. (39) we obtain

$$\begin{aligned} \varepsilon' &= \frac{\beta_n}{\bar{C}(\beta_o)} \left(\frac{\mathrm{d}\langle C(\beta)\rangle}{\mathrm{d}\beta} \right)_{\beta=\beta_n} \\ &= \frac{\beta_n^2}{\bar{C}(\beta_o)} \left(\frac{\mathrm{d}S(\beta)}{\mathrm{d}\beta} \right)_{\beta=\beta_n} . \end{aligned} \tag{A11}$$

Hence, for $\beta_n \ll 1$ and $\mathrm{d}^2S(\beta)/\mathrm{d}\beta^2 < 0$ eq. (A11) becomes

$$\varepsilon' \leqslant \beta_n \frac{S(\beta_n)}{\bar{C}(\beta_o)} < \beta_n \frac{S_\infty}{\bar{C}(\beta_o)} . \tag{A12}$$

Using eq. (34) one obtains

$$\beta_n > \frac{\bar{C}(\beta_o)\,\varepsilon'}{S_\infty} = \frac{\bar{C}(\beta_o)\,\varepsilon'}{\ln|\mathscr{R}|} . \tag{A13}$$

Combining eqs (A9) and (A13) then results in

$$n < \frac{1}{\alpha\varepsilon'\bar{C}(\beta_o)} \ln|\mathscr{R}| . \tag{A14}$$

Thus, the total number of steps in the cooling control parameter is bounded by $\eta \ln|\mathscr{R}|$, where $\eta = (\alpha\varepsilon'\bar{C}(\beta_o))^{-1}$.

REFERENCES

[1]) C. H. Papadimitriou and K. Steiglitz, Combinatorial optimization: Algorithms and complexity, Prentice Hall, New Jersey, 1982, and references therein.
[2]) R. M. Garey and J. D. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W. H. Freeman and Company, San Francisco, 1979.
[3]) A. V. Aho, J. E. Hopcroft and J. D. Ullman, The design and analysis of computer algorithms, Addison Wesley, Reading, 1974.
[4]) M. Breuer, Ed., Design automation of digital systems, Prentice Hall, New York, 1972.
[5]) R. Karp, Math. Oper. Res. **2**, 209 (1977).
[6]) S. Lin, Networks **5**, 33 (1975).
[7]) S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, Science **220**, 671 (1983).
[8]) List of Abstracts from the Workshop on Statistical Physics in Engineering and Biology, IBM Research Center, Yorktown Heights, April 1984.
[9]) M. P. Vecchi and S. Kirkpatrick, IEEE Trans. CAD **2**, 215 (1983).
[10]) D. W. Jepsen and C. D. Gelatt Jr., Proc. IEEE Int. Conf. on Computer Design, Port Chester, New York, 1983; p. 495.
[11]) J. Becker, A. J. Nices and J. A. Storer, Proc. Int. Workshop on Parallel Computing and VLSI, May 23-25, Amalfi, Italy, 1984, Elsevier's Science Publ., Amsterdam, 1985.
[12]) E. Bonomi and J.-L. Lutton, SIAM Rev. **26**, 551 (1984).
[13]) V. Cerny, J. Opt. Th. and Appl. **45**, 41 (1985).
[14]) F. Romeo, A. Sangiovanni Vincentelli and C. Sechen, Proc. IEEE Int. Conf. on Computer Design, Port Chester, New York, 1984; p. 652.
[15]) S. R. White, Proc. IEEE Int. Conf. on Computer Design, Port Chester, New York, 1984; p. 646.
[16]) C. Rowen and J. L. Hennesy, to appear in Proc. 22nd Design Automation Conference, Las Vegas, 1985.
[17]) E. H. L. Aarts, P. J. M. van Laarhoven, R. Burgess and B. Culleton, Philips Research preprint MS 13.166.
[18]) E. H. L. Aarts and P. J. M. van Laarhoven, Philips Research preprint MS 13.219.
[19]) W. Feller, An introduction to probability theory and applications, vols. 1 and 2, Wiley, New York, 1950 and 1966, respectively.
[20]) F. P. Kelly, Reversibility and stochastic networks, Wiley, Chichester, 1979.
[21]) E. Seneta, Non negative matrices and Markov chains, Springer Verlag, New York, 2nd edition, 1981.
[22]) M. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[23]) M. Toda, R. Kubo and N. Saito, Statistical physics, Springer Series in Solid-State Sciences, Springer, Berlin, 1983.
[24]) D. Isaacson and R. Madsen, J. Appl. Prob. **11**, 829 (1974).
[25]) M. Lundy and A. Mees, submitted for publication in Math. Programming.
[26]) R. H. J. M. Otten and L. P. P. P. van Ginneken, Proc. IEEE Int. Conf. Comp. Aided Des., Santa Clara, 1984, p. 96.
[27]) J. L. Doob, Stochastic processes, Wiley, New York, 1953; p. 221.
[28]) S. Lin, Bell Syst. Tech. J. **44**, 2245 (1965).
[29]) S. Lin and B. W. Kernighan, Oper. Res. **21**, 498 (1973).