Local Search in Combinatorial Optimization
Edited by E. Aarts and J. K. Lenstra
© 1997 John Wiley & Sons Ltd.

4

Simulated annealing

Emile H. L. Aarts, Jan H. M. Korst
Philips Research Laboratories, Eindhoven

Peter J. M. van Laarhoven
Eindhoven University of Technology

1 INTRODUCTION

Simulated annealing belongs to a class of local search algorithms that are known as threshold algorithms. These algorithms play a special role within local search for two reasons. First, they appear to be quite successful when applied to a broad range of practical problems, which has given them quite a reputation among practitioners. Second, some threshold algorithms such as simulated annealing have a stochastic component, which facilitates a theoretical analysis of their asymptotic convergence, and this has made them very popular to mathematicians.

The emphasis of our presentation is on the mathematics of threshold algorithms, with special attention paid to simulated annealing. We discard the application of threshold algorithms to specific problems, since this issue is discussed in detail in Chapters 8 to 13 of this volume. However, we mention some general aspects of the algorithms' practical use to give the reader a feeling of what he might expect from their application.

2 THRESHOLD ALGORITHMS

Let $(\mathcal{S}, f)$ be an instance of a combinatorial minimization problem with solution set $\mathcal{S}$ and cost function $f: \mathcal{S} \rightarrow \mathbb{R}$. Furthermore, let $\mathcal{N}: \mathcal{S} \rightarrow \mathcal{P}(\mathcal{S})$ be a neighborhood function, which defines for each $i \in \mathcal{S}$ a set $\mathcal{N}(i) \subseteq \mathcal{S}$ of neighboring solutions. The question is to find an optimal solution $i^* \in \mathcal{S}$ that minimizes the cost of all solutions over $\mathcal{S}$. Consider the class of *threshold algorithms* given by the pseudocode of Figure 4.1.

The procedure INITIALIZE selects a start solution from $\mathcal{S}$, the procedure GENERATE selects a solution from the neighborhood of a current solution, and the procedure STOP evaluates a stop criterion that determines termination of the algorithm. Threshold algorithms continually select a neighbor of a current solution and compare the difference in cost between these solutions to a threshold. If the cost difference is below the threshold, the neighbor replaces the current solution. Otherwise, the search continues with the current solution. The sequence $(t_k | k = 0, 1, 2, \ldots)$ denotes the thresholds, where t_k is used at iteration k of the local search algorithm.

We distinguish between the following three types of threshold algorithms depending on the nature of the threshold:

- *Iterative improvement*: $t_k = 0$, $k = 0, 1, 2, \ldots$ Clearly, this is a variant of the classical greedy local search in which only cost-reducing neighbors are accepted.
- *Threshold accepting*: $t_k = c_k$, $k = 0, 1, 2, \ldots,$ where $c_k \geqslant 0, c_k \geqslant c_{k+1}$, and $\lim_{k \rightarrow \infty} c_k = 0$. Threshold accepting uses a nonincreasing sequence of deterministic thresholds. Due to the use of positive thresholds, neighboring solutions with larger costs are accepted in a limited way. In the course of the algorithm's execution, the threshold values are gradually lowered, eventually to 0, in which case only improvements are accepted.

```
procedure THRESHOLD_ALGORITHM;
begin
    INITIALIZE ($i_{start}$);
    $i := i_{start}$;
    $k := 0$;
    repeat
        GENERATE ($j$ from $\mathcal{N}(i)$);
        if $f(j) - f(i) < t_k$ then $i := j$;
        $k := k + 1$;
    until STOP;
end;
```

Figure 4.1 Pseudocode of a class of threshold algorithms

- *Simulated annealing*: t_k = a random variable with expected value $\mathbb{E}(t_k) = c_k \in \mathbb{R}^+$, $k = 0, 1, 2, \ldots$. The t_k's follow a probability distribution function F_{c_k} over $\mathbb{R}^+$. Simulated annealing uses randomized thresholds with values between zero and infinity, and the probability of a threshold t_k being at most $y \in \mathbb{R}^+$ is given by $\mathbb{P}_{c_k}\{t_k \leqslant y\} = F_{c_k}(y)$. This implies that each neighboring solution can be chosen with a positive probability to replace the current solution. In practice the function F_{c_k} is chosen such that solutions corresponding to large increases in cost have a small probability of being accepted, whereas solutions corresponding to small increases in cost have a larger probability of being accepted.

As an important matter we remark that, in the original simulated annealing version, Kirkpatrick, Gelatt & Vecchi [1983] and Černý [1995] take for F_{c_k} the negative exponential distribution with parameter $1/c_k$. This choice is identical to the following *acceptance criterion*. For any two solutions $i, j \in \mathcal{S}$ the probability of accepting j from i at the kth iteration is given by

$$\mathbb{P}_{c_k}\{\text{accept } j\} = \begin{cases} 1 & \text{if } f(j) \leqslant f(i), \\ \exp\left(\dfrac{f(i) - f(j)}{c_k}\right) & \text{if } f(j) > f(i). \end{cases} \tag{1}$$

The parameter c_k is used in the simulated annealing algorithm as a *control parameter*, and it plays an important role in the convergence analysis of the algorithm. We will drop the subscript k of the control parameter if it is not explicitly needed.

Some preliminary convergence results

For iterative improvement it is not possible to give nontrivial convergence results. For *multistart iterative improment*, which consists of single runs of iterative improvement that are repeated with different start solutions, it is easily verified that an optimal solution is found with probability 1 if an infinite number of restarts is allowed.

Threshold accepting was introduced by Dueck & Scheuer [1990] as a deterministic version of simulated annealing. One of the major unresolved problems is the determination of appropriate values for the thresholds. Furthermore, as in the case of iterative improvement, no general convergence results can be proved, but one can do slightly better. Althöfer & Koschnick [1991] have related some convergence properties of threshold accepting to those of simulated annealing. The proofs of the convergence results are not constructive. They make use of the fact that, in some sense, simulated annealing generalizes threshold accepting. For instance, one of their propositions states that if there is a finite sequence of thresholds for simulated annealing, leading to an optimal solution with probability $1 - \varepsilon$ for some $\varepsilon > 0$, then there also exists a finite sequence of thresholds for threshold accepting, leading to an optimal solution with probability $1 - \varepsilon$. Furthermore, they give a simple example for which suboptimal solutions can be reached with threshold sequences of any length, demonstrating that even asymptotically the algorithm can get trapped in local minima.

Finally, for simulated annealing there exist general convergence results, which state that under certain mild conditions an optimal solution is found with probability 1. These results are obtained by analyzing the algorithm in terms of Markov chains. This central issue is the subject of Section 5 and subsequent sections. Before we present these quantitative performance results, we first discuss some qualitative results.

3 A QUALITATIVE PERFORMANCE ANALYSIS

To analyze the performance of the threshold algorithms introduced in the previous section, we consider the following combinatorial optimization problem, which is a simplified version of a problem introduced by Lundy & Mees [1986]. Let the set of solutions be given by $\mathscr{S} = \{0, 1, \ldots, N\}$, with $1 \ll N$, and let the cost function $f: \mathscr{S} \to \mathbb{R}$ be given by

$$f(i) = i - \lfloor i/n \rfloor \delta,$$

with $n \in \mathbb{N}$, $1 \ll n \ll N$, $\delta \in \mathbb{R}$, and $1 < \delta < n$. Figure 4.2 illustrates the cost function for $\delta = 2$. The problem is to find an element in $\mathscr{S}$ with minimum cost.

This problem formulation can be generalized to solution sets of d dimensions as follows. If $\mathscr{S} = \{0, 1, \ldots, N\}^d$, the cost function is given by $f\{i_0, i_1, \ldots, i_{d-1}\} =$

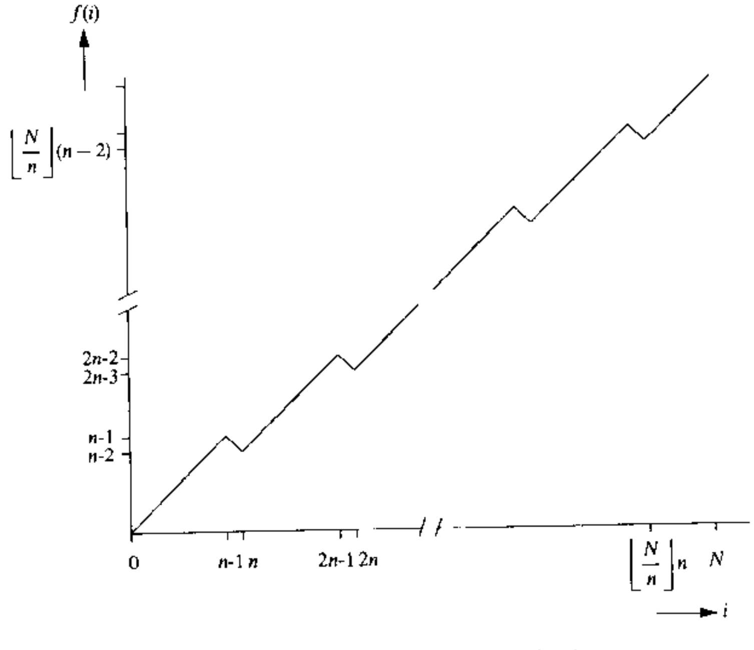

Figure 4.2 The cost function f for $\delta = 2$

$r - \lfloor r/n \rfloor \delta$, with $r = \max\{i_0, i_1, \ldots, i_{d-1}\}$. Lundy & Mees [1986] discuss a two-dimensional version of this problem.

Furthermore, let $\mathcal{N}$ be a neighborhood function defined by $\mathcal{N}(i) = \{i-1, i, i+1\}$, for $i \in \{1, 2, \ldots, N-1\}$, $\mathcal{N}(0) = \{0, 1\}$, and $\mathcal{N}(N) = \{N-1, N\}$. We now consider the performance of the threshold algorithms on this problem, for two choices of δ, namely $\delta \approx 1$ and $\delta \gg 1$, corresponding to the situations where a small or a large deterioration is required to escape from a local minimum, respectively.

The behavior of multistart iterative improvement is identical for both cases. A single iteration is described as follows. It randomly selects an initial solution i from $\mathcal{S} = \{0, 1, \ldots, N\}$ and terminates in a local optimum given by $i' = \lfloor i/n \rfloor n$. Clearly, the algorithm only finds a global optimum in a given iteration if the initial solution is in $\{0, 1, \ldots, n-1\}$. The expected number of iterations to obtain a global optimum can be shown to be $O(M)$, with $M = N/n$, for the one-dimensional case, and $O(M^d)$ for the case with d dimensions. This illustrates that multistart iterative improvement can always find a global optimum, but that the required effort may be very large – equivalent to enumerating all solutions in $\mathcal{S}$.

We now consider the behavior of threshold accepting and simulated annealing for the case that $\delta \approx 1$. For threshold accepting, we consider three situations, depending on the value of the thresholds t_k. If $t_k \geqslant 1$, each transition from a solution i to one of its neighbors $i-1$, $i+1$ is accepted, and the algorithm performs a random walk in the solution space $\mathcal{S}$. If $\delta - 1 \leqslant t_k < 1$, the algorithm only accepts transitions from a solution i to its neighbor $i-1$. Hence, in that case the algorithm performs as an optimization algorithm. If $t_k < \delta - 1$, the algorithm performs as iterative improvement and terminates in the local minimum i' that corresponds to the current solution i. From this example we observe that threshold accepting may perform very well, provided that the thresholds t_k are properly chosen. If $\delta - 1 \leqslant t_k < 1$, the algorithm finds a global optimum in $O(N)$ transitions or, assuming that n is constant, in $O(M)$ transitions. Note that this number of transitions does not depend on the number of dimensions. Consequently, provided the threshold is chosen properly, threshold accepting may do considerably better than multistart iterative improvement.

For the case that $\delta \approx 1$, simulated annealing behaves similarly to threshold accepting. This can be seen as follows. If $c \geqslant 1$, simulated annealing more or less performs a random walk, but having a slight preference for solutions with a small cost. If $\delta - 1 \leqslant c \ll 1$, the algorithm nearly only accepts transitions from a solution i to its neighbor $i-1$. In that case the algorithm also performs as an optimization algorithm, requiring $O(N)$ transitions to obtain a global optimum. If $t_k < \delta$, the algorithm probably terminates in the local minimum i' that corresponds to the current solution i, although there is a small probability of obtaining better solutions.

Next we consider the case that $\delta \gg 1$. In that case, threshold accepting cannot find on average better solutions than (single-start) iterative improvement. This can be shown as follows. If $t_k \geqslant \delta - 1$, the algorithm accepts any proposed transition. Consequently, as long as $t_k \geqslant \delta - 1$, threshold accepting performs a random walk on the set of solutions. If $t_k < \delta - 1$, the algorithm behaves at least as bad as (single-start) iterative improvement. Let i be the current solution at that moment. Then the best obtainable solution is $i' = \lfloor i/n \rfloor i$. The algorithm even has a positive probability of terminating in a local minimum j' with $j' > i'$. Since, on average, the initial solution is identical to the solution obtained after the random walk, we conclude that, on average, threshold accepting cannot find better solutions than (single-start) iterative improvement. Hence, even if the algorithm is given enough time and the thresholds are carefully chosen, threshold accepting cannot guarantee to find a global optimum, whereas simulated annealing always has a positive probability of reaching a global optimum, when given enough time. As we show in the next section, simulated annealing asymptotically finds a global optimum with probability 1.

Summarizing, we have the following conclusions.

- For some optimization problems, the expected number of transitions necessary for reaching a global optimum is much smaller for simulated annealing and threshold accepting than for multistart iterative improvement.
- For some optimization problems, the expected cost of a final solution obtained by threshold accepting is not better than the expected cost of a final solution obtained by (single-start) iterative improvement.

Hence, the interest in simulated annealing can be motivated by the fact that, compared with multistart iterative improvement and threshold accepting, its performance is less dependent on the specific topology of the 'cost function landscape'.

4 THE PHYSICS ANALOGY

The origin of simulated annealing and the choice of the acceptance criterion lie in the physical annealing process [Kirkpatrick, Gelatt & Vecchi, 1983; Černý, 1985]. In *condensed matter physics, annealing* is a thermal process for obtaining low-energy states of a solid in a *heat bath*. It consists of the following two steps: (1) the temperature of the heat bath is increased to a maximum value at which the solid melts; (2) the temperature is carefully decreased until the particles of the melted solid arrange themselves in the ground state of the solid. In the liquid phase all particles of the solid arrange themselves randomly. In the ground state the particles are arranged in a highly structured lattice and the energy of the system is minimal.

The physical annealing process can be modeled successfully by computer simulation methods based on *Monte Carlo techniques*. An introductory overview of the use of these techniques in statistical physics is given by Binder [1978]. Here, we discuss one of the early techniques proposed by Metropolis et al. [1953], who gave a simple algorithm for simulating the evolution of a solid in a heat bath to *thermal equilibrium*. Their algorithm is based on Monte Carlo techniques and generates a sequence of states of the solid in the following way. Given a current state i of the solid with *energy* E_i, a subsequent state j is generated by applying a perturbation mechanism, which transforms the current state into the next state by a small distortion, for instance by displacement of a single particle. The energy of the next state is E_j. If the *energy difference*, $E_j - E_i$, is less than or equal to 0, the state j is accepted as the current state. If the energy difference is greater than 0, the state j is accepted with a probability given by

$$\exp\left(\frac{E_i - E_j}{k_B T}\right),$$

where T denotes the *temperature* of the heat bath and k_B is a physical constant known as the *Boltzmann constant*. The acceptance rule described above is known as the *Metropolis criterion*, the corresponding algorithm as the *Metropolis algorithm*.

If the temperature is lowered sufficiently slowly, the solid can reach thermal equilibrium at each temperature. Thermal equilibrium is characterized by the *Boltzmann distribution*, which relates the probability of the solid of being in a state i with energy E_i to the temperature T, and is given by

$$\mathbb{P}_T\{\mathbf{X} = i\} = \frac{\exp(-E_i/k_B T)}{\sum_j \exp(-E_j/k_B T)},$$

where $\mathbf{X}$ is a random variable denoting the current state of the solid and the summation extends over all possible states. As we show in the following sections, the Boltzmann distribution plays an essential role in the analysis of the simulated annealing algorithm.

Returning to simulated annealing, we can apply the Metropolis criterion to generate a sequence of solutions of a combinatorial optimization problem. For this purpose we assume an analogy between a physical many-particle system and a combinatorial optimization problem based on the following equivalences:

- Solutions in a combinatorial optimization problem are equivalent to states of a physical system.
- The cost of a solution is equivalent to the energy of a state.

Next we introduce a parameter that plays the role of the temperature. This parameter is the same as the control parameter used in equation (1).

A characteristic feature of simulated annealing is that, besides accepting improvements in cost, it also accepts to a limited extent deteriorations in cost. Initially, at large values of c, large deteriorations are accepted; as c decreases, only smaller deteriorations are accepted, and as the value of c approaches 0, no deteriorations are accepted at all. Furthermore, there is no limitation on the size of a deterioration with respect to its acceptance, such as occurs in threshold accepting. In simulated annealing, arbitrarily large deteriorations are accepted with positive but small probability.

5 MARKOV MODELS

The simulated annealing algorithm can be mathematically modeled using the theory of finite Markov chains [Feller, 1950; Isaacson & Madsen, 1976; Seneta, 1981].

Definition 1 *Let $\mathcal{O}$ denote a set of possible outcomes of a sampling process. A* Markov chain *is a sequence of* trials, *where the probability of the outcome of a given trial depends only on the outcome of the previous trial. Let $\mathbf{X}(k)$ be a random variable denoting the outcome of the kth trial. Then the* transition probability *at the kth trial for each pair of outcomes $i, j \in \mathcal{O}$ is defined as*

$$P_{ij}(k) = \mathbb{P}\{\mathbf{X}(k) = j \mid \mathbf{X}(k-1) = i\}. \tag{2}$$

The matrix $P(k)$, whose elements are given by equation (2), is called the *transition matrix*. A Markov chain is called *finite* if the set of outcomes is finite. It is called *inhomogeneous* if the transition probabilities depend on the trial number k. If they do not depend on the trial number, the Markov chain is called *homogeneous*.

Let $a_i(k)$ denote the probability of outcome $i \in \mathcal{O}$ at the kth trial, i.e.,

$$a_i(k) = \mathbb{P}\{\mathbf{X}(k) = i\}. \tag{3}$$

Then for all $i \in \mathcal{O}$, $a_i(k)$ is given as

$$a_i(k) = \sum_{l \in \mathcal{O}} a_l(k-1) P_{li}(k).$$

Definition 2 *An n-vector x is called* stochastic *if its components x_i satisfy the conditions*

$$x_i \geqslant 0,\ i = 1, \ldots, n, \quad \text{and} \quad \sum_{i=1}^{n} x_i = 1.$$

An $n \times m$ matrix X is called stochastic *if its components X_{ij} satisfy the conditions*

$$X_{ij} \geqslant 0,\ i = 1, \ldots, n, j = 1, \ldots, m, \quad \text{and} \quad \sum_{j=1}^{m} X_{ij} = 1,\ i = 1, \ldots, n.$$

In the case of simulated annealing, a trial corresponds to a transition, and the set of outcomes is given by the finite set of solutions. Furthermore, the outcome of a trial only depends on the outcome of the previous trial. Consequently, we can safely apply the concept of finite Markov chains.

Definition 3 (transition probability) *Let $(\mathscr{S}, f)$ be an instance of a combinatorial optimization problem and $\mathcal{N}$ a neighborhood function. Then the transition probabilities for the simulated annealing algorithm are defined as*

$$\forall i, j \in \mathscr{S}: P_{ij}(k) = \begin{cases} G_{ij}(c_k) A_{ij}(c_k) & \text{if } i \neq j, \\ 1 - \sum_{l \in S, l \neq i} G_{il}(c_k) A_{il}(c_k) & \text{if } i = j, \end{cases} \tag{4}$$

where $G_{ij}(c_k)$ denotes the generation probability, i.e., the probability of generating a solution j from a solution i, and $A_{ij}(c_k)$ denotes the acceptance probability, i.e., the probability of accepting a solution j that is generated from solution i.

Note that the matrix P of equation (4) is stochastic. The $G_{ij}(c_k)$'s and $A_{ij}(c_k)$'s of (4) are conditional probabilities, i.e., $G_{ij}(c_k) = \mathbb{P}_{c_k}$ {generate $j|i$} and $A_{ij}(c_k) = \mathbb{P}_{c_k}$ {accept $j|i,j$}. The corresponding matrices $G(c_k)$ and $A(c_k)$ are the *generation matrix* and *acceptance matrix*, respectively, and need not be stochastic.

In the original version of simulated annealing the following probabilities are used.

Definition 4 *The* generation probability *is defined by*

$$\forall i, j \in \mathscr{S}\colon\ G_{ij}(c_k) = G_{ij} = \frac{1}{\Theta}\chi_{(\mathscr{N}(i))}(j), \tag{5}$$

where $\Theta = |\mathscr{N}(i)|$, *for all* $i \in \mathscr{S}$. *The characteristic function* $\chi_{(X')}$ *for a subset* X' *of a given set* X *is a mapping of the set* X *onto the set* $\{0, 1\}$, *such that* $\chi_{(X')}(x) = 1$ *if* $x \in X'$, *and* $\chi_{(X')}(x) = 0$ *otherwise.*

The acceptance probability *is defined by*

$$\forall i, j \in \mathscr{S}\colon\ A_{ij}(c_k) = \exp\left(-\frac{(f(j) - f(i))^+}{c_k}\right), \tag{6}$$

where, for all $a \in \mathbb{R}$, $a^+ = a$ *if* $a > 0$, *and* $a^+ = 0$ *otherwise.*

Thus, the generation probabilities are chosen to be independent of the control parameter c and uniform over the neighborhoods $\mathscr{N}(i)$, where it is assumed that all neighborhoods are of equal size, i.e., $|\mathscr{N}(i)| = \Theta$, for all $i \in \mathscr{S}$.

The above definitions apply to most combinatorial optimization problems, and close examination of the literature reveals that in many practical applications these definitions – or minor variations – are indeed used. But it is also possible to formulate a set of conditions guaranteeing asymptotic convergence for a more general class of acceptance and generation probabilities. We return to this subject below.

We now concentrate on the asymptotic convergence of simulated annealing. A simulated annealing algorithm finds an optimal solution with probability 1 if, after a possibly large number of trials, say k, we have

$$\mathbb{P}\{\mathbf{X}(k) \in \mathscr{S}^*\} = 1,$$

where $\mathscr{S}^*$ denotes the set of optimal solutions. In the following sections we show that under certain conditions the simulated annealing algorithm converges asymptotically to the set of optimal solutions, i.e.,

$$\lim_{k \to \infty} \mathbb{P}\{\mathbf{X}(k) \in \mathscr{S}^*\} = 1.$$

An essential property in the study of Markov chains is that of *stationarity*. Under certain conditions on the transition probabilities associated with a Markov chain there exists a unique stationary distribution; see Feller [1950] and Isaacson & Madsen [1976].

Definition 5 (stationarity) *A* stationary distribution *of a finite homogeneous Markov chain with transition matrix* P *on a set of outcomes* $\mathscr{O}$ *is defined as the stochastic*

$|\mathcal{O}|$-vector q, whose components are given by

$$q_i = \lim_{k\to\infty} \mathbb{P}\{\mathbf{X}(k) = i \,|\, \mathbf{X}(0) = j\}, \quad \textit{for all } j \in \mathcal{O}.$$

If such a stationary distribution q exists, we have $\lim_{k\to\infty} a_i(k) = q_i$, where $a_i(k)$ is given by equation (3). Furthermore, it follows directly from the definitions that $q^T = q^T P$. Thus, q is the probability distribution of the outcomes after an infinite number of trials and the left eigenvector of P with eigenvalue 1. In the case of the simulated annealing algorithm, as P depends on c, q depends on c, i.e., $q = q(c)$.

Before we can prove the existence of a stationary distribution for the simulated annealing algorithm, we need the following definitions.

Definition 6 (irreducibility) *A Markov chain with transition matrix P on a set of outcomes $\mathcal{O}$ is* irreducible *if for each pair of outcomes $i, j \in \mathcal{O}$ there is a positive probability of reaching j from i in a finite number of trials, i.e.,*

$$\forall i, j \in \mathcal{O}\ \exists n \in \mathbb{Z}^+ : (P^n)_{ij} > 0.$$

Definition 7 (aperiodicity) *A Markov chain with transition matrix P is* aperiodic *if for each outcome $i \in \mathcal{O}$ the greatest common divisor* gcd$(\mathcal{D}_i) = 1$, where *$\mathcal{D}_i$ is the set of all integers $n > 0$ with $(P^n)_{ii} > 0$.*

The integer gcd $(\mathcal{D}_i)$ is called the *period* of i. Thus, aperiodicity requires all solutions to have period 1. As a corollary we have that for an irreducible Markov chain aperiodicity holds if

$$\exists j \in \mathcal{O}: P_{jj} > 0. \tag{7}$$

We now come to the following important theorem; see Feller [1950] and Isaacson & Madsen [1976] for its proof.

Theorem 1 *Let P be the transition matrix associated with a finite homogeneous Markov chain on a set of outcomes $\mathcal{O}$, and let the Markov chain by irreducible and aperiodic. Then there exists a unique stationary distribution q whose components q_i are uniquely determined by*

$$\sum_{j\in\mathcal{O}} q_j P_{ji} = q_i \quad \textit{for all } i \in \mathcal{O}.$$

As a corollary we have that any probability distribution q that is associated with a finite, irreducible and aperiodic homogeneous Markov chain and that satisfies the equations

$$q_i P_{ij} = q_j P_{ji} \quad \text{for all } i, j \in \mathcal{O}, \tag{8}$$

is the unique stationary distribution in the sense mentioned in Theorem 1. The equations of (8) are called the *detailed balance equations*, and a Markov chain for which they hold is called *reversible*.

6 A HOMOGENEOUS MODEL

We now can prove asymptotic convergence of simulated annealing based on a model in which the algorithm is viewed as a sequence of Markov chains of infinite length. In this case we say that the value of the control parameter is independent of k, i.e., $c_k = c$ for all k. This leads to the following result.

Theorem 2 *Let $(\mathscr{S}, f)$ be an instance of a combinatorial optimization problem, $\mathscr{N}$ a neighborhood function, and $P(k)$ the transition matrix of the homogeneous Markov chain associated with the simulated annealing algorithm defined by* (4), (5), *and* (6), *with $c_k = c$ for all k. Furthermore, let the following condition be satisfied:*

$$\forall i, j \in \mathscr{S}\ \exists p \geqslant 1\ \exists l_0, l_1, \ldots, l_p \in \mathscr{S}$$

with

$$l_0 = i,\ l_p = j, \quad and \quad G_{l_k l_{k+1}} > 0, \quad k = 0, 1, \ldots, p-1.$$

Then the associated homogeneous Markov chain has a stationary distribution $q(c)$, whose components are given by

$$q_i(c) = \frac{\exp(-f(i)/c)}{\sum_{j \in \mathscr{S}} \exp(-f(j)/c)} \ for\ all\ i \in \mathscr{S}, \tag{9}$$

and

$$q_i^* \overset{\text{def}}{=} \lim_{c \downarrow 0} q_i(c) = \frac{1}{|\mathscr{S}^*|} \chi_{(\mathscr{S}^*)}(i), \tag{10}$$

where i^* denotes an optimal solution, and $\mathscr{S}^*$ the set of optimal solutions.

The proof of the theorem follows directly from the previous results. Indeed, the condition in the theorem guarantees that the Markov chain is irreducible; see Definition 6. The transition probabilities given by (4) with (5) and (6) guarantee aperiodicity through (7); see Definition 7. Hence, according to Theorem 1, there exists a unique stationary distribution, and the correctness of the components of (9) follows directly from the detailed balance equations of (8), which proves the first part of the theorem. The second part follows directly from (9). The distribution given by (9) is the equivalent of the Boltzmann distribution in the Monte Carlo simulations of the physical annealing process mentioned in Section 4. It is characteristic of simulated annealing and, as we show below, it plays an important role in the analysis of the algorithm.

As a result of Theorem 2 we have that

$$\lim_{c \downarrow 0} \lim_{k \to \infty} \mathbb{P}_c\{\mathbf{X}(k) \in \mathscr{S}^*\} = 1. \tag{11}$$

This result reflects the basic property of the simulated annealing algorithm, i.e., the guarantee that the algorithm asymptotically finds an optimal solution. Furthermore, (11) expresses the characteristic of the homogeneous model for simulated annealing: first take the limit of the homogeneous Markov chain for an infinite number of trials, then take the limit for the control parameter to zero. In the inhomogeneous model of Section 7 these two limits are combined into a single limit.

The convergence properties discussed above strongly depend on the original choice of the transition probabilities of (4), (5), and (6). Several authors have investigated convergence properties for a broader class of probabilities. This has led to the following formulation.

Theorem 3 *Let $(\mathcal{S}, f)$ be an instance of a combinatorial optimization problem, $\mathcal{N}$ a neighborhood function, and $P(k)$ the transition matrix of the homogeneous Markov chain associated with the simulated annealing algorithm defined by* (4), *with $c_k = c$ for all k. Furthermore, let the following conditions hold:*

(G1) $\forall c > 0\ \forall i, j \in \mathcal{S}\ \exists p \geqslant 1\ \exists l_0, l_1, \ldots, l_p \in \mathcal{S}$ *with* $l_0 = i$, $l_p = j$,
and $G_{l_k l_{k+1}}(c) > 0, \quad k = 0, 1, \ldots, p-1$,
(G2) $\forall c > 0\ \forall i, j \in \mathcal{S}$: $\quad G_{ij}(c) = G_{ji}(c)$,
(A1) $\forall c > 0\ \forall i, j \in \mathcal{S}$: $\quad A_{ij}(c) = 1$ *if* $f(i) \geqslant f(j)$,
$A_{ij}(c) \in (0, 1)$ *if* $f(i) < f(j)$,
(A2) $\forall c > 0\ \forall i, j, k \in \mathcal{S}$: $A_{ij}(c) A_{jk}(c) A_{ki}(c) = A_{ik}(c) A_{kj}(c) A_{ji}(c)$,
(A3) $\forall i, j \in \mathcal{S}$ *with* $f(i) < f(j)$: $\lim_{c \downarrow 0} A_{ij}(c) = 0$.

Then the Markov chain has a unique stationary distribution $q(c)$, whose components are given by

$$q_i(c) = \frac{1}{\sum_{j \in \mathcal{S}} (A_{ij}(c)/A_{ji}(c))} \quad \textit{for all } i \in \mathcal{S}, \tag{12}$$

and

$$\lim_{c \downarrow 0} q_i(c) = q_i^*,$$

where the q_i^* are defined by (10).

Condition (G1) again guarantees irreducibility and aperiodicity of the corresponding Markov chain, conditions (G2), (A1) and (A2) guarantee reversibility, and condition (A3) guarantees that stationary distributions concentrate on the set of optimal solutions as c approaches 0. In general terms, the conditions require the following. Condition (G1) requires that each solution can be reached from any other solution by generating a finite sequence of neighboring solutions. To ensure this, the corresponding neighborhood graph should be strongly connected. Condition (G2) requires symmetry of the generation matrix. Conditions (A1) through (A3) require that the acceptance matrix is well behaved, i.e., improvements are always accepted and deteriorations are accepted with positive probability for $c > 0$ (condition (A1)) and with zero probability for $\lim c \downarrow 0$ (condition (A3)); the factorization required by condition (A2) guarantees detailed balance as defined by (8).

Conditions (G1) through (A3) are sufficient but not necessary. Thus, there may be acceptance and generation matrices not satisfying these conditions and still ensuring the existence of the stationary distribution. An example of such an acceptance matrix is

$$A_{ij}(c) = \frac{1}{1 + \exp(-(f(i) - f(j))/c)}. \tag{13}$$

This acceptance matrix does not satisfy conditions (A1) and (A2), but it can be shown to lead to the stationary distribution of (9) by using the detailed balance equations of (8).

Furthermore, several authors have addressed the generality issue of the acceptance probability. We discuss this issue following the lines of the work of Schuur [1997].

Theorem 4 *Let $P(k)$ be the transition matrix of the homogeneous Markov chain associated with the simulated annealing algorithm defined by* (4), *with $c_k = c$ for all k, and let conditions* (G1) *and* (G2) *of Theorem 3 hold. Furthermore, let the acceptance probabilities be defined as follows. Given are two functions $\phi:(0,\infty) \times \mathbb{R} \rightarrow (0,\infty)$ and $H:(0,\infty) \times \mathbb{R} \times \mathbb{R} \rightarrow (0,1]$, such that for $c > 0$, and $x, y \in \mathbb{R}: H(c,x,y) = H(c,y,x)$, and*

$$A_{ij}(c) = H(c, f(i), f(j)) \min\left(1, \frac{\phi(c, f(j))}{\phi(c, f(i))}\right),$$

and

$$\forall x, y \in \mathbb{R}: x > y \Rightarrow \lim_{c \downarrow 0} \frac{\phi(c, y)}{\phi(c, x)} = 0. \tag{14}$$

Then the Markov chain has a unique stationary distribution $q(c)$, whose components are given by

$$q_i(c) = \frac{\phi(c, f(i))}{\sum_{j \in \mathcal{S}} \phi(c, f(j))} \quad \text{for all } i \in \mathcal{S}, \tag{15}$$

and

$$\lim_{c \downarrow 0} q_i(c) = q_i^*,$$

where the q_i^* are again given by (10).

As a corollary to Theorem 4 it is argued that the only well-behaved function $\phi(c, f(j))$ that satisfies (14) is of the form

$$\phi(c, f(j)) = \exp(\gamma(c) f(j)), \tag{16}$$

where γ: $(0,\infty) \rightarrow (0,\infty)$ and $\lim_{c \downarrow 0} \gamma(c) = \infty$.

Kesidis [1990] and Romeo & Sangiovanni-Vincentelli [1991] provide arguments for the assertion that the fastest convergence to the stationary distribution of (15) with (16) and $\gamma = c^{-1}$ is given by the acceptance probabilities of (6).

Below we give alternatives for conditions (G1) and (G2), respectively.

Condition (G1) can be replaced by a more general condition. This condition guarantees that the Markov chain associated with the generation matrix G is irreducible. If this is not the case, asymptotic convergence to a subset of the set of globally optimal solutions can still be proved if condition (G1) is replaced by the following necessary and sufficient condition:

$$(G1') \quad \forall i \in \mathscr{S} \; \exists i^* \in \mathscr{S}^*, p \geqslant 1 \; \exists l_0, l_1, \ldots l_p \in \mathscr{S}$$

$$\text{with } l_0 = i, \, l_p = i^*, \text{ and } G_{l_k l_{k+1}} > 0, \quad k = 0, 1, \ldots, p-1. \tag{17}$$

According to this condition it should be possible to construct a finite sequence of transitions with nonzero generation probability, leading from an arbitrary solution i to some optimal solution i^*. For the proof of the validity of this condition, a distinction must be made between *transient* and *recurrent* solutions, where a solution is called transient if the probability that the Markov chain ever returns to that solution equals zero, and recurrent if the Markov chain may return to the solution with a positive probability [Feller, 1950]. Furthermore, the stationary distribution of (12) does not apply any more and should be replaced by a *stationary matrix* $Q(c)$ whose elements q_{ij} denote the probability of finding a solution j after an infinite number of transitions, starting from a solution i. A more detailed treatment is beyond the scope of this chapter; the reader is referred to Connors & Kumar [1987], Gidas [1985], Van Laarhoven [1988], and Van Laarhoven, Aarts & Lenstra [1992].

In practice one does not want to bother about the requirement of condition (G2) that the generation matrix must be symmetric. Easier to implement is a uniform distribution over the neighborhoods, similar to that used in the original version of simulated annealing. Lundy & Mees [1986] show that for the choice of the generation probabilities given by

$$G_{ij} = \frac{1}{|\mathscr{N}(i)|} \chi_{(\mathscr{N}(i))}(j) \quad \text{for all } i, j \in \mathscr{S}, \tag{18}$$

condition (G2) is no longer needed to guarantee asymptotic convergence, and the components of the stationary distribution are then given by

$$q_i(c) = \frac{|\mathscr{N}(i)|}{\sum_{j \in \mathscr{S}} |\mathscr{N}(j)| A_{ij}(c)/A_{ji}(c)}.$$

Moreover, it follows directly that these components again converge to the q_i^* of (10) as $c \downarrow 0$. Finally, we mention that a generation matrix satisfying both condition (G2) and (18) implies that $|\mathscr{N}(i)|$ is independent of i.

7 AN INHOMOGENEOUS MODEL

In the previous section it was shown that, under certain conditions on the generation and acceptance matrices, the simulated annealing algorithm converges to a global minimum with probability 1, if for each value of the control parameter c_k, $k = 0, 1, 2, \ldots$, the corresponding homogeneous Markov chain is infinitely long and the sequence $(c_k|k=0,1,2,\dots)$ eventually converges to 0 as $k\to\infty$. In this section we discuss conditions to guarantee asymptotic convergence for the case where each Markov chain is of finite length. Thus, the simulated annealing algorithm is modeled as an inhomogeneous Markov chain with the following transition probabilities:

$$P_{ij}(c_k)=\begin{cases} G_{ij}(c_k)A_{ij}(c_k) & j\neq i, \\ 1-\sum\limits_{l\in\mathscr{S},l\neq i} G_{il}(c_k)A_{il}(c_k) & j=i. \end{cases} \tag{19}$$

Furthermore, we assume that the sequence $(c_k|k=0,1,2,\dots)$ satisfies the conditions

$$\lim_{k\to\infty} c_k=0, \text{ and} \tag{20}$$

$$c_k\geqslant c_{k+1},\ k=0,1,\dots. \tag{21}$$

Thus, c_k is kept constant during a number of transitions, in which case we obtain an inhomogeneous Markov chain consisting of an infinite number of homogeneous Markov chains of finite length each.

We show that, under certain conditions on the rate of convergence of the sequence $(c_k|k=0,1,2,\dots)$, the inhomogeneous Markov chain associated with the simulated annealing algorithm converges in distribution to q^*, whose components are given by (10). In other words we prove that

$$\lim_{k\to\infty}\mathbb{P}\{\mathbf{X}(k)\in\mathscr{S}^*\}=1.$$

To discuss the convergence of inhomogeneous Markov chains we need the following definitions; see Seneta [1981].

Definition 8 *Let $P(k)$ be the transition matrix associated with an inhomogeneous Markov chain on a set of outcome $\mathcal{O}$. Then the matrix $U(m,k)$ is defined as*

$$U(m,k)=\prod_{n=m}^{k}P(n),\quad 0<m\leqslant k.$$

In other words the components of $U(m,k)$ are equal to

$$U_{ij}(m,k)=\mathbb{P}\{\mathbf{X}(k)=j|\mathbf{X}(m-1)=i\}\quad \text{for all } i,j\in\mathcal{O}.$$

Definition 9 (ergodicy) *A finite inhomogeneous Markov chain on a set of outcomes $\mathcal{O}$ is* weakly ergodic *if*

$$\forall i,j,l\in\mathcal{O},\forall m>0{:}\ \lim_{k\to\infty}(U_{il}(m,k)-U_{jl}(m,k))=0.$$

It is strongly ergodic *if there exists a stochastic vector q^* such that*

$$\forall i,j\in\mathcal{O},\forall m>0{:}\ \lim_{k\to\infty}U_{ij}(m,k)=q_j^*.$$

Thus, for a given m, weak ergodicity implies that $\mathbf{X}(k)$ becomes independent of

$\mathbf{X}(m)$ as $k \to \infty$, whereas strong ergodicity implies *convergence in distribution*, i.e., for any stochastic vector $a(m)$ denoting the probabilities of the outcomes of the mth trial we have that

$$\lim_{k \to \infty} a^T(m-1) \prod_{n=m}^{k} P(n) = (q^*)^T,$$

or

$$\lim_{k \to \infty} \mathbb{P}\{\mathbf{X}(k) = j\} = \lim_{k \to \infty} \left(\sum_{i \in \mathcal{O}} U_{ij}(m,k)\, \mathbb{P}\{\mathbf{X}(m-1) = i\} \right) = q_j^* \quad \text{for all } i,j \in \mathcal{O}.$$

The difference between weak and strong ergodicity can be understood from the following example. Let the transition probabilities $P_{ij}(k)$ of an inhomogeneous Markov chain be independent of j. Then the Markov chain is clearly weakly ergodic, but it is not strongly ergodic if the $P_{ij}(k)$ vary forever with k for a given i. Note that for a homogeneous Markov chain there is no distinction between weak and strong ergodicity.

The following two theorems provide conditions for weak and strong ergodicity of inhomogeneous Markov chains. The proofs can be found in Isaacson & Madsen [1976] and Seneta [1981].

Theorem 5 *Let $\tau_1(X)$ denote the coefficient of ergodicity of the $n \times n$ stochastic matrix X defined as*

$$\tau_1(X) = \frac{1}{2} \max_{i,j=1,\ldots,n} \sum_{l=1}^{n} |X_{il} - X_{jl}|$$

$$= 1 - \min_{i,j=1,\ldots,n} \sum_{l=1}^{n} \min(X_{il}, X_{jl}).$$

Then an inhomogeneous Markov chain is weakly ergodic if and only if there is a strictly increasing sequence of positive numbers $(k_i | i = 0, 1, 2, \ldots)$ such that

$$\sum_{i=0}^{\infty} (1 - \tau_1(X(k_i, k_{i+1}))) = \infty. \tag{22}$$

Theorem 6 *A finite inhomogeneous Markov chain is strongly ergodic under the following conditions:*

(C1) *The Markov chain is weakly ergodic.*

(C2) *For all k there exists a stochastic vector $q(k)$ such that $q(k)$ is the left eigenvector of $P(k)$ with eigenvalue* 1.

(C3) *The eigenvectors $q(k)$ satisfy*

$$\sum_{k=1}^{\infty} \| q(k) - q(k+1) \|_1 < \infty, \tag{23}$$

where the 1*-norm of an n-vector x is defined as $\| x \|_1 = \sum_{i=1}^{n} |x_i|$.*

Moreover, if $q^* = \lim_{k \to \infty} q(k)$, then q^* is the vector of Definition 9.

To prove convergence in distribution for the simulated annealing algorithm it must be shown that the associated inhomogeneous Markov chain is strongly ergodic.

Theorem 7 *Let $(\mathscr{S}, f)$ be an instance of a combinatorial optimization problem, $\mathscr{N}$ a neighborhood function, and $P(k)$ the transition matrix of the inhomogeneous Markov chain associated with the simulated annealing algorithm defined by* (19), (5), *and* (6). *Furthermore, let the following conditions be satisfied:*

(D1)
$$\forall i,j\in\mathscr{S}\ \exists p\geqslant 1\ \exists l_0, l_1,\ldots,l_p\in\mathscr{S}$$

with $l_0=i$, $l_p=j$, *and* $G_{l_k l_{k+1}}>0$, $k=0,1,\ldots,p-1$.

(D2)
$$c_k\geqslant\frac{\Gamma}{\log(k+k_0)},\quad k=0,1,\ldots, \qquad (24)$$

for some value of $\Gamma>0$ *and* $k_0>2$.

Then the Markov chain converges in distribution to the vector q^*, *with components given by* (10), *or in other words*

$$\lim_{k\to\infty}\mathbb{P}\{\mathbf{X}(k)\in\mathscr{S}^*\}=1. \qquad (25)$$

Theorem 7 can be proved by showing that conditions (D1) and (D2) are sufficient to satisfy conditions (C1), (C2), and (C3) of Theorem 5, along the following lines.

Condition (D1) guarantees the existence of the left eigenvector $q(k)$ of $P(k)$, given by $q(k)=q(c_k)$, i.e., the stationary distribution of the homogeneous Markov chain with transition matrix $P=P(k)$; see Theorem 2. The components of the eigenvectors are given by (9). Furthermore, from (20) and (9) we have $\lim_{k\to\infty}q(c_k)=\lim_{c\downarrow 0}q(c)=q^*$, where the components of q^* are given by (10). So condition (C2) of Theorem 6 holds and, by using the explicit form of the eigenvectors $q(k)$, condition (C3) can be shown to hold. What remains to be shown is that the Markov chain is weakly ergodic, as stipulated by condition (C1). This can be done by using Theorem 5 and condition (D2).

The latter proof is quite technical, and several authors have come up with different approaches, which vary predominantly in the estimates of the values of the constant Γ.

One of the first results was obtained by Mitra, Romeo & Sangiovanni-Vincentelli [1986]. To discuss this we need the following definition.

Definition 10 *The distance $d(i,j)$ between two solutions $i,j\in\mathscr{S}$ is defined as the length d of the shortest sequence of solutions $(l_0, l_1,\ldots,l_d)$, with $l_0=i$, $l_d=j$, and $P_{l_m l_{m+1}}(k)>0$, $l_m\in\mathscr{S}$, $m=0,1,\ldots,d-1$.*

Mitra, Romeo & Sangiovanni-Vincentelli [1986] found that $\Gamma\geqslant r\Delta$ with

$$\Delta=\max_{i\in\mathscr{S}}\max_{j\in\mathscr{N}(i)}\{|f(j)-f(i)|\}, \qquad (26)$$

and

$$r = \min_{i \in \mathcal{S} \setminus \hat{\mathcal{S}}} \max_{j \in \mathcal{S}} d(i,j), \tag{27}$$

where $\hat{\mathcal{S}}$ denotes the set of all locally minimal solutions.

Other values of Γ are given by Anily & Federgruen [1987a], Gelfand & Mitter [1985], Geman & Geman [1984], Gidas [1995], and Holley & Stroock [1988]; for an overview see Romeo & Sangiovanni-Vincentelli [1991].

The conditions for asymptotic convergence given above are *sufficient* but not *necessary*. Necessary and sufficient conditions are derived by Hajek [1988]. The difference with the conditions presented above again lies in the difference in value of the constant Γ. To discuss Hajek's result we need the following definitions.

Definition 11 *Let $i, j \in \mathcal{S}$, then j is* reachable *at height h from i if $i = j$ and $f(i) \leqslant h$, or $\exists p \geqslant 1\ \exists l_0, \ldots, l_p \in \mathcal{S}$ with $l_0 = i$ and $l_p = j$ such that $G_{l_k, l_{k+1}} > 0$ and $f(l_k) \leqslant h$ for all $k = 0, \ldots, p-1$.*

Definition 12 *Let $\hat{i}$ be a local minimum. Then the* depth *$d(\hat{i})$ of $\hat{i}$ is the smallest number $x, x > 0$, such that there is a solution $j \in \mathcal{S}$ with $f(j) < f(\hat{i})$ that is reachable at height $f(\hat{i}) + x$ from $\hat{i}$. By definition, for an optimal solution i^*, $d(i^*) = \infty$.*

We now can formulate the results obtained by Hajeck.

Theorem 8 *Let $(c_k | k = 0, 1, \ldots)$ be a sequence of values of the control parameter defined as*

$$c_k = \frac{\Gamma}{\log(k+2)}, \quad k = 0, 1, \ldots,$$

for some constant Γ. Then asymptotic convergence of the simulated annealing algorithm, using the transition probabilities of (19), (5), *and* (6), *is guaranteed if and only if*

- *the Markov chain is irreducible,*
- *i is reachable from j at height h if and only if j is reachable from i at height h, for arbitrary $i, j \in \mathcal{S}$ and h, and*
- *the constant Γ satisfies $\Gamma \geqslant D$, where*

$$D = \max_{\hat{i} \in \hat{\mathcal{S}} \setminus \mathcal{S}^*} d(\hat{i}), \tag{28}$$

i.e., D is the depth of the deepest local, nonglobal minimum.

Kern [1993] has addressed the problem of calculating the value of D. In particular, he showed for a number of problems how it is unlikely that D can be calculated in polynomial time for arbitrary instances of a combinatorial optimization problem. He also presents bounds on the value of D for several combinatorial optimization problems.

Under certain conditions, asymptotic convergence of the inhomogeneous Markov chain associated with the simulated annealing algorithm can also be proved for general conditions on the generation and acceptance probabilities. This result was first proved by Anily & Federgruen [1987b] and can be formulated as follows.

Theorem 9 *Let the transition probabilities of the inhomogeneous Markov chain associated with the simulated annealing algorithm be defined by* (19), *and let the generation probabilities $G_{ij}(c)$ and acceptance probabilities $A_{ij}(c)$ satisfy conditions* (G1) *through* (A3) *of Theorem 3. Furthermore, let*

$$A(c) = \min_{i,j}\{A_{ij}(c) \mid i \in \mathscr{S}, j \in \mathscr{N}(i)\}.$$

Then

$$\lim_{k \to \infty} \mathbb{P}\{\mathbf{X}(k) \in \mathscr{S}^*\} = 1,$$

if

$$\sum_{k=0}^{\infty} (\underline{A}(c_k))^n = \infty, \tag{29}$$

where n denotes the maximum number of steps needed to reach an optimal solution from any arbitrary solution, and the constant Γ satisfies

$$\Gamma \geqslant n\Delta, \tag{30}$$

where Δ is given by (26).

Finally, we mention that Gelfand & Mitter [1985] derived sufficient conditions for convergence to an arbitrary set of solutions. These conditions are similar to those given above.

8 ASYMPTOTIC BEHAVIOR

We have shown that, under mild conditions, the simulated annealing algorithm converges in probability to the set of optimal solutions, or in other words, asymptotically the algorithm finds an optimal solution with probability 1. As a result of the limits of (11) or (25), asymptotic convergence to the set of optimal solutions is achieved only after an infinite number of transitions. In any finite-time implementation one must resort to approximations of the asymptotic convergence.

With respect to the approximation of the stationary distribution we have the following two properties; see Seneta [1981].

Property 10 *Let $P(k)$ denote the transition matrix of the homogeneous Markov chain associated with the simulated annealing algorithm defined by* (4), *and let $q(c)$ denote the corresponding stationary distribution given by the left eigenvector with eigenvalue 1 of P. Then, as $k \to \infty$, we have*

$$\| a(k) - q(c) \|_1 = O(k^s |\lambda_2(c)|^k), \tag{31}$$

where $a(k)$ denotes the probability distribution of the outcomes after k trials, $\lambda_2(c)$ $(0 < |\lambda_2(c)| < 1)$ denotes the second largest eigenvalue of $P(k)$ with multiplicity m_2, and $s = m_2 - 1$.

Hence, the speed of convergence to the stationary distribution is determined by $\lambda_2(c)$. Unfortunately, computation of $\lambda_2(c)$ is impracticable, due to the large size of matrix $P(k)$. Approximation of the norm in (31) leads to the following property; see Aarts & Van Laarhoven [1985b].

Property 11 *Let ε denote an arbitrarily small positive number. Then*

$$\| a(k) - q(c) \|_1 < \varepsilon, \tag{32}$$

if

$$k > K\left(1 + \frac{\ln(\varepsilon/2)}{\ln(1 - \gamma^K(c))}\right), \tag{33}$$

where $\gamma(c) = \min_{i,j \in \mathscr{S}} P^+_{ij}(c)$ and $K = |\mathscr{S}|^2 - 3|\mathscr{S}| + 3$.

Hence, (32) and (33) indicate that the stationary distribution is approximated arbitrarily closely, only if the number of transitions is at least quadratic in the size of the solution space. Moreover, the size $|\mathscr{S}|$ is for most problems exponential in the size of the problem itself; for instance, in the n-city traveling salesman problem, $|\mathscr{S}| = (n-1)!$. Thus, the analysis presented above indicates that approximating the stationary distribution arbitrarily closely results in an exponential-time execution of the simulated annealing algorithm.

With respect to the asymptotic convergence of the inhomogeneous Markov chain associated with the simulated annealing algorithm, we have the following result; see Mitra, Romeo & Sangiovanni-Vincentelli [1986].

Property 12 *Let the transition probabilities of the inhomogeneous Markov chain associated with the simulated annealing algorithm be defined by (19), (5), and (6), and let the sequence $(c_k | k = 0, 1, \ldots)$ be given by (24), with $\Gamma > r\Delta$, where r and Δ are defined as in (26) and (27). Furthermore, let q^* be the uniform probability distribution on the set of optimal solutions defined by (10). Then for $k \to \infty$,*

$$\| a(k) - q^* \|_1 < \varepsilon,$$

for an arbitrarily small positive number ε, if

$$k = O(\varepsilon^{\quad \max(a,b)}),$$

where

$$a = \frac{\Gamma^{r\Delta/\Gamma}}{w^r}, \quad \text{and} \quad b = \frac{r\Delta}{\hat{f} - f^*},$$

with $\hat{f} = \min_{i \in \mathscr{S} \setminus \mathscr{S}^*} f(i)$ and $w = \min_{i \in \mathscr{S}} \min_{j \in \mathscr{S}(i)} G_{ij}$.

Evaluation of this bound for particular problems typically leads to a number of transitions that is larger than the size of the solution space and thus to an exponential-time execution for most problems. For instance, in the case of the traveling salesman problem, Aarts & Korst [1989a] show that

$$k = O(n^{n^{2n-1}}).$$

Note that $|\mathscr{S}| = (n-1)!$ Hence, complete enumeration of all solutions would take less time than approximating an optimal solution arbitrarily closely by the simulated annealing algorithm.

Summarizing, we have shown that optimal simulated annealing algorithms require an infinite number of transitions and that the rate of convergence is logarithmic, leading to exponential time complexities for arbitrarily close approximation of an optimal solution.

Several authors have investigated possibilities of speeding up the convergence of optimal simulated annealing for specific problems by taking into account the combinatorial structure of the problem at hand. For instance, Sorkin [1991] proved that, if the neighborhoods of a problem exhibit certain fractal properties, the time complexity of optimal simulated annealing is polynomial. More specifically, he showed that, for problems with properly scaled cost functions between 0 and 1, and a fractal neighborhood structure, a solution of expected cost no greater than ε can be found in a time bounded by a polynomial in $1/\varepsilon$, where the exponent of the polynomial depends on the fractal.

Stander & Silverman [1994] discuss a simple global optimization problem and propose an optimal method for lowering the value of the control parameter based on the use of dynamic programming techniques. The resulting time complexity is still exponential but the method provides optimal choices for the initial and final values of the control parameter. Christoph & Hoffmann [1993] address the scaling behavior of optimal annealing. They found that *dominating barriers* exist at which the value of the control parameter must be lowered more slowly than in between the barriers.

Rajasekaran & Reif [1992] obtained improved convergence rate of optimal annealing by exploiting a special property of the cost function, if present, which they call *small-separability*. Based on this concept, they developed an algorithm called *nested annealing*, which is a simple modification of the classical simulated annealing algorithm obtained by assigning different control parameter values to different regions. For a specific class of problems in computer vision and circuit layout, they proved that the time complexity of their optimal simulated algorithm is $2^{O(\sqrt{n})}$ instead of $2^{\Omega(n)}$, where n refers to the size of the problem instance at hand.

9 COOLING SCHEDULES

We now leave the issue of optimal annealing and turn to finite-time implementations of the algorithm. Earlier sections have indicated that finite-time implementations can no longer guarantee to find an optimal solution, but may result in much faster executions of the algorithm without significantly compromising the solution quality. Ingber [1993] refers to such implementations as *simulated quenching*.

A *finite-time* implementation of the simulated annealing algorithm is obtained by generating a sequence of homogeneous Markov chains of finite length at descending values of the control parameter. For this, a set of parameters must be specified that govern the convergence of the algorithm. These parameters are combined in what is called a cooling schedule.

Definition 13 *A* cooling schedule *specifies a finite sequence of values of the control parameter, and a finite number of transitions at each value of the control parameter. More precisely, it is specified by*

- *an* initial *value of the control parameter* c_0,
- *a* decrement function *for lowering the value of the control parameter,*
- *a* final value *of the control parameter specified by a stop criterion,*
- *a* finite length *of each homogeneous Markov chain.*

Central to the discussion of cooling schedules is the concept of quasi-equilibrium. Let L_k denote the length of the kth Markov chain and c_k the corresponding value of the control parameter. Then *quasi-equilibrium* is achieved if the probability distribution $a(L_k, c_k)$ of the solutions, after L_k trials of the kth Markov chain, is 'sufficiently close' to the stationary distribution at c_k, i.e.,

$$\| a(L_k, c_k) - q(c_k) \|_1 < \varepsilon, \tag{34}$$

for some specified positive value of ε.

From Property 11 we recall that a number of transitions is required that is quadratic in the size of the solution space in order to satisfy (34) for arbitrarily small values of ε, which leads to an exponential-time execution for most problems. Thus, in order to be of practical use, a less rigid quantification of the quasi-equilibrium concept is needed than that of (34). For this one may resort to the following arguments. For the acceptance probabilities of (6), and well-behaved generation probabilities, the stationary distribution is of the form given by (9). For $c \to \infty$, the stationary distribution is given by a uniform distribution on the set of solutions $\mathcal{S}$, i.e., if

$$\lim_{c \to \infty} q_i(c) = \frac{1}{|\mathcal{S}|}. \tag{35}$$

Thus, at sufficiently large values of c_k allowing acceptance of virtually all proposed transitions – quasi-equilibrium is obtained by definition, since all solutions occur with equal probability given by the uniform distribution of (35). Next, the decrement function and the Markov chain lengths must be chosen such that quasi-equilibrium is restored at the end of each individual Markov chain. In this way the equilibrium distributions for the various Markov chains are 'closely followed', so as to arrive eventually, as $c_k \downarrow 0$, close to q^*, the uniform distribution on the set of optimal solutions given by (10).

It is intuitively clear that large decrements in c_k require larger Markov chain lengths in order to restore quasi-equilibrium at the next value c_{k+1} of the control parameter. Thus, there is a trade-off between large decrements of the control parameter and small Markov chain lengths. Usually, one chooses small decrements, in c_k to avoid extremely long chains, but one could use large values for L_k in order to be able to make large decrements in c_k.

The search for adequate cooling schedules has been the subject of many studies over the past year. Reviews are given by Van Laarhoven & Aarts [1987], Collins, Eglese & Golden [1988], and Romeo & Sangiovanni-Vincentelli [1991].

9.1 Optimal schedules

Recently, researchers have been investigating optimal finite-time schedules, where optimal refers to the best average cost obtained in finite time. Strenski & Kirkpatrick [1991] analyzed a small instance of a graph partitioning problem and used an approach based on evaluating exactly the probability distributions of outcomes of the Markov chain associated with the simulated annealing algorithm. They found that different schedules, including iterative improvement, may be optimal depending on the employed schedule length. When a sufficiently long schedule is employed, annealing replaces iterative improvement as the optimal schedule. Furthermore, they observed that optimal schedules may be nonmonotone. This result was rather unexpected since the convergence proofs of simulated annealing suggest a monotone lowering of the control parameter value; see for instance Aarts & Korst [1989a]. Nevertheless, it was in accordance with earlier theoretical results obtained by Hajek & Sasaki [1989], who found for a small artificial problem that the control parameter values of an optimal annealing schedule are all either 0 or ∞.

The approach of Strenski & Kirkpatrick [1991] has been further pursued by Boese & Kahng [1994]. They introduce the concept of *best-so-far* versus *where-you-are*. More specifically, they use an acceptance criterion based on the cost of the best solution found so far, instead of the cost of the current solution. They determine optimal cooling schedules for two small instances of the traveling salesman problem and the graph partitioning problem and found that optimal sequences of control parameter values may not be monotone. The analysis of optimal finite-time schedules is interesting, but the results obtained so far are only proved to hold for extremely small instances. At present it is not clear which impact they have on larger instances. One might argue that the whimsical structure of small instances may introduce artifacts that are absent from the more regularly structured large instances. In that case the nonmonotonicity results would only hold for a specific class of small problem instances.

9.2 Heuristic schedules

Most of the existing work on cooling schedules presented in the literature deals with heuristic schedules. We distinguish between two broad classes: static schedules and dynamic schedules. In a *static* cooling schedule the parameters are fixed: they cannot be changed during execution of the algorithm. In a *dynamic* cooling schedule the parameters are adaptively changed during execution of the algorithm. Below we present some examples.

Static cooling schedules

The following simple schedule is known as the *geometric schedule*. It originates from the early work on cooling schedules by Kirkpatrick, Gelatt & Vecchi [1983], and is still used in many practical situations.

Initial value of the control parameter. To ensure a sufficiently large value of c_0, one may choose $c_0 = \Delta f_{\max}$, where $\Delta f_{\max}$ is the maximal difference in cost between any two neighboring solutions. Exact calculation of $\Delta f_{\max}$ is quite time-consuming in many cases. However, one often can give simple estimates of its value.

Lowering the control parameter value. A frequently used decrement function is given by

$$c_{k+1} = \alpha \cdot c_k,\ k = 0, 1, \ldots,$$

where α is a positive constant smaller than but close to 1. Typical values lie between 0.8 and 0.99.

Final value of the control parameter. The final value is fixed at some small value, which may be related to the smallest possible difference in cost between two neighboring solutions.

Markov chain length. The length of Markov chains is fixed by some number that may be related to the size of the neighborhoods in the problem instance at hand.

Dynamic cooling schedules

There exist many extensions of the simple static schedule presented above that lead to a dynamic schedule. For instance, a sufficiently large value of c_0 may be obtained by requiring that the *initial acceptance ratio* χ_0 - defined as the number of accepted transitions at c_0 - is close to 1. This can be achieved by starting off at a small positive value of c_0 and multiplying it with a constant factor, larger than 1, until the corresponding value of χ_0, which is calculated from a number of generated transitions, is close to 1. Typical values of χ_0 lie between 0.9 and 0.99.

An adaptive calculation of the final value of the control parameter may be obtained by terminating the execution of the algorithm at a c_k value for which the value of the cost function of the solution obtained in the last trial of a Markov chain remains unchanged for a number of consecutive chains. Clearly such a value exists for each local minimum that is found. The length of a Markov chain may be determined by requiring that at each value c_k a minimum number of transitions is accepted. However, since transitions are accepted with decreasing probability, one would obtain $L_k \to \infty$ for $c_k \downarrow 0$. Therefore, L_k is usually bounded by some constant $L_{\max}$ to avoid extremely long Markov chains for small values of c_k.

In addition to this basic dynamic schedule, the literature presents a number of more elaborate schedules. Most of them are based on a statistical analysis of the simulated annealing process, thus allowing a more theoretical estimation of the parameters. For the transition probabilities of (4), (5), and (6), the statistical analysis leads to a model for the cost distribution that resembles an exponential distribution at low c values and a normal distribution at high c values. Within this model, the first two moments of the resulting distribution are given by Aarts, Korst & Van Laarhoven [1988]:

$$\mathbb{E}_c(f) = \mathbb{E}_\infty(f) - \frac{\sigma^2_\infty(f)}{c}\left(\frac{\gamma c}{\gamma c + 1}\right) \tag{36}$$

and

$$\sigma^2_c(f) = \sigma^2_\infty(f)\left(\frac{\gamma c}{\gamma c + 1}\right)^2,$$

where γ is given by

$$\gamma = \frac{\mathbb{E}_\infty(f) - f^*}{\sigma^2_\infty(f)}.$$

To compute $\mathbb{E}_c(f)$ and $\sigma^2_c(f)$, values for $\mathbb{E}_\infty(f)$ and $\sigma^2_\infty(f)$ can be approximated by the average cost value of the solutions and the corresponding standard deviation, respectively.

The analysis given above is used by several authors to derive adaptive parameter estimates. As an example, we discuss the schedule proposed by Huang, Romeo & Sangiovanni-Vincentelli [1986], since it is quoted in the literature as the most efficient one among those that require only a modicum of sophistication. The schedule of Lam & Delosme [1986] is conjectured to be even more efficient but its intricacy generally hinders practical use.

Initial value of the control parameter. From (36) it follows directly that $\mathbb{E}_c(f) \approx \mathbb{E}_\infty(f)$ for $c \gg \sigma^2_\infty(f)$. Hence c_0 may be chosen as

$$c_0 = K\sigma^2_\infty(f),$$

where K is a constant typically ranging from 5 to 10.

Lowering the control parameter value. Here the concept of quasi-equilibrium is quantified by requiring that the average cost difference for two consecutive Markov chains is small, i.e., $\mathbb{E}_{c_{k+1}}(f) - \mathbb{E}_{c_k}(f) = -\varepsilon$ for some small positive number ε. Next, by using

$$\frac{\partial}{\partial \ln c}\mathbb{E}_c(f) = \frac{\sigma^2_c(f)}{c}, \tag{37}$$

and replacing the left-hand side of (37) with the differential quotient, we obtain

$$\frac{\mathbb{E}_{c_{k+1}}(f) - \mathbb{E}_{c_k}(f)}{\ln c_{k+1} - \ln c_k} = \frac{\sigma^2_{c_k}(f)}{c_k}.$$

This results in a decrement rule given by

$$c_{k+1} = c_k \exp\left(-\frac{\varepsilon c'}{\sigma^2_{c_k}(f)}\right), \tag{38}$$

where, for practical purposes, $\sigma_{c_k}(f)$ is approximated by the measured deviation. In their original paper, Huang, Romeo & Sangiovanni-Vincentelli [1986] replace ε by $\lambda\sigma_{c_k}$, $\lambda < 1$, which gives only a slight modification of (38).

Final value of the control parameter. Execution is terminated if at the end of a Markov chain

$$f'_{\max} - f'_{\min} = \Delta f'_{\max}, \tag{39}$$

where $f'_{\max}$ and $f'_{\min}$ denote the maximum and minimum cost value, respectively, and $\Delta f'_{\max}$ the maximum cost difference of the solutions accepted during the generation of that chain. If (39) holds, c is set to 0, and the execution is concluded with a simple local search to ensure local optimality of the final solution.

Markov chain length. Statistical analysis leads to the observation that, in equilibrium, the fraction of solutions generated with cost values within a certain range ε from the expected cost reaches a stationary value κ. Assuming a normal distribution of the cost values, Huang, Romeo & Sangiovanni-Vincentelli [1986] show that $\kappa = -\operatorname{erf}(\varepsilon/\sigma_c(f))$, where $\operatorname{erf}(x)$ is the *error function*; see Abramowitz & Stegun [1970]. The Markov chain length is determined by the number of trials L_k for which

$$L_k^* = p\kappa,$$

where p is a parameter depending on the size of the problem instance, and L_k^* is defined as the number of accepted solutions with a cost value within the interval $(\mathbb{E}_c - \varepsilon, \mathbb{E}_c + \varepsilon)$. An additional bound on L_k is introduced to avoid extremely long Markov chains.

10 ISSUES FROM PRACTICE

Four basic ingredients are needed to apply simulated annealing in practice: a concise problem representation, a neighborhood function, a transition mechanism, and a cooling schedule. The algorithm is usually implemented as a sequence of homogeneous Markov chains of finite length, generated at descending values of the control parameter specified by the cooling schedule. As for the choice of the cooling schedule, we have seen in the previous section that there exist some general guidelines. However, no general rules are known that guide the choice of the other ingredients. The way they are handled is still a matter of experience, taste, and skill left to the annealing practitioner, and we expect that this will not change in the near future.

During the years of its existence, simulated annealing has been applied to a large variety of problems, ranging from practical real-life situations to theoretical tests. Two appealing examples of real-life applications are the scheduling of the Australian state cricket season by Willis & Terrill [1994] and the design of keyboards for typewriters by Light & Anderson [1993]. VLSI design, atomic and molecular physics, and picture processing are the three problem areas in which simulated annealing is most frequently applied. The set of theoretical test problems includes almost all the well-known problems in discrete mathematics and operations research, such as coding, graph coloring, graph partitioning, sequencing and scheduling problems; see the references given below and Chapters 8 through 13 of this book.

So far, about a thousand papers have been published reporting applications of the algorithm. Many of these studies have led to modifications of the algorithm such as the use of penalty functions, alternative generation and acceptance probabilities, implementation-specific aspects, parallel versions, etc. Due to the large variety of approaches and the many different implementation details, it is virtually impossible to give a balanced overview of the experience that has been gathered. Therefore, we restrict ourselves to some general statements and appropriate references.

We start with the references. General overviews of applications of simulated annealing are given by Aarts & Korst [1989a], Collins, Eglese & Golden [1988], Dowsland [1993], Van Laarhoven & Aarts [1987], and Vidal [1993]. Overviews of applications in operations research are given by Eglese [1990] and Koulamas, Antony & Jaen [1994]. Studies emphasizing performance issues for theoretical test problems are given by several authors. One of the most elaborate studies is presented by Johnson et al. [1989, 1991], who report on an extensive numerical study for several combinatorial optimization problems, including graph partitioning, graph coloring and number partitioning. This work provides many practical findings that in our opinion reflect the general experience of annealing practitioners.

Perhaps the most striking element is the observed performance ambivalence. For the graph partitioning problem, simulated annealing seems to outperform all existing approximation algorithms, whereas for the number partitioning problem the performance is hopelessly poor. Although this bad performance for the number partitioning problem can be understood from analytical arguments, there seems no way to adapt the algorithm in order to improve it. A similar ambivalence is encountered in the area of code design, where for the football pool problem, simulated annealing is able to improve on the best known results; see Van Laarhoven et al. [1989] and Chapter 13 of this book. On the other hand, Beenker, Claasen & Hermens [1985] found that for problems related to the design of binary sequences, the algorithm is inferior to simple constructive methods.

Furthermore, the literature presents results of studies in which the performance of simulated annealing is compared with that of other local search algorithms. Results for the job shop scheduling problem are presented by Van Laarhoven, Aarts & Lenstra [1992], Aarts et al. [1994], and Vaessens, Aarts & Lenstra [1996], and for the traveling salesman problem by Ulder et al. [1991].

With some restraint one may conclude from these studies that simulated annealing, if large running times are allowed, can outperform about all other algorithms with respect to effectiveness. More general conclusions cannot be drawn due to the many different quality measures that can be applied. To illustrate this we mention the performance studies for quadratic assignment. Pardalos, Murty & Harrison [1993] report that simulated annealing can find acceptable solutions with fewer iterations than tabu search. Battiti & Tecchiolli [1994] question this conclusion and argue that it no longer holds for difficult problem instances if high quality solutions are required.

Broadly speaking, simulated annealing can find good solutions for a wide variety of problems, but often at the cost of substantial running times. Consequently, the true merits of the algorithm become obvious in industrial problem settings, where running times are of little or no concern. As an example we mention design problems, since in those cases one is primarily interested in finding high-quality solutions, whereas design time often plays only a minor role. A well-known successful simulated annealing area in this respect is VLSI design [Sechen & Sangiovanni-Vincentelli, 1985; Shahookar & Mazumder, 1991; Wong, Leong & Liu, 1988].

The success of simulated annealing can be explained from the fact that the algorithm is easy to implement and capable of handling almost any optimization problem and any constraint, either by appropriate neighborhoods or by relaxation through the use of penalty functions. These properties are, however, not unique for simulated annealing. They also hold for simple local search algorithms. The main advantage of simulated annealing is that it is able to improve upon the relatively poor performance of local search by simply replacing the deterministic (strict improvement) acceptance criterion by a stochastic criterion, thus circumventing the need of an in-depth study of the problem structure in order to construct more effective neighborhoods, or to design more tailored algorithms. It almost goes without saying how this is a great advantage in an industrial environment, since often the required expertise is unavailable and, even more important, it saves development time.

11 SPEEDING UP

The literature presents many variations on the basic simulated annealing approach presented in the previous sections. Many of these variations concentrate on alternatives that should reduce the potentially burdensome running times required by simulated annealing to converge to near-optimal solutions. Roughly speaking, the existing approaches fit into three categories: fast sequential algorithms, hardware acceleration, and parallel algorithms. We mention a few examples.

Szu & Hartley [1987] present an annealing algorithm for the optimization of continuous-valued functions, using a generation mechanism given by a Cauchy distribution instead of the frequently used Gaussian distribution. They claim that their generation mechanism leads to an inverse linear cooling rate, instead of an inverse logarithmic cooling rate as was found for the Gaussian distribution. This approach has been further refined by Ingber [1989], who proposes a technique he calls *very fast simulated reannealing*, permitting an exponential cooling rate. The application of these approaches is limited to the optimization of continuous real-valued functions, which prohibits their use in the many existing combinatorial optimization problems.

Greene & Supowit [1986] introduce the *rejectionless method* as an example of a deterministic simulated annealing approach based on an improved generation mechanism. They propose to generate new solutions with a probability proportional to the effect of a transition on the cost function. In this way, a subsequent solution is directly chosen from the neighborhood of a given solution, i.e., no rejection of solutions takes place. This method leads to shorter Markov chains for a number of problems. However, the efficient use of the method depends strongly on some additional conditions on the neighborhood function, which unfortunately cannot be met by many combinatorial optimization problems. Fox [1993, 1994] further elaborates on this issue. He introduces the concept of *self-loop elimination* and shows how it not only speeds up simulated annealing, but also causes the algorithm to be more efficient than multistart iterative improvement with random restarts. This contradicts Ferreira & Žerovnik [1993], who asserted the opposite.

Parallel simulated annealing algorithms aim at distributing the execution of the various parts of a simulated annealing algorithm over a number of communicating parallel processors. This is a promising approach to the problem of speeding up the execution of the algorithm, but it is by no means a trivial task, due to the intrinsic sequential nature of the algorithm. Over the years a large variety of approaches have been proposed, leading to algorithms that are generally applicable and to tailored algorithms. For overviews we refer to Aarts & Korst [1989a], Azencott [1992], Boissin & Lutton [1993], Greening [1990], and Verhoeven & Aarts [1996]. A special approach to parallel simulated annealing is provided by the use of neural network models. To this end, the optimization problem at hand is cast into a 0-1 programming formulation and the values of the decision variables are associated with the states of the neurons in the network. This has led to randomized approaches such as the *Boltzmann machine* [Aarts & Korst 1989a, 1989b; Aarts & Korst, 1991], and to deterministic approaches such as the *mean field method* [Peterson & Söderberg, 1989]; see also Chapter 7 of this book. In addition to the speedup obtained by parallel execution, neural networks also offer a speedup through their hardware implementation. This has led to fast VLSI implementations of simulated annealing [Lee & Sheu, 1991] and even to optical implementations [Lalanne et al., 1993].

12 COMBINED APPROACHES

Recent approaches to local search concentrate on the combined use of different local search algorithms, known as *multilevel approaches* [Vaessens, Aarts & Lenstra, 1992]. Simulated annealing is used in several of these approaches, and we mention some examples. Martin, Otto & Felten [1991, 1992] propose a successful simulated annealing algorithm for the traveling salesman problem, which uses a restricted 4-exchange neighborhood, combined with a simple local search algorithm using a 3-exchange neighborhood. Eiben, Aarts & Van Hee [1991] present a stochastic search procedure that combines elements of population genetics with those of simulated annealing. They prove that their stochastic approach exhibits convergence properties similar to those of simulated annealing. Lin, Kao & Hsu [1994] introduce a genetic approach to simulated annealing using population-based transitions, genetic-operator based quasi-equilibrium control, and Metropolis-criterion selection operations in the jargon of genetic algorithms. They find empirically that their approach works quite well for the zero–one knapsack, set partitioning, and traveling salesman problems.

Clearly, the issue of combined approaches opens many possibilities for the design of new variants of local search algorithms. However, one should be careful not to propose these variants as new algorithmic concepts. Research on local search has been fascinating over the past 10 years. It has also suffered from considerable confusion, created by so-called new concepts, which, after their fancy names had been demystified, turned out to be only coarse or well-known heuristic rules.