Bouncing towards the optimum: Improving the results of Monte Carlo optimization algorithms

Johannes Schneider* and Ingo Morgenstern
Fakultät Physik, Universität Regensburg, Universitätsstrasse 31, D-93053 Regensburg, Germany

Johannes Maria Singer
Physikinstitut, Universität Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland

(Received 22 September 1997)

Simulated annealing and related Monte Carlo-type optimization algorithms are used to apply statistical physics concepts, in particular ideas from the statistical mechanics of spin glasses, to find optimal configurations for combinatorial optimization problems. There are formal proofs showing that these algorithms converge asymptotically (i.e.—possibly—for infinitely long simulation times) to a global optimum. Practical implementations, however, only allow for finite simulation times, and, thus, the annealing process is often trapped in energetically higher, suboptimal configurations. In this work we present an algorithm—we call it bouncing—which takes the final low-energy configuration of, e.g., a conventional monotonically cooled annealing run as an input and subjects it to a schedule of repeatedly reheating and cooling. The maximum of a susceptibility and a specific-heat-like quantity sampled during the initial monotonic cooling process serve as lower and upper starting temperature bounds for this secondary heating and cooling. We present, in addition to a serial implementation, a recipe for a parallel computer, and provide a number of results showing the success of the bouncing method for a particularly prominent example of a combinatorial optimization problem: the traveling salesman problem. [S1063-651X(98)04707-2]

PACS number(s): 02.70.Lq, 02.50.−r, 02.50.Ga, 89.20.+a

I. INTRODUCTION

There are two possible approaches to solving combinatorial optimization (CO) problems: Either one can use an optimization algorithm in the strict sense of the word, yielding a globally optimal solution, or an approximation algorithm, yielding a very good or *possibly* globally optimal solution. A classical method to find low-energy states of complex physical systems such as solids is to heat the system up to some high temperature, where all states can be reached, and then cool it down slowly. This annealing process lets the system settle into regions of low energy, while not becoming trapped in higher-lying local minima. Simulated annealing (SA), a Monte Carlo-type algorithm, is an optimization technique that uses these methods from statistical physics. It was first employed successfully for these problems in Refs. [1,2]. The work of Kirkpatrick *et al.* [1] is strongly based on this analogy between the annealing of a solid and the optimization of a system with many independent variables. In this technique the states of a physical system σ_i are generalized to states of a system being optimized, the energy $\mathcal{H}$ in physical problems is generalized to the objective function to be optimized, and the temperature T is generalized to a control parameter for the optimization process. To apply the simulated annealing algorithm, a mechanism is used to generate a new configuration (i.e., a new state) from a given one by a small perturbation, i.e., by applying a set of moves to the system. The ''size'' of a class of moves within the state space of the system can be defined as the average change in the energy induced by moves of that class [3], and the decision of accepting or rejecting the moves is based on a generalized Metropolis criterion. Depending on the control parameter this method accepts deteriorations of the system, i.e. configurations that correspond to an increase in the energy or cost function. There are formal proofs from a number of authors [4–6] that under certain conditions the algorithm converges asymptotically (i.e., for long enough, maybe infinitely long, simulation times) to an optimal solution. Thus, asymptotically, the algorithm is not only an approximation, but also an optimization algorithm in the strict sense. In practical applications, however, this kind of asymptoticity is usually never attained, and, thus, an optimal solution cannot be reached within a given simulation time due to the extremely slow convergence. Consequently, in practice, SA usually appears to be an approximation algorithm only. For a discussion on the convergence of SA, we refer the reader to Ref. [7].

II. SIMULATED ANNEALING: ALGORITHM

To explore the state space, instead of doing an unguided random walk in state space, known as simple sampling, in SA a sequence of Markov chains of states is constructed where each state σ_{i+1} is repeatedly generated out of the previous state σ_i with a suitable transition probability $p(\sigma_i \rightarrow \sigma_{i+1})$ depending on the difference of their energies

$$\Delta\mathcal{H} = \mathcal{H}(\sigma_{i+1}) - \mathcal{H}(\sigma_i). \tag{2.1}$$

With this choice, the distribution function $\pi(\sigma_i)$ of the states tends toward the Boltzmann equilibrium distribution

*Author to whom correspondence should be addressed. FAX: +49 941 943 1968. Electronic address: Johannes.Schneider@physik.uni-regensburg.de

$$\pi_{\text{equ}}(\sigma_i)=\frac{1}{Z}\exp\left(-\frac{\mathcal{H}(\sigma_i)}{k_BT}\right), \tag{2.2}$$

with the Boltzmann constant k_B [8], the system temperature T, and the physical partition function Z [9]. A sufficient condition is the principle of detailed balance,

$$\pi_{\text{equ}}(\sigma_i)p(\sigma_i\to\sigma_j)=\pi_{\text{equ}}(\sigma_j)p(\sigma_j\to\sigma_i). \tag{2.3}$$

The most common choice for p is

$$p(\sigma_i\to\sigma_{i+1})=\begin{cases}\exp(-\Delta\mathcal{H}/T) & \text{if } \Delta\mathcal{H}>0\\ 1 & \text{otherwise,}\end{cases} \tag{2.4}$$

which satisfies the condition of detailed balance. This acceptance criterion allows for intermediate deteriorations in the optimization objective (i.e., an increase of the energy), thus enabling the algorithm to escape from local minima depending on the value of the control parameter temperature T.

A variation of the above principle is called threshold accepting (TA): here a control parameter called the threshold T is introduced much in the same way as a pseudotemperature, and p can be written as

$$p(\sigma_i\to\sigma_{i+1})=\begin{cases}1 & \text{if } \Delta\mathcal{H}\leq T\\ 0 & \text{otherwise.}\end{cases} \tag{2.5}$$

Although TA does not satisfy the condition of detailed balance, it leads to very low-energetic minima or even the ground state, usually in shorter simulation time than simulated annealing [10].

Among other criteria the quality of the final solution obtained by the algorithm is determined by the convergence of the algorithm, which is controlled by a set of parameters, called the cooling schedule. In the literature [11] the behavior of the simulated annealing algorithm as an approximation algorithm is usually analyzed in an empirical way. This involves the analysis of computation time and quality of final solutions obtained by running the algorithms on a set of problem instances, e.g., from Reinelt's TSPLIB [12], and the comparison of the results to those obtained by other methods.

Commonly, one refers to an implementation of the SA algorithm in which a sequence of Markov chains of finite length is generated at monotonically decreasing values of the temperature (usually called ''sequential SA'' in the literature, e.g., [7]). Optimization is carried out by starting off at an initial value of the control parameter, and repeatedly generating a Markov chain for decreasing values of T until $T\to 0$. Ideally, the system is allowed to approach equilibrium at each temperature level; then the temperature is reduced, the system is allowed to equilibrate again, and so on. The procedure is stopped at a temperature low enough that no further improvements can be expected. This protocol is governed by the cooling schedule, with an appropriate starting value T_0, a decrement prescription for the control parameter, and a finite length of the individual chains.

Since the early publications on simulated annealing an exponential cooling schedule has been often proposed,

$$T=T_0a^t, \tag{2.6}$$

with an initial temperature $T_0>0$ and the cooling factor $0<a<1$. For finite simulations, this schedule is believed to be an excellent cooling recipe, since it provides a rather good compromise between a computationally fast schedule and the ability to reach low-energy states. Catoni [13] reported that suitably adjusted exponential cooling schedules have interesting robustness properties as soon as one deals with a finite amount of available computing time. Almost all practical uses of SA, as documented in the literature, rely on exponential cooling schedules, and it is our method of choice for SA and threshold accepting too. Several other decrement rules for the control parameter temperature can be found in the literature; see, e.g., Ref. [14]; another useful source is the detailed study by Bonomi *et al.* for the traveling salesman problem [15].

We are interested in the value of the energy or cost function [8] at a temperature T,

$$\langle\mathcal{H}\rangle_T=\frac{1}{M}\sum_{i=1}^{M}\mathcal{H}(\sigma_i), \tag{2.7}$$

where the summation goes over a sequence of M independent configurations σ_i, and a quantity C_T, which we associate in analogy to the statistical physics equivalents with a specific heat

$$C_T=\frac{\partial\langle\mathcal{H}\rangle_T}{\partial T}. \tag{2.8}$$

It is straightforward to show that this specific heat, defined as the derivative of the energy with respect to temperature T, is proportional to the variance of the energy

$$C_T\equiv\frac{1}{T^2}\{\langle\mathcal{H}^2\rangle_T-\langle\mathcal{H}\rangle_T^2\}=\frac{1}{T^2}\text{var}_T\{\mathcal{H}\}. \tag{2.9}$$

This particular identity is strictly valid only for SA; nevertheless we also adopt this definition of a (pseudo) specific heat measuring the fluctuations for TA.

As test instances for CO problems, we focus on traveling salesman problems (TSP's), since most other examples of CO problems can be easily transformed into a formulation quite similar to a TSP with some additional constraints. Since a TSP is defined via entries in a distance matrix, an asymmetric overlap η is generally taken as a substitute for an order parameter like, e.g., the magnetization in Ising problems. To calculate η, an edge matrix e is constructed from the ground state permutation γ of the N cities as follows:

$$e(k,l)=\begin{cases}1 & \text{if } \gamma^{-1}(l)-1=\gamma^{-1}(k) \mod N\\ -1 & \text{if } \gamma^{-1}(k)-1=\gamma^{-1}(l) \mod N\\ 0 & \text{otherwise,}\end{cases} \tag{2.10}$$

with k and l being city numbers, and γ^{-1} being the inverse permutation of γ. Therefore, $e(k,l)=1$ if l is the successor of k in γ and $e(k,l)=-1$ if l is predecessor of k. Let σ_i be the current state, i.e., the actual permutation of the cities, and j a counter; then

$$\eta(\sigma_i)=\left|\frac{1}{N}\sum_{j=0}^{N-1} e[\sigma_i(j),\sigma_i(j+1)]\right|, \qquad (2.11)$$

with $\sigma_i(0)=\sigma_i(N)$. A susceptibility χ_T, corresponding to η, is defined as

$$\chi_T=\frac{1}{T}\{\langle \eta^2\rangle_T-\langle \eta\rangle_T^2\}=\frac{1}{T}\mathrm{var}_T\{\eta\}, \qquad (2.12)$$

which can be achieved for SA by formally adding a term $\mathcal{H}_1=-\lambda\eta$ to the Hamiltonian $\mathcal{H}$, with $\lambda\to 0$, and deriving $\langle\eta\rangle_T$ for λ. For TA, we simply adopt definition (2.12).

A well-known TSP instance is the benchmark problem to find the shortest tour through 442 drilling holes in an IBM printed circuit board. It was proposed by Grötschel and solved by Holland [16], and it is usually referred to as PCB442, e.g., in Reinelt's library of TSP instances (TSPLIB) [12]. Its optimum has a length of 50 783.5475... in REAL*8 metric [17] (which is used throughout this paper), or of 50 778 in the rounded integer metric [12]. This problem has a highly degenerate ground state [17], and there is a very large amount of nearly optimal (i.e., approximate) solutions energetically close to the global optima. Moreover, it is known that there are many optimization algorithms which suffer from severe difficulties treating this problem. We restrict ourselves in the following discussion on the PCB442. Although, intuitively, it seems to be very easy to solve at a first glance, PCB442 has a very complicated search space structure, and shows properties which are well known from other physical problems.

Because of the highly degenerate ground state of PCB442, one has to adapt the calculation of η. As shown in an earlier reference [17] a complete set of degenerate ground states of this problem can be built out of 84 pieces (''backbones''). Thirty-four of these consist only of one city, so that 50 backbones of length $l>1$ remain. If we fix the direction of a long backbone, we automatically determine the directions of all other backbones with the exception of 18 ''blinkers,'' each of them consisting of only two cities (see Fig. 1). These 32 backbones form one ''superbackbone'' consisting of 340 edges, out of which an edge matrix e is built. Furthermore, the 18 blinkers are explicitly considered in a symmetric edge matrix f, leading us finally to

$$\eta(\sigma_i)=\frac{1}{358}\left(\left|\sum_{j=0}^{N-1} e[\sigma_i(j),\sigma_i(j+1)]\right| + \sum_{j=0}^{N-1} f[\sigma_i(j),\sigma_i(j+1)]\right). \qquad (2.13)$$

To generate an appropriate neighborhood structure in search space, we use moves of the Lin-Kernighan-type [18] lin-n-opt, mostly lin-2-opt and lin-3-opt.

Finite Markov chain length, SA-like Monte Carlo (MC) optimization algorithms cannot guarantee that the global minimum of a combinatorial optimization problem is reached. On cooling, the system would probably be trapped in a suboptimal configuration because of the rough energy landscape of the problem.

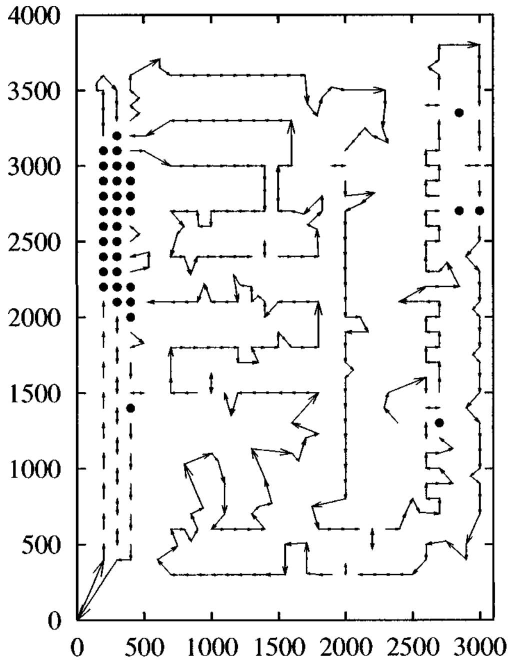

FIG. 1. 84 ''backbones'' of the PCB442 problem. Out of these backbones all optimal solutions can be constructed. We arbitrarily fixed the direction of one of the long backbones which, in turn, automatically determines the directions of most of the other backbones (with the exception of 18 blinkers, as described in the text). All lengths (i.e., energies) of the TSP instances are measured in arbitrary, dimensionless units (a.u.), which makes most expressions dimensionless [8].

There are different approaches to ''free'' a simulation frozen in at a suboptimal valley in the energy landscape, and thus to obtain better results (i.e., results closer to or identical with the global optimum). One method in this category calls for a very complex set of cleverly constructed moves in order to give the system more degrees of freedom; this leads to a largely extended neighborhood of a certain state, and is believed to provide ''escape routes'' in state space where otherwise the MC walker would become stuck in a suboptimal energy valley. This can easily be seen at the traveling salesman problem: It has been shown that with the lin-2-opt, which changes the direction of a part of the tour, one usually obtains better results than with a simple move exchanging two points in a solution [19]. An already more complicated move is the $n=3$ lin-3-opt. There are four possibilities to perform a lin-3-opt: the most effective one is to exchange a part of the tour with its succeeding part without changing their directions; at least three lin-2-opt moves are needed to build the same configuration out of the previous one, whereas the other three versions of the lin-3-opt can be constructed with two lin-2-opts only. Further higher order lin-n-opts are possible, but already lin-4-opts do not have a significant influence on the results because the number of trial moves has to be systematically increased in such a way that the higher order moves can find improvements at low temperatures which are not found with the described low n moves. We should mention that this method can be useful for specific instances, but the construction of an appropriate combination of moves is unfortunately highly problem specific. As a result, it has not been widely used.

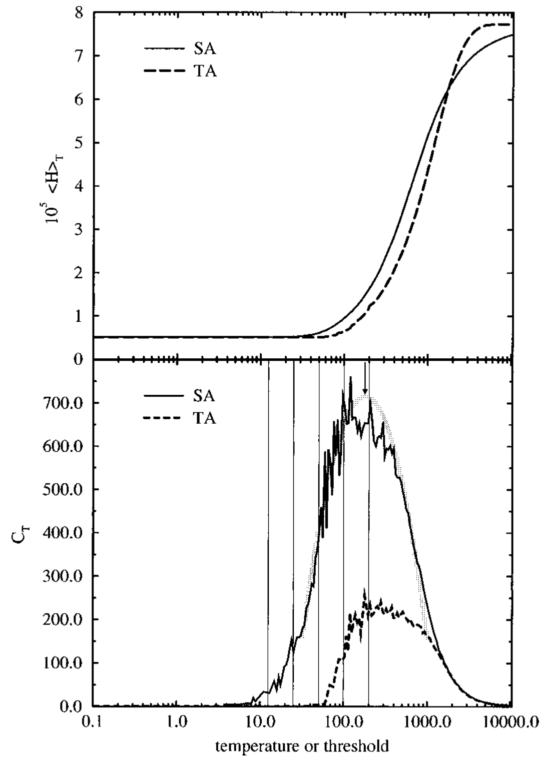

FIG. 2. Energy (i.e., length) $\langle \mathcal{H} \rangle_T$ (upper part) and specific heat C_T (lower part) vs T for simulated annealing (SA, solid line) and threshold accepting (TA, dashed line) PCB442 traveling salesman problem. The vertical lines in the C_T curve mark the five temperature values used as ''bouncing temperatures'' T_B in this paper. The gray curve is a least square fit of C_T^{SA} to a parabola, with an arrow marking the maximum.

Other approaches, like, e.g., ''basin hopping'' [20] or ''search space smoothing'' [21,22], which belong to the family of ''hypersurface deformation'' methods, can be used to try to transform a complex rough energy landscape into a smoother surface with fewer minima, which allows an efficient relaxation to the global minimum. Such ''smooth'' surfaces are ideally characterized by only one deep ''funnel'' leading to the global optimum, and any optimization method should find surfaces with a single funnel relatively easy to tackle. The difficulty of this technique is the choice of an appropriate mapping, which relates the global minima of the original and transformed surfaces.

The idea which we present in the following sections is based on a different approach. To discuss it in depth (see Sec. III), we first analyze a sequential SA/TA run for a TSP: In Fig. 2, we provide the energy (i.e., tour length) $\langle \mathcal{H} \rangle_T$ and the specific heat C_T vs temperature (threshold) both for SA and TA. This figure shows a typical annealing curve for these types of problems, many of which have been studied and show similar curves, and each has similar overall features. If we look at the energy decrease as a function of decreasing temperature, we see that for TA the length is roughly constant at high thresholds. Because the TA criterion allows one to accept every move at large T, each edge has the same probability to appear in the actual configuration, and therefore the energy oscillates around the value

$$\langle \mathcal{H} \rangle_\infty = \frac{1}{N-1} \sum_{i,j=1}^{N} d(i,j), \tag{2.14}$$

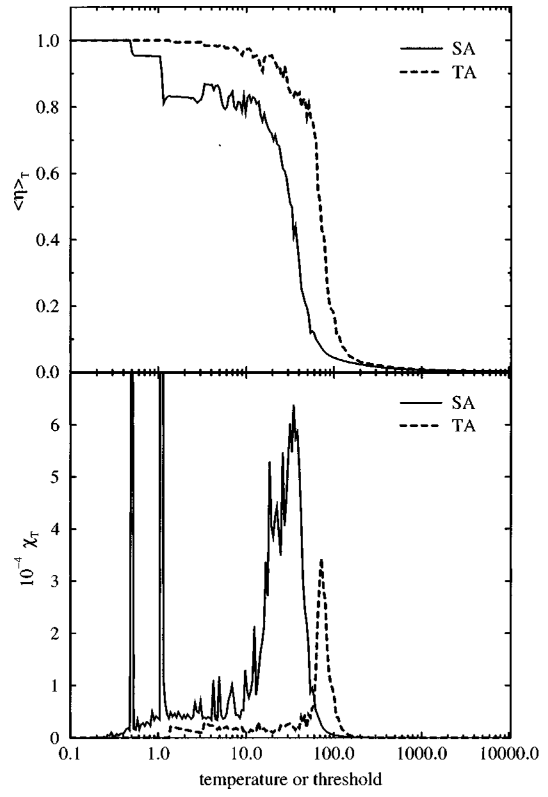

FIG. 3. Asymmetric overlap $\langle \eta \rangle_T$ with the ground state and corresponding susceptibility χ_T vs T for simulated annealing (SA, solid line) and threshold accepting (TA, dashed line) PCB442.

with the distance matrix d. At high but finite temperatures, SA does still not allow deteriorations of any magnitude, and thus the length at corresponding temperatures is smaller than in TA. Certainly there must be an energy maximum, since the system is finite, and some configuration provides a finite upper bound on the energy. Beyond this, in the infinite-temperature limit, any attempted move will be accepted [3].

Lowering the temperature, there is a characteristic transition into a low-energy regime; the TA energy decrease is steeper than in the run with SA, so that both curves cross each other. There is also a finite global minimum because the system is finite; the values in $[d(i,j)]$ are finite, and, since the tour is a spanning tree with an additional edge to close it, therefore each configuration has a larger length than the minimum spanning tree.

In the affiliated specific heat C_T we find a peak with the maximum located at a temperature T_f, in case of TA the absolute height of the peak is lower than for SA. Moreover, the TA peak appears to be at a higher value than the SA peak, which is located at a temperature $T_f \approx 180$ for SA. In the specific heat figure, we include a nonlinear least square fit for $C_{T,\mathrm{SA}}$. The peak of the specific heat (the arrow in Fig. 2 for SA) marks the temperature range where the most important rearrangements (in terms of the proportionality specific heat $\propto$ variance) of the system happen. Stretching the previous analogy between solid state and statistical physics and the MC algorithms, we naively associate the peak with a transition region between a partially ordered solid, where rearrangements up to a certain scale are still possible, and a totally disorderd liquidlike regime.

Figure 3 provides the asymmetric overlap η and the corresponding susceptibility vs temperature (threshold), both for SA and TA. These plots show a typical behavior for an order parameter: η increases from 0 in the high-temperature regime to 1 for low temperatures in a narrow transition range, with a characteristic spontaneous $\pm$ symmetry breaking between the two possible tour orientations (backward-forward traveling). It is nonzero only in the ordered phase. The affiliated susceptibility has its peak at a temperature $T_c \approx 32$ for SA, much lower than T_f. The peak of the susceptibility gives the temperature range where the system tries to anneal due to the additionally imposed constraint given by $\mathcal{H}_1$. It is the range where it converges against the bottom of the valley in which the ground state lies. In this sense, it may be regarded as a classical susceptibility according to the definition of Ref. [23], measuring the system response to constraint $\mathcal{H}_1$.

We are aware that, until now, there has not been stringent evidence of a critical temperature in precise phase transition terminology, as noted earlier in Refs. [24,25] (There, considerable evidence for such a critical temperature below the specific heat peak was deduced from spin glass analogies.) Nevertheless, we would like to point to the fact that all quantities—energy, specific heat, overlap (which is a kind of magnetization of the system), and susceptibility—are plotted with a logarithmic temperature axis, which gives a ''natural looking'' figure from the point of view of phase transition physics, e.g., in the field of spin glasses [24].

If we carry on the initial physical annealing picture we might be able to improve the basic algorithm considerably. The idea arises naturally if we look at the techniques of a blacksmith; after cooling down the molten iron rather quickly into a particular shape, the craftsman uses a sophisticated heat treatment schedule consisting of repeatedly reheating and cooling the workpiece. This cyclic reheating is done only up to a certain temperature (e.g., red glow), and then the piece is either slowly cooled or quickly quenched. Depending on the particular treatment, the craftsman is able to produce the desired major structural rearrangements in the workpiece, without ''crossing'' back over the phase transition line from the solid to the totally disordered liquid.

In Sec. III, we will show how this ''craftsman's approach'' to the annealing type algorithms can be used to improve finite length MC optimizations. In tribute to this blacksmith picture we call the method ''bouncing.''

III. BOUNCING

Due to the fact that finite length Markov chains lead to an approximation algorithm only, as stated above, a simulation often does not reach the global minimum of an energy landscape. Instead, the system ''freezes'' in an energetically high-lying minimum at a small value of the control parameter (temperature or threshold), i.e., the system is no longer able to find a sufficient move to ''escape'' from a local energy valley.

On the other hand, there is a basic finding of Romeo and Sangiovanni-Vincentelli [26]: They observed that, in order to obtain a final configuration close to the globally minimal one, there should always be a sufficiently large probability to leave any configuration, possibly a local minimum, found during the execution of the algorithm.

Finally, there is the further conceptual ingredient that the specific heat has a maximum at a value T_f of the control parameter, corresponding to a freezing temperature between a disordered high-temperature and a partially (i.e., locally) ordered low-temperature regime, and the order parameter susceptibility has a peak at T_c, which corresponds to a transition to a long-range ordered regime. As mentioned above, we do not use the term transition temperature in the strict sense of a physical phase transition theory. For that purpose, further investigations on the nature of the particular transition are necessary.

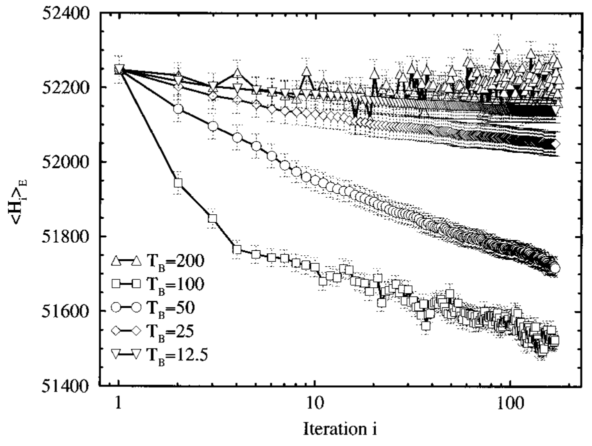

FIG. 4. Ensemble mean of the energy, $\langle \mathcal{H}_i \rangle_E$ vs index i of the bouncing iteration for different bouncing temperatures T_B, TA, and PCB442.

Using these basic informations we come to the following algorithm. Instead of using a cooling schedule with a monotonically decreasing value of the control parameter only, we add a second part to the prescription: After a first conventional simulation, where the temperature is lowered monotonically ($T \rightarrow 0$), according to some of the above-mentioned schedules, e.g., the exponential one, the simulation reaches a point where no further decrease of the energy can be achieved within a finite simulation time. We call this first part ''primary monotonic cooling.'' At this point we instantaneously increase the value of the control parameter again (''inverse quench'') up to a value T_B, virtually similar to the blacksmith in the above cited technical analogy. Thereby, the system regains its ''ability to move'' in state space, depending on the new value of the control parameter ($T_B \geqslant T \geqslant 0$). The main advantage of this second stage is that—in contrast to the previous start of the simulation at a very high temperature in a totally disordered state—the system has already gained a high degree of order. Connected with this is a certain degree of remaining information about the system stored in the ordered configuration. The question now is the following: How far should this second ''heat-up'' go, i.e., what is the appropriate starting temperature T_B for the bouncing process schedule $T \rightarrow 0 \rightleftharpoons T_B$? The local order information must be retained, as few as possible of the structures should be ''broken'' up.

We can distinguish three temperature regimes for T_B (Fig. 4), which we will discuss in Sec. IV using a set of representative simulations of the PCB442 TSP sample instance.

(1) Slight warming: the reheating after the first monotonic cooling will be only very slightly, i.e. in a temperature win-

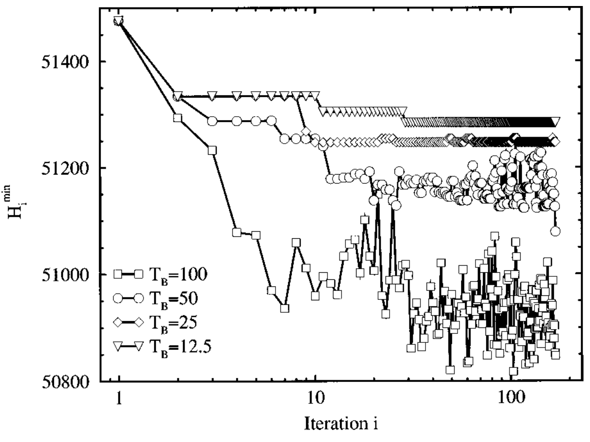

FIG. 5. Minimal energy $\mathcal{H}_i^{\rm min}$ vs index i of the bouncing iteration for different bouncing temperatures T_B, TA, and PCB442.

dow below the temperature where the initial ordering transition took place: $T_B \leqslant T_c$.

(2) Bouncing to the C_T maximum: the reheating goes back up to (almost) the freezing temperature, $T_c < T_B < T_f$. It covers the whole temperature range of the partially ordered phase, but avoids the transition into the totally disordered regime, similar to the craftsman, who reheats his workpiece to gain maximal structural rearrangements without leaving the solid phase and thus losing the shape of the piece.

(3) Bouncing beyond T_f: finally, we will show the effects of a bouncing process crossing T_f, i.e., $T_B > T_f$.

Technically, for the bouncing process a certain number of reheating and cooling iterations are performed after the first monotonic cooling, and at the end of each iteration $0 \uparrow T_B \searrow 0$, the actual low-temperature configuration serves as an initial configuration for the next bouncing iteration.

Figure 4 shows five regimes for T_B, T_B=12.5, 25, 50, 100, and 200, as indicated by vertical lines in the specific heat graph (Fig. 2). The value T_B=200 is the only one beyond the T_f transition. We present the ensemble mean energy $\langle \mathcal{H}_i \rangle_E$ (i.e., the mean TSP tour length) in Fig. 4, and the minimum energy $\mathcal{H}_i^{\rm min}$ in Fig. 5, of an ensemble of 128 statistically independent TA optimization runs. Each run starts initially with different random start configurations. Each one starts with an initial monotonic cooling at T =1000, and continues with bouncing iterations using different starting values of T. The technique used is TA. After conventional monotonic cooling, more than 100 bouncing iterations are performed. There the system is instantaneously heated up to the particular T_B, and then the temperature is lowered with an exponential cooling schedule from the indicated T_B value by a factor of 0.9 in 120 discrete steps. In each step 240 lin-2-opt sweeps and 1200 lin-3-opt sweeps are performed. Finally, at the end of each iteration, a ''greedy'' step is performed to reach the bottom of the local energy valley, and the obtained configuration is taken again as an initial input for the next iteration. We find the following results.

(1) $\langle \mathcal{H}_i \rangle_E$ decreases with an increasing number of iterations if $T_B \leqslant 100$, i.e., $T_B \leqslant T_f$.

(2) If $T_B \leqslant T_c$, the mean value of the energy of succeeding iterations is roughly constant, except for seldom fluctuations. This can be nicely seen in the graphs for T_B=12.5 and 25. The system is therefore trapped in a local valley. There is certainly a dependency of the bouncing results on the resulting configuration of the initial cooling process, which does not disappear for $T_B \leqslant T_c$.

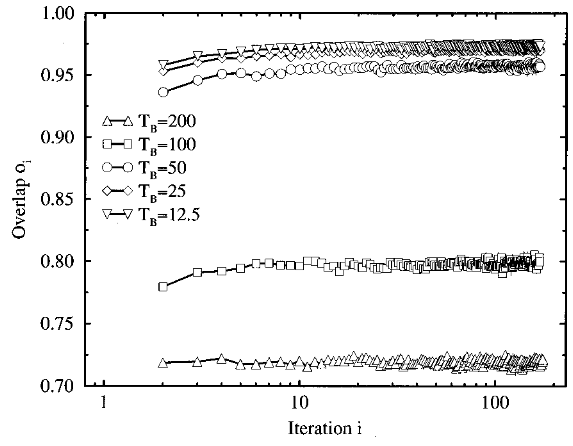

FIG. 7. Overlap o_i between the resulting configurations of successive bouncing iterations i and $i-1$ for different bouncing temperatures T_B, TA, and PCB442 (the error bars are of the size of the symbols).

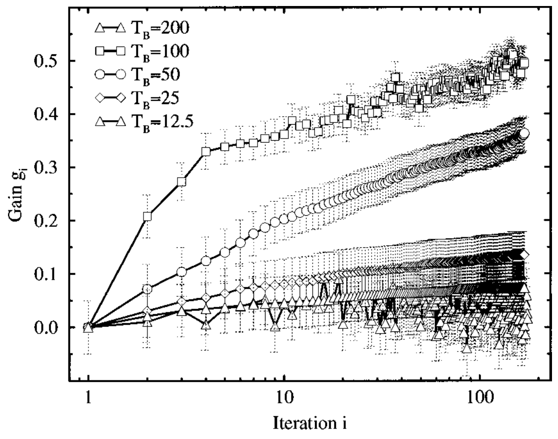

FIG. 6. Gain g_i vs the index i of the bouncing iteration for different bouncing temperatures T_B, TA, and PCB442.

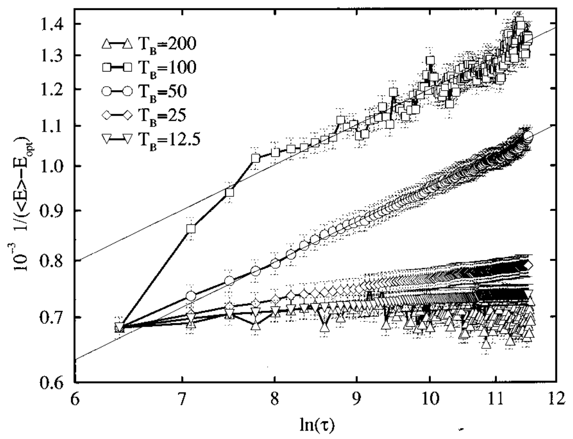

FIG. 8. Grest hypothesis: $1/(\langle \mathcal{H}_i \rangle_E - \mathcal{H}_{\rm opt})$ vs ln $(i\tau)$ for different bouncing temperatures T_B, TA, and PCB442. The thin solid lines are fits, giving an exponent ζ=0.8, both for T_B=100 and T_B=50.

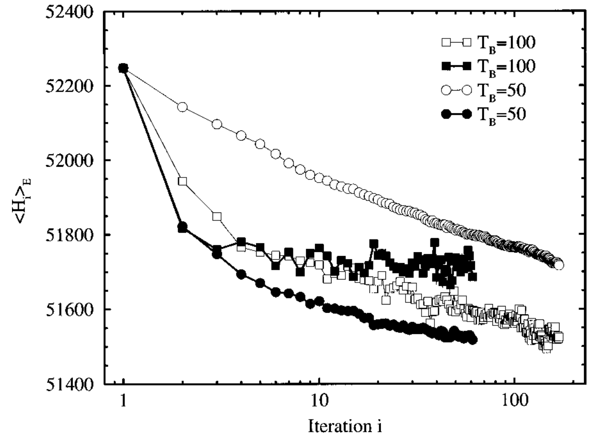

FIG. 9. Ensemble mean of the energy, $\langle\mathcal{H}_i\rangle_E$ vs index i of the bouncing iteration for $T_B=100$ (squares) and $T_B=50$ (circles), TA, and PCB442. Filled symbols denote simulation runs with a doubled calculation time in contrast to the originally obtained data (open symbols).

(3) For $T_B>T_f$ we find no decrease of the energy as a function of bouncing iterations; ''bouncing beyond the ordering temperature'' causes oscillations of the objective only, since the Markovian walker is ''kicked'' up into the (disordered) high-energy regime at the beginning of each bouncing iteration.

The energies drop as a function of the bouncing iterations for all T_B's besides $T_B=200$, i.e., $T_B>T_f$; there the energy is virtually constant. Nevertheless, the minimal energy (tour length) for $T_B=200$ oscillates strongly, even approaching values $50\,900<\mathcal{H}_i^{\min}<51\,000$. They are usually lost in the next iteration, whereas the minimal energy decreases initially and then remains in a certain range for $T_B<T_f$. Finally, for $T_B<T_c$ it is virtually constant.

In Fig. 6, we show the absolute improvement due to the bouncing process. As a measure for the improvement we define a normalized gain

$$g_i=\frac{\langle\mathcal{H}_1\rangle_E-\langle\mathcal{H}_i\rangle_E}{\langle\mathcal{H}_1\rangle_E-\mathcal{H}_{\text{opt}}} \tag{3.1}$$

(i denotes the index of the current iteration, $\langle\mathcal{H}_i\rangle_E$ the ensemble average of the energy in iteration i, and $i=1$ refers to the final energy of the initial monotonic optimization run). Here the success of the bouncing idea becomes evident.

(1) We can achieve a significant gain over the simple monotonic cooling schedule (as it is still used in the primary stage).

(2) The gain strongly depends on T_B, the criterion for choosing T_B is due to the evaluation of the specific heat C_T and the susceptibility χ_T during the initial cooling process. The largest gain can be obtained for T_B slightly below T_f (in our case $T_B=100$ and 50), whereas the gain completely disappears for $T_B>T_f$ ($T_B=200$). Also, the gain becomes marginal and ultimately vanishes in the limit $T_B\rightarrow 0$, as expected, since the local energy valley in which the MC walker sits cannot be left. A good estimate for a lower bound for T_B is the width of the specific heat peak, its low-temperature leg extends in case of the PCB442 instance to a temperature of roughly 40 in case of SA. This is in good agreement with the peak of χ_T, which marks the lower bound, too.

In Fig. 7, we provide the overlap o_i between final configurations of the succeeding bouncing iterations i and $i-1$. This quantity measures the percentage of TSP edges present in the final results of succeeding bouncing iterations, thus giving an estimate of how strongly the configurations are changed between succeeding iterations. Interestingly, this overlap becomes nearly constant after a few iterations, i.e., the rate of change remains constant during the bouncing process for a certain T_B. We clearly find that the overlap strongly depends on T_B (i.e., the smaller the bouncing temperature T_B, the larger the overlap), and thus only a few rearrangements take place. We want to point to the fact that although the overlap for $T_B=50$ is rather large, the gain is also large. This regime is very interesting for technical optimization applications, since it allows an optimization of an existing process—e.g., a tour schedule or production environment, which would serve as the input configuration for a bouncing treatment—without the introduction of major rearrangements in large parts (see Fig. 7).

An interesting point is the question whether for the bouncing algorithm the Grest hypothesis [27]

$$\frac{1}{\langle\mathcal{H}_i\rangle_E-\mathcal{H}_{\text{opt}}}\propto[\ln(i\tau)]^{\zeta}, \tag{3.2}$$

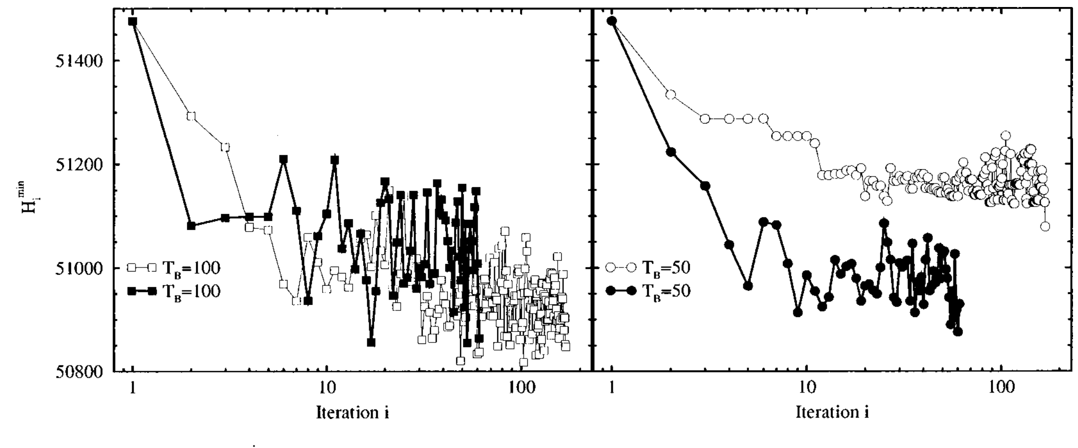

FIG. 10. Minimal length values $\mathcal{H}_i^{\min}$, data, and notation correspond to the previous figure, $T_B=100$ (left part) and $T_B=50$ (right part), TA, and PCB442.

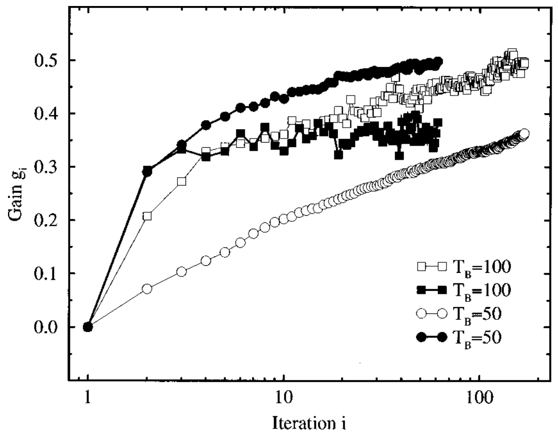

FIG. 11. Gain g_i vs index i of the bouncing iteration for $T_B = 100$ (squares) and $T_B = 50$ (circles), TA, and PCB442. Filled symbols denote simulation runs with a doubled calculation time, in contrast to the originally obtained data (open symbols).

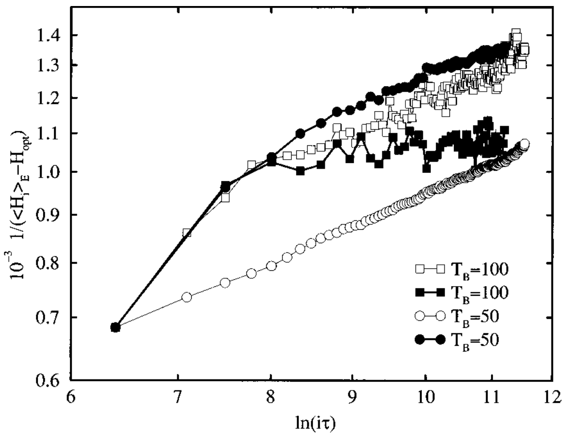

FIG. 13. Grest hypothesis: $1/(\langle\mathcal{H}_i\rangle_E - \mathcal{H}_{\text{opt}})$ vs $\ln(i\tau)$ for $T_B = 100$ (squares) and $T_B = 50$ (circles), TA, and PCB442. Filled symbols denote simulation runs with a doubled calculation time, in contrast to the originally obtained data (open symbols).

[where τ is the calculation time in each bouncing iteration, and $\zeta \approx 1$] is able to describe the quality of the results achieved after a certain time. Interestingly, we obtain relatively small values for ζ, e.g., $\zeta \approx 0.8$ for $T_B = 50$ and 100, and $\zeta \approx 0.2$ for $T_B = 25$. ζ vanishes for $T_B > T_f$ and for $T_B \to 0$ (Fig. 8).

In a further step, we want to study the effect of additional computation time on the results discussed earlier. For this purpose we use a nonuniform computation time distribution within the bouncing iterations, i.e., in addition to the previously used time we spend the same amount of computational effort at the first temperature step T_B, the beginning of the iteration, only. In our example we have initially (index $i \geqslant 2$) 121 steps at T_B, 119 steps at $0 < T < T_B$, and one step at $T = 0$. The monotonically cooled run in front of the bouncing process (index $i = 1$), however, remains unchanged, i.e., we bounce the same input configurations as previously.

The most challenging results are achieved for $T_B = 100$ and 50, which we want to discuss in Figs. 9–13. For $T_B = 50$, we obtain far lower energies for each bouncing iteration i using the additional computation time, whereas for $T_B = 100$ the runs with more computation time produce the better results in the first few bouncing iterations only (Figs. 9 and 10). Later, for $T_B = 100$, the runs with the original amount of calculation time become superior (Fig. 11), the runs with additional time cannot provide further improvements, and the gain is nearly constant after a few iterations.

Comparing the two runs which lead us to the lowest energies (these are the $T_B = 50$ run with doubled computation time, and the $T_B = 100$ bouncer with the original settings) we come to the following findings: $T_B = 50$ with the enlarged calculation time has a significantly lower mean energy and larger gain than even the $T_B = 100$ run with the original computational investment (which, in turn, has been much better than the unchanged $T_B = 50$ run). However, the enhanced $T_B = 50$ calculation does not lead to solutions with an energy below 50 850 within its total of 60 iterations, whereas the unchanged (i.e., nonprolonged) $T_B = 100$ bouncer was able to break below the 50 850 value at least six times within its initial 119 bouncing iterations (which, in turn, correspond to the same calculation time as the time-doubled $T_B = 50$ run).

In Fig. 12, we show the overlaps o_i for the runs with $T_B = 100$ and 50, both for the original and doubled calculation times. If additional time is used, the overlap niveau decreases. This is quite clear because the system has more time to walk around at T_B, and therefore results of succeeding bouncing iterations show larger differences. For $T_B = 200$ and 12.5 (not shown here), the overlap remains roughly constant; there is no significant gain. But for $T_B = 25$ with additional calculation time, we obtained nearly the same gain and overlap as for $T_B = 50$ with the original calculation time.

In conclusion, an overlap below 0.75 leads to a gain which is either zero from the beginning or converges against a constant greater than zero, and an overlap between 0.78 and 0.962 leads to a strong and continuous increase of the gain (at least in the observed range); again, for an overlap larger than 0.962, the gain is relatively small. (These numbers are, of course, only valid for the PCB442 problem in combination with TA.)

In order to show that our bouncing method also works with SA, we also performed 128 runs with SA with the original number of steps, but a smaller number of bouncing itera-

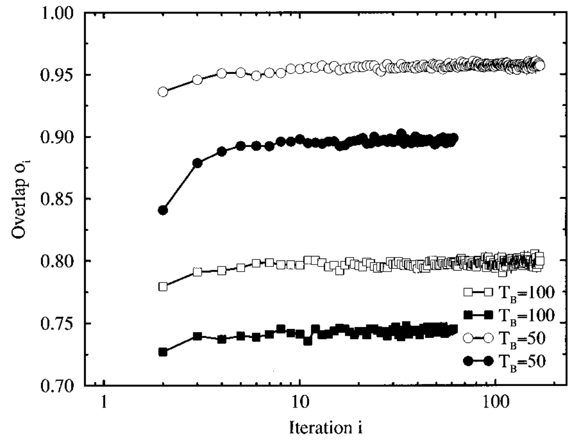

FIG. 12. Overlap o_i between the resulting configurations of successive bouncing iterations i and $i-1$ for $T_B = 100$ (squares) and $T_B = 50$ (circles), TA, and PCB442. Filled symbols denote simulation runs with a doubled calculation time, in contrast to the originally obtained data (open symbols).

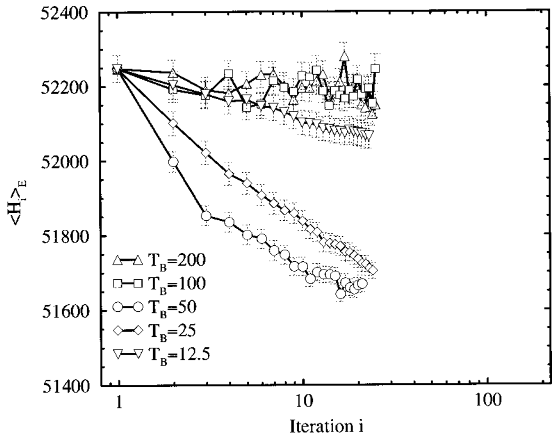

FIG. 14. Ensemble mean of the energy, $\langle \mathcal{H}_i \rangle_E$ vs index i of the bouncing iteration for different bouncing temperatures T_B, SA, and PCB442.

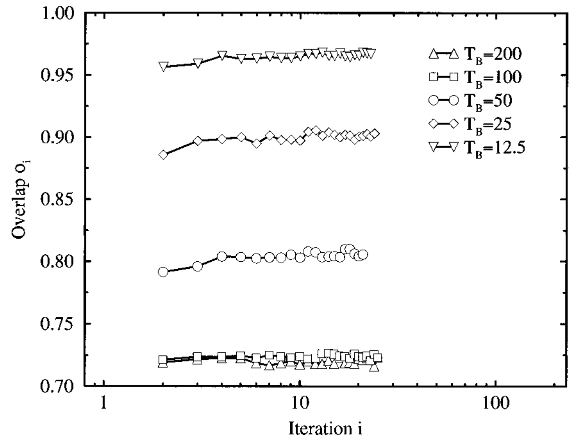

FIG. 16. Overlap o_i between the resulting configurations of successive bouncing iterations i and $i-1$ for different bouncing temperatures T_B, SA, and PCB442 (the error bars are of the size of the symbols).

tions. We took the same configurations from the initial monotonous iteration (therefore, the values for the iteration index $i=1$ are the same) and bounced it then with SA. In Fig. 14, we provide the mean energy, which decreases for $T_B \leqslant 50$. Comparing Fig. 14 with Fig. 4, we see that $T_B=100$ with TA and $T_B=50$ with SA lead to nearly the same curve, similarly $T_B=25$ with TA and $T_B=12.5$ with SA. The curve for $T_B=25$ with SA lies in between the curves $T_B=50$ and 100 with TA. Interestingly, the $T_B=100$ curve with SA stays nearly constant. From this we conclude that the T_B temperature ranges are renormalized using SA instead of TA, down to smaller values. We interpret this as follows: SA is able to climb with a certain probability over high mountains in the energy landscape even at small T, and therefore to reach better configurations easily. This is not possible for TA at similar threshold values. This factor 2 can correspondingly be extracted from the shift of the peak positions of C_T and χ_T for SA and TA, as shown in Figs. 2 and 3. In Fig. 15, we provide the gain achieved with SA. The results can be interpreted similarly to those in Fig. 14. Of interest, however, is that the gain for $T_B=100$ stays positive during all iterations. The results for the SA overlap in Fig. 16 confirm our earlier TA results: The bouncing temperature regime splits into three ranges, with corresponding effects on gain and overlap as described for TA. Very interesting are the results concerning Grest, shown in Fig. 17: we obtain $\zeta \approx 1$ for $T_B=50$ and $\zeta \approx 1.2$ for $T_B=25$. For $T_B=12.5$ or for $T_B \geqslant 100$, the exponent ζ is very small and nearly vanishes. A cross-check of these results with other instances from Reinelt's library allowed us to draw similar conclusions.

IV. PARALLEL BOUNCING

There is a simple path to extend bouncing for an implementation on a parallel computer. Several parallelization strategies are suitable.

At the end of the primary cooling process, the configuration obtained by the monotonic cooling is distributed over a number of processors, each of which performs one or several statistically independent bouncing iterations. After that, the best-so-far configuration among the processors is distributed again over all processors, and the previous iteration is repeated.

The results of a number of bouncing iterations can serve as an input for mutations of a parallel genetic algorithm or of a parallel evolution strategy.

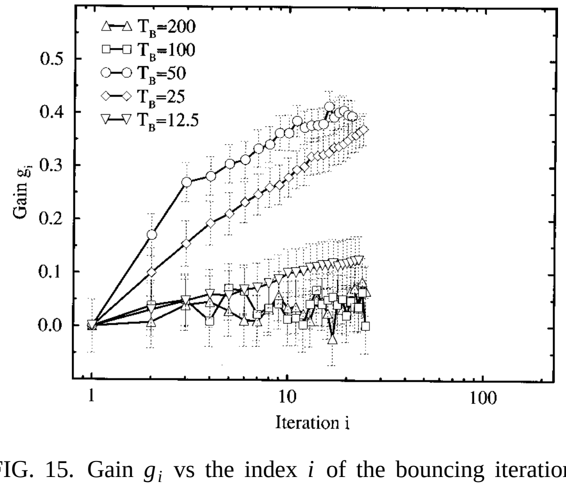

FIG. 15. Gain g_i vs the index i of the bouncing iteration for different bouncing temperatures T_B, SA, and PCB442.

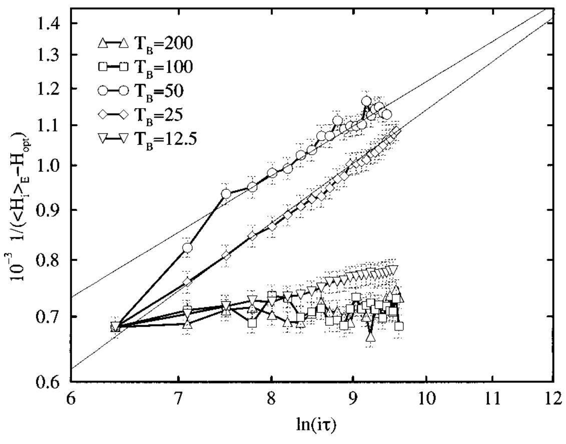

FIG. 17. Grest hypothesis: $1/(\langle \mathcal{H}_i \rangle_E - \mathcal{H}_{\rm opt})$ vs ln $(i\tau)$ for different bouncing temperatures T_B, SA, and PCB442. The thin solid lines are fits, giving exponents $\zeta=1$ for $T_B=50$ and $\zeta=1.2$ for $T_B=25$.

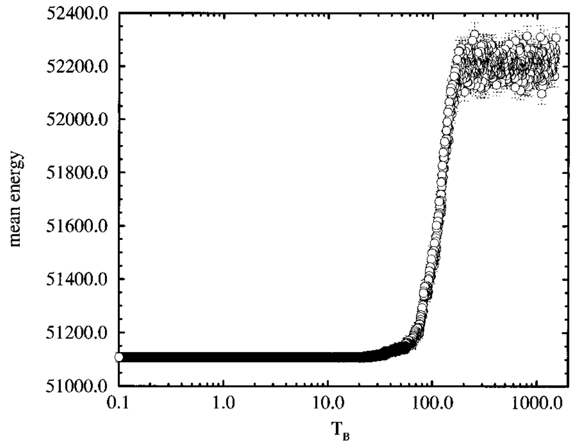

FIG. 18. Mean energy vs T_B for the parallel ensemble based bouncing algorithm, ensemble size 128, with controlled decrementation of T_B, TA, and PCB442.

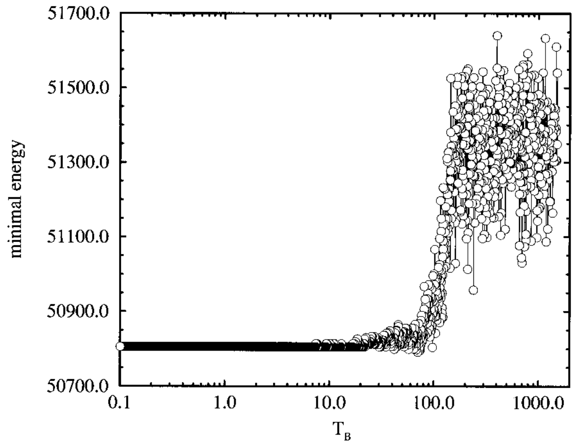

FIG. 19. Minimal energy vs T_B for the parallel ensemble based bouncing algorithm, ensemble size 128, with controlled decrementation of T_B, TA, and PCB442.

Ensemble based bouncing (EBB), which is the schedule we will use and describe in this section.

In the nonparallelized bouncing algorithm described in Sec. III, we chose a fixed bouncing temperature T_B, up to which the system is instantaneously heated and then cooled down. In a further step we propose an adaptive selection, i.e., a decremental lowering of T_B from an initial high value down to zero, leading to a kind of ''second'' control parameter with a second ''cooling schedule.'' Moreover, we are able to control and change the compute time continuously, i.e., the number of MC steps spent at a particular bouncing iteration.

For both purposes we use a parallel computer running an ensemble of independent bouncers, and extend the initial ensemble based cooling idea of Ruppeiner *et al.* [ensemble based simulated annealing (EBSA) [28]]: the bouncing temperature T_B is lowered if the condition

$$\langle \mathcal{H}_{i+1} \rangle_E \geq \langle \mathcal{H}_i \rangle_E \tag{4.1}$$

becomes true; here $\langle \cdots \rangle_E$ denotes the ensemble average over independent bouncers at the ith iteration. Note that we do not decrease the temperature as in EBSA or the threshold as in the TA version EBTA inside a bouncing iteration using the ensemble based rule: only the starting temperature T_B is decreased according to the adaptive kind of the ensemble based ansatz. Therefore we do not use the energies after each sweep for $\langle \cdots \rangle_E$ in contrast to Ref. [28], but we take the average over the lengths of the final results of different runs. Inside the bouncing iteration we work with TA; parameters such as the number of steps are the same as above. T_B is decreased by factors of 0.99.

Figure 18 shows the mean energy (i.e., TSP length) vs bouncing temperature T_B: We find a decrease, quite similar to the initially obtained decrease of the energy during the simple monotonic cooling, as shown in Fig. 2 for TA. Although it seems to be identical at a first glance, we want to make a strong point out of the fact that the temperature and energy ranges are completely different. Figure 19 shows the minimal energy vs the bouncing temperature T_B: we find that with this parallelized version of bouncing we are able to obtain even better results than with the serial bouncing, e.g., we reached at an optimum, and in six iterations energies smaller than 50 800, which we have not reached with the serial bouncing method at the above T_B. (With the serial TA algorithm, we came several times to the optimum with $T_B = 120$, but we had to invest much more calculation time.)

Finally we want to control the effort, i.e., the MC time or rather the number of bouncing iterations spent per T_B, using the EBB concept, again in analogy to the original EBSA concept. Figure 20 depicts a time (which in fact is a measure for the iterations spent at a particular T_B) in arbitrary units vs T_B. It increases when $T_B \rightarrow T_f$, as expected. At the end, for $T_B \rightarrow 0$, the bouncing process dies out. Most of the computation time is invested in the important temperature range between T_c and T_f. Thus the bouncing process can be automatically controlled.

It is certainly possible to apply the ensemble based cooling rule twice, for decreasing T_B between bouncing iterations as well as for decreasing T within a bouncing iteration, combining ensemble based bouncing and ensemble based threshold accepting. It is possible to save a large amount of CPU time using ''conventional'' EBSA-TA alone; however, this approach does not necessarily provide results very close to or

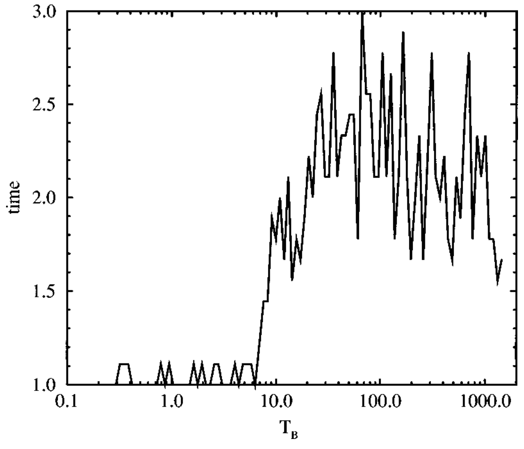

FIG. 20. Effort: Time (i.e., the number of bouncing iterations) spent per T_B using a parallel ensemble based bouncing algorithm (EBB) (ensemble size 128), TA, and PCB442; the data are binned to reduce statistical noise.

even identical with the ground state. Combining EBB and EBTA, we obtain the same quasioptimal quality of results compared to EBB alone, but saving 95% of the computation time.

V. CONCLUSION

We provide an algorithm to improve results obtained by finite Markov chain length Monte Carlo optimization procedures: the bouncing algorithm. It is based on the concept of simulated annealing type MC optimization algorithms. After a first conventional monotonic cooling process, a second cyclic heating-cooling treatment is applied, using the final configuration of the initial process as an input. The specific heat maximum obtained in the initial cooling process determines an appropriate upper bound for the ‘‘bouncing temperature’’ T_B, i.e., an upper bound for the cyclic reheating-cooling process. The susceptibility corresponding to an asymmetric overlap with the ground state delivers a lower bound. There is considerable evidence that within this temperature window the most dramatic rearrangements take place. The bouncing process enables the Markovian walker in state space to escape from local energetic minima, ‘‘forcing’’ a relaxation into energetically lower configurations.

The energy, i.e., the optimization objective, decreases, if we take a bouncing temperature T_B closely below the peak of the specific heat. The algorithm is described in detail, and its efficiency is demonstrated for a particular class of combinatorial optimization problems. The procedure can be easily transferred to other CO problems; we have applied it to, e.g., vehicle routing and production lines optimization problems among others. Therefore, it provides a rather universally applicable scheme. Finally we introduce an efficient extension of the bouncing scheme for parallel computers based on the ensemble parallelization concept, which allows an almost automatic control.

ACKNOWLEDGMENTS

We thank G. Dueck, T. Scheuer, and other members of the former TOP-C group at the IBM Scientific Center in Heidelberg, as well as U. Krey from the University of Regensburg, for useful discussions. Additionally, we kindly acknowledge the generous grant of computation time on the Intel Paragon system of the HLRZ/KFA, Jülich. Part of the work was supported by the Swiss National Foundation for Science.

[1] S. Kirkpatrick *et al.*, Science **220**, 671 (1983).
[2] X. Cerny, J. Optim. Theory Appl. **45**, 41 (1985).
[3] S. R. White, IBM T. J. Watson Research Lab. Computer Science Research Report No. 10 661, 1984.
[4] See, e.g., S. Geeman and D. Geeman, IEEE Trans. Pattern. Anal. Mach. Intell. **6**, 721 (1984).
[5] B. Hajek, Math. Op. Res. **13**, 311 (1988).
[6] T. S. Chiang *et al.*, SIAM J. Control Optim. **26**, 1455 (1988).
[7] R. Azencott, in *Simulated Annealing*, edited by R. Azencot (Wiley, New York, 1992).
[8] For simplicity we made the deliberate assumption $k_B \equiv 1$; therefore, k_B is simply neglected in the course of this presentation, but it may enter in an elaborate treatment in terms of units. Additionally, all lengths (energies) of the TSP instances are measured in arbitrary, dimensionless units (a.u.), making most expressions dimensionless.
[9] K. Binder and D. W. Heermann, *Monte Carlo Simulation in Statistical Physics* (Springer, Heidelberg, 1992).
[10] G. Dueck and T. Scheuer, J. Comput. Phys. **90**, 161 (1990).
[11] E. H. L. Aarts *et al.*, J. Stat. Phys. **50**, 187 (1988).
[12] G. Reinelt, *TSPLIB95*, University of Heidelberg, 1995; *The Traveling Salesman*, Springer Lecture Notes in Computer Science Vol. 840 (Springer, Heidelberg, 1994).
[13] O. Catoni, in *Simulated Annealing* (Ref. [7]).
[14] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications* (Kluwer, Dordrecht, 1989).
[15] E. Bonomi *et al.*, SIAM (Soc. Ind. Appl. Math.) Rev. **26**, 551 (1984).
[16] M. Grötschel and O. Holland, Math Program. **51**, 141 (1991).
[17] J. Schneider *et al.*, Comput. Phys. Commun. **96**, 173 (1996).
[18] S. Lin, Bell Syst. Tech. J. **44**, 2245 (1965); S. Lin and B. W. Kernighan, Oper. Res. **21**, 498 (1973).
[19] P. F. Stadler and W. Schnabl, Physica D **48**, 65 (1991).
[20] J. P. K. Doye *et al.*, J. Chem. Phys. **105**, 1495 (1997); Phys. Rev. Lett. **80**, 1357 (1998).
[21] Jun Gu, IEEE Trans. Syst. Man Cybern. **24**, 5 (1994).
[22] J. Schneider *et al.*, Physica A **243**, 77 (1997).
[23] L. D. Landau and E. M. Lifschitz, *Statistical Physics I* (Pergamon, Oxford, 1968).
[24] I. Morgenstern, in *Heidelberg Colloquium on Glassy Dynamics*, edited by L. van Hemmen and I. Morgenstern (Springer, Heidelberg, 1987).
[25] I. Morgenstern and D. Würtz, Z. Phys. B **67**, 397 (1987).
[26] F. Romeo and A. Sangiovanni-Vincentelli, Algorithmica **6**, 302 (1991); D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, Adv. Appl. Probab. **18**, 747 (1986).
[27] G. S. Grest *et al.*, Phys. Rev. Lett. **56**, 1148 (1986); G. S. Grest, in *Heidelberg Colloquium on Glassy Dynamics* (Ref. [24]).
[28] G. Ruppeiner *et al.*, J. Phys. I **1**, 455 (1991).