15 Cooling Techniques

15.1 Standard Cooling Schedules

Using the theory of Markov processes, several authors (see, e. g., [64, 142, 78, 65]) have proved the general existence of cooling schedules for simulated annealing (SA) with which the simulation ends up in the global optimum of the considered problem, however, after an infinite amount of time. Geman and Geman [65] showed for the classical case that it is necessary and sufficient for having a probability of one of ending in a global optimum that the temperature decreases like

$$T = \frac{a}{b + \log(t)}\,, \tag{15.1}$$

with a and b being positive constants that depend on the specific problem. t is the time elapsed since the start of the simulation and is usually measured in Monte Carlo steps. There are also cooling functions $T(t)$ for the other algorithms introduced in the previous chapters that lead to the globally optimum solution for some specific problem.

In practical applications, the available amount of time to produce a solution is finite. Therefore, faster ways of cooling the system down were developed that could be applied generally and that lead to very good solutions. These empirically found cooling schedules let the temperature decrease much faster to zero. However, they do not guarantee a convergence to the global optimum of the problem. Mostly two main cooling schedules are used:

- Linear/arithmetic cooling:

$$T = a - b \times t\,, \tag{15.2}$$

  where a is the initial temperature and b is the decrement by which the temperature is decreased. Usually, b is chosen in the interval $[0.01; 0.2]$. The initial temperature strongly depends on the problem considered.
- Exponential cooling:

$$T = a \times b^t\,. \tag{15.3}$$

  Again a is the initial temperature and b is a cooling factor, usually in the interval $[0.8; 0.999]$. In the literature, this cooling schedule is called by many names: "logarithmic", "geometric", or "exponential". We refer to it as exponential cooling.

The best cooling schedule will depend on the problem and the computing resources available (see, e. g., [207], where an optimal annealing schedule with a fixed number of steps was created for a particular problem). Much more complex cooling schedules than Eqs. (15.2) and (15.3) have been employed, with the decrease in temperature adapting to evidence of rapid or slow equilibration. The temperature need not even decrease monotonically. An example of these approaches is the following consideration for SA. One considers the relative weight $W_T(\sigma)$ of a configuration σ in relation to the weight of the ground state configuration with minimum energy $\mathcal{H}_0$, i. e.,

$$W_T(\sigma) = \frac{\pi_{\mathrm{equ}}(\sigma)}{\pi_{\mathrm{equ}}(\sigma_0)} = \exp\left(-\frac{\mathcal{H}(\sigma) - \mathcal{H}_0}{k_{\mathrm{B}}T}\right) \tag{15.4}$$

and demands that $W_T(\sigma)$ not change too much if the temperature T is decreased from a value T_k to a new value T_{k+1}; thus, one demands

$$\frac{1}{1+\delta} < \frac{W_{T_k}(\sigma)}{W_{T_{k+1}}(\sigma)} < 1 + \delta\,, \tag{15.5}$$

with some small constant $\delta > 0$. For $T_{k+1} < T_k$, the left inequality is always fulfilled. Solving the right inequality for T_{k+1}, one gets

$$T_{k+1} > \frac{T_k}{1 + \dfrac{T_k \log(1+\delta)}{\mathcal{H}(\sigma) - \mathcal{H}_0}}\,. \tag{15.6}$$

Of course, one will not check how to fulfill this inequality for all states σ. Instead, one considers either the thermal average at temperature T_k and replaces the deviation $\mathcal{H}(\sigma) - \mathcal{H}_0$ in the last formula by $\langle\mathcal{H}\rangle_{T_k} - \mathcal{H}_0$. However, usually the optimum value for a specific problem instance is not known. In these cases, one considers the variance $\mathrm{Var}_{T_k}(\mathcal{H})$ and approximates the mean deviation from the minimum energy value with $3\sqrt{\mathrm{Var}_{T_k}(\mathcal{H})}$, as the individual configurations are supposed to occur according to the Gaussian distribution, in which 99.7% of all occurrences are within three times of the standard deviation range around the mean value. Thus, the formula reduces in its applicable version to

$$T_{k+1} > \frac{T_k}{1 + \dfrac{T_k \log(1+\delta)}{3\sqrt{\mathrm{Var}_{T_k}(\mathcal{H})}}}\,. \tag{15.7}$$

But if there is no time for implementing and testing such a tuned cooling schedule, which is usually based on additional measurements that also require calculation time, the two schedules of linear and exponential cooling are the obvious choice. To choose between them, consider the characteristics of the specific heat: if $C(T)$ is more or less symmetric when plotted on a linear temperature scale, then linear cooling is preferable. If, however, the peak only becomes rather symmetric when plotted against $\log(T)$, then the system

organizes itself over several orders of magnitude in temperature, and the exponential schedule is preferred.

It is rather straightforward to find a proper initial value of the temperature. At the beginning of the optimization run, a random walk is performed and $|\Delta\mathcal{H}|_{\max}$, the largest absolute value of the energy differences occurring between successive configurations, is saved. Then, if using SA, the initial value of the temperature is chosen in such a way that the acceptance rate of all moves exceeds a chosen value, e. g., it should be at least 90% at the beginning. Then the rule of thumb

$$T_{\text{initial}} = -\frac{|\Delta\mathcal{H}|_{\max}}{\ln(0.9)} \approx 10 \times |\Delta\mathcal{H}|_{\max} \tag{15.8}$$

is applied. It is even simpler with threshold accepting:

$$\mathcal{T}h_{\text{initial}} = |\Delta\mathcal{H}|_{\max}\,. \tag{15.9}$$

In contrast, the maximum occurring energy value $\mathcal{H}_{\max}$ has to be saved when using the great deluge algorithm (GDA). One simply sets

$$\mathcal{T}_{\text{initial}} = \mathcal{H}_{\max}\,. \tag{15.10}$$

Instead of the maximum energy difference (or the maximum energy in the case of the GDA), often the mean value of the measurements is considered because the choice above might lead to too large initial temperatures and therefore some waste of calculation time. However, if the various moves used show different ranges of energy differences, then the problem occurs that if the initial temperature is chosen too small, the system might explore only a restricted area of the energy landscape.

Of course, other possibilities for choosing the initial value of the temperature are imaginable. In summary, if one starts at too low temperatures, the problem arises that the quality of the results decreases as the system is restricted to the local valley of the initial configuration. The Monte Carlo walker can only climb down to the local minimum and fails to reach better solutions for which he/she would have to leave the local valley and climb over barriers. Alternatively, too much calculation time might be wasted at high temperatures if the initial temperature was chosen too large. Therefore, sometimes the following procedure is chosen: a rather fast run with an initial temperature chosen as described above is performed. The decrease in the energy and the progression of the specific heat are plotted vs. the temperature T. Then a smaller initial temperature is chosen for the production runs. Besides guessing intuitively a good value for the initial temperature, one can set the initial temperature in the range of the peak of the specific heat C: let τ be the temperature at which C is maximum and $\Delta\tau$ be the width of the peak. Then one usually chooses the initial temperature as

$$T_{\text{initial}} = \tau \pm \Delta\tau\,. \tag{15.11}$$

The three points $(\tau - \Delta\tau, 0)$, $(\tau, C_{\max})$, and $(\tau + \Delta\tau, 0)$ in the specific heat vs. temperature diagram form a "magic triangle", which indicates both the temperature range in which most rearrangements of the system take place and nearly the whole amount of energy the system loses while being cooled down. The final temperature has to be chosen in such a way that the system is really frozen or only trivial moves with $\Delta\mathcal{H} = 0$ are accepted. Therefore, the acceptance rate of the nontrivial moves has to vanish at the end of the optimization run. However, there is no good criterion for how long one has to wait to be absolutely sure that no nontrivial move would no longer be accepted, as this depends on the underlying energy landscape. (Of course, all possible moves from the actual solution to neighboring configurations could be considered. However, how many neighbors a configuration exhibits depends on the move set and the size of the system. The number of neighbors might be so large that it is impossible to check them all in order to prove that a certain state is really a local minimum in the energy landscape.) Therefore, one usually performs a few greedy steps at the end of each optimization run, which last long enough to be quite sure to be in a local minimum. One simply sets $T = 0$, $\mathcal{T}h = 0$, and $\mathcal{T}$ = "best result so far".

15.2 Nonmonotonic Cooling Schedules

So far, we have only considered monotonic cooling schedules by which the control parameter, e. g., the temperature is not increased during the optimization run. This approach is motivated by the physical picture of a molten metal block: if it is cooled down very fast, i. e., quenched down, then only a polycrystalline structure can develop, in contrast to the nice results one gets if the melting is cooled down very slowly.

This physical picture can be extended even further by looking at the work of a blacksmith in former times: after cooling down the molten metal rather fast in a special form, the blacksmith treats it with a certain iterated reheating-cooling down strategy: first of all, the metal is reheated, but only up to a certain temperature at which it, e. g., glows red but does not start to melt. While the block cools down again, the blacksmith works on the block in order to improve it. Sometimes he also quenches the block down again with, e. g., water. Usually, he only works on the block during the cooling-down phase and not during the reheating phase. Due to this special treatment, the blacksmith can perform larger structural rearrangements on the considered block without having to melt his already partially completed work and therefore to start again. The temperature up to which the block is reheated usually determines the amount by which the block can be changed.

This approach can be transferred to SA and related optimization algorithms in various ways in order to get an optimum cooling schedule. However, it is usually impossible to determine an optimum cooling schedule, which is an optimization problem on its own, but there are still many ways to

develop nonmonotonic cooling schedules that perform well if one considers the basic finding of Romeo and Sangiovanni–Vincentelli [173] that a result nearly as good as the global optimum can only be reached if there is always a probability large enough for leaving any configuration, i. e., also any local optimum.

Strenski and Kirkpatrick showed in their early work [207] on this subject, in which they optimized the annealing schedule for a very simple problem using a fixed number of iterations, that if the amount of calculation time available is very small, then the optimum annealing schedule is nonmonotonic: their results for various systems indicate that after starting at a random solution, one should proceed with the greedy algorithm, i. e., the $T = 0$ mode of SA and its relatives. By this approach, the system is quenched down in the fastest possible way, until it freezes in a local optimum. After that the system is reheated up to a certain problem-dependent temperature. At this temperature, the system is able to cross smaller barriers in the energy landscape and therefore to leave a local valley in order to get to a better local optimum. Depending on the calculation time available, the system should either immediately switch to the greedy mode again in order to get stuck in a better local optimum nearby or go through a more or less pronounced cooling procedure (this "more or less" depends on the computing time available), which monotonically decreases the control parameter from the reheating temperature and ends up in the greedy mode.

Another approach is called bouncing [114, 190]: after a first conventional optimization run, in which the control parameter is reduced monotonically from a large initial value T_0 to 0, such that the system is transferred from a random configuration to a locally optimum solution, the final configuration is used as the initial configuration of a further optimization run. This second optimization run starts at a value $T_\mathrm{B} < T_0$ of the control parameter in order to correspond to the blacksmiths approach, who holds the metal block in the fire while not working on it. (This approach could be called an inverse quench.) Now the system is again at least partially able to move through the energy landscape. The amount of rearrangement possible is determined by the value of the bouncing temperature T_B. In this bouncing iteration, a standard optimization run is performed, i. e., the control parameter is decreased step by step from T_B to 0. Then the new final configuration is used as the initial configuration for the next bouncing iteration in which the control parameter is again reduced from T_B to 0. This approach is iterated several times.

The main advantage of this bouncing approach compared to the monotonic cooling approach is that the standard monotonic optimization run starts at a totally random system, whereas the bouncing scheme can start from preoptimized solutions in the bouncing iterations. These preoptimized solutions already inherit some structure, i. e., some information about the system. The question is now up to which value T_B of the control parameter will the system be reheated, i. e., which T_B is appropriate for the cooling process $T_\mathrm{B} \geq T \geq 0$? On the one hand, local information will be kept and only a small number of

structures shall be destroyed. On the other hand, the optimization process will be able to leave a bad local optimum and get to a better one.

One can distinguish three regimes for T_B:

- Bouncing as "slightly warming up":
  After the first conventional cooling process, the reheating is done only very slightly, i. e., in a temperature range below the critical temperature T_C, at which the ordering transition of the system happens. This critical temperature can be determined for systems for which an order parameter can be defined. In that case, one can measure the susceptibility, which peaks at the critical temperature.
  In this regime, the energy values of the final configurations of successive bouncing iterations first decrease monotonically and then stay nearly constant [190]. Only seldom does the system jump to a worse solution, and then it is able to jump back in the next bouncing iteration. In this regime, bouncing more than ten iterations under the same external constraints seldom gives any further significant improvement.
- Bouncing up to the maximum of the specific heat:
  The system is reheated nearly to the freezing temperature T_f, i. e., $T_C < T_B < T_f$. This temperature range corresponds to a partially ordered phase of the system. A transition to the totally unordered range is not performed, similar to the approach of the blacksmith, who reheats the metal block in order to perform some structural rearrangements but does not melt it. The results of previous reheating iterations of the considered item do not get lost completely.
  In this temperature regime, the energy values of the final configurations of successive bouncing iterations also decrease monotonically, but only on average: there are now large fluctuations; sometimes a bouncing iteration leads to an improvement, sometimes it leads to a deterioration. But all in all, in this regime, it is possible to arrive at much better solutions than in the small T_B regime. The system is able to climb also over larger barriers in the energy landscape and therefore to leave a suboptimal local area.
  Generally, the optimum T_B value depends on the calculation time spent in each bouncing iteration: if the amount of calculation time is increased, the system has a greater ability to climb over barriers such that a smaller T_B value can be used than if spending less calculation time in order to get the same number of improvements [190].
- Bouncing above T_f:
  Finally, one can also consider the behavior of a bouncing process with $T_B > T_f$: the system melts such that no qualitative change of the results compared to a standard monotonically cooled optimization can exist, as the system is kicked up into the unordered high-energy regime.

Summarizing, one must work along these lines: one performs a conventional optimization run in which the specific heat C and, if possible, the susceptibility χ are measured during the cooling process. The peak of the specific heat

marks the freezing temperature T_f, the peak of the susceptibility the critical temperature T_C. One can define a gain that can be achieved with bouncing by

$$g_i = \frac{\langle \mathcal{H}_0 \rangle_\mathrm{E} - \langle \mathcal{H}_i \rangle_\mathrm{E}}{\langle \mathcal{H}_0 \rangle_\mathrm{E} - \mathcal{H}_\mathrm{opt}}, \tag{15.12}$$

with i being the number of the bouncing iteration and $\langle \mathcal{H}_i \rangle_\mathrm{E}$ the ensemble average of the energy values of the final configurations of the bouncing iteration i (where $i = 0$ denotes the initial monotonically cooled optimization run). This gain g_i is normalized in such a way that it vanishes if no improvements can be achieved and that it reaches a value of 1 if the optimum is reached.

The extent of this gain g_i depends on the value of T_B; it is largest for T_B slightly smaller than T_f if only a small amount of calculation time is used in each bouncing iteration. When spending more calculation time, one should reduce this T_B. If $T_\mathrm{B} > T_\mathrm{f}$, then there is no gain at all. Similarly, the gain vanishes in the limit $T_\mathrm{B} \to 0$ as expected as the Monte Carlo walker is stuck in the local valley in the energy landscape, and cannot leave. A lower bound for T_B is the critical temperature T_C and also the width τ of the peak of the specific heat, such that one should not choose a T_B smaller than T_C or smaller than $T_\mathrm{f} - \tau/2$.

However, the question remains why this bouncing approach works: the algorithm works with random moves that could also lead to deterioration. However, the algorithm obviously can make use of the information stored in the initial configuration, i. e., the final result of the previous optimization run. By not melting the system, many structures of the previous solution are retained—they need not be found again—such that the optimization run has some preoptimized background on which it works, which leads to better solutions if T_B is chosen correctly.

A further explanation for the success of this bouncing idea can be obtained by measuring the Hamming distance between successive results of bouncing iterations: first of all, one must consider the mean Hamming distance between quasioptimum solutions produced by independent optimization runs. Then one will find that there are really three T_B regimes that can be distinguished also in the Hamming distance graphics: if $T_\mathrm{B} > T_\mathrm{f}$, then the Hamming distance between successive results of bouncing iterations is of the same size as that of independent results. On the contrary, if T_B is very small, the Hamming distance is (nearly) zero, as successive bouncing iterations lead to (roughly) the same result. One should therefore plot the graphic Hamming distance vs. bouncing iteration number for $T_\mathrm{B} = T_\mathrm{f}$ and $T_\mathrm{B} = T_\mathrm{C}$ and choose a T_B in such a way that the curve in this graphic is between these extreme curves but nearer to the curve for $T_\mathrm{B} = T_\mathrm{f}$, e. g., the distance to the curve for $T_\mathrm{B} = T_\mathrm{C}$ should be one to three times larger than the distance to the curve for $T_\mathrm{B} = T_\mathrm{f}$ [190].

Instead of using a fixed value for T_B, one can also think of a variable T_B. One can even go so far as to introduce a cooling schedule for T_B; for example, one could use the same type of cooling schedulte for T_B as for the control parameter itself, thus bouncing the system more and more gently, such that it is at first enabled to climb over larger barriers in the energy landscape and then only able to cross smaller barriers. This approach also leads to (quasi) optimum results [190].

There are also other opportunities to bounce: for example, one can perform a conventional optimization run, in which the control parameter is decreased monotonically. However, after a certain number of control parameter steps, one or more greedy steps are introduced in order to quench the system in a local optimum. After these greedy steps, the conventional optimization run is continued at the next value of the control parameter. After one or more additional control parameter steps, again some greedy steps are performed, and the system is again led to a local optimum. This approach is repeated until the value of the control parameter is so small that the system is mostly in the greedy mode. At the end of the optimization run, the best local optimum solution found is returned as the result of the optimization procedure.

15.3 Ensemble Based Schedules

There are also ways to find either good or fast cooling schedules by using parallel computers. In this case, usually the same instance of a problem is treated simultaneously on several processors. The number of the processor or the process number often serves as a seed for the random number generator, such that all processors start with different initial solutions or depart from a common initial solution in different directions.

These schedules have to avoid wasting too much calculation time in large values of the control parameter, in which the system performs a quasi-RW. On the other hand, the schedule should not start at too small values and should not decrease the control parameter too fast in important ranges as then only bad results will be achieved. The ensemble based simulated annealing (EBSA) approach tries to fulfill these requirements: if the system to be optimized is equilibrated at a certain temperature, then the energy values of the successive configurations fluctuate in a certain range around the mean value of the energy at this temperature. If the temperature T is decreased, then the energy values decrease until they oscillate around the new and smaller mean value of the energy at this smaller temperature. This decrease in the energy values of the successive configurations does not occur monotonically, but it is quite noisy. If the temperature is only decreased very slightly, it is hard to detect this decrease in an energy value vs. time diagram. However, if one averages over many optimization runs, then the fluctuations during the decrease of

the mean energy cancel out, such that one usually gets a strongly monotonic decreasing curve until the new equilibrium value is reached. Small oscillations remain in the equilibrium phase, as the number of parallel optimization runs is finite. This property of such an averaging is used on parallel computers for determining for an ensemble of optimization runs whether the proposed amount of calculation time was sufficient to reach the equilibrium at the new temperature.

Therefore, EBSA starts on an ensemble of p processors with one SA step, using different initial configurations at a large value of the temperature. At the end of this step, the ensemble average $\langle \mathcal{H}_0 \rangle_{\mathrm{E}}$ over the energy values of the p configurations is calculated. Then the following steps are iterated in a loop (i denotes the number of the iteration):

- The p slaves perform one SA step at the current value of the temperature T.
- Then they send the energy values of their final configurations to the master processor.
- This master processor calculates the new ensemble average $\langle \mathcal{H}_{i+1} \rangle_{\mathrm{E}}$ and the corresponding variance $\mathrm{Var}_{\mathrm{E}}(\mathcal{H}_{i+1})$.
- If the condition

$$\langle \mathcal{H}_{i+1} \rangle_{\mathrm{E}} - \langle \mathcal{H}_i \rangle_{\mathrm{E}} \geq \gamma \left(\frac{\mathrm{Var}_{\mathrm{E}}(\mathcal{H}_{i+1})}{p-1} \right)^{\nu} \tag{15.13}$$

  (containing two parameters γ and ν) is fulfilled (often a $>$ sign is used instead of $\geq$ in the literature, but according to our experience $\geq$ is better), then the temperature is decreased; otherwise the old value of the control parameter is kept.
- The master processor overwrites the old ensemble average with the new one and sends the temperature value back to the slaves.

This way, the communication time between the master and the slaves can be kept to a minimum. Even the decision to finish the run can be easily transferred from the master to the slaves, i. e., by sending a negative value for the control parameter from the master to the slave processors, by which the slaves can identify the end of the simulation, at which they simply have to send their final configuration to the master, which chooses the best of these.

The main parameter of this parallel approach is the factor γ, which can be chosen in various ways:

- $\gamma \to -\infty$
  In this case, one gets the so-called exponential cooling scheme, because Eq. (15.13) reduces to

$$\langle \mathcal{H}_{i+1} \rangle_{\mathrm{E}} \geq -\infty \,. \tag{15.14}$$

This condition is always fulfilled, leading to an automatic decrease of the control parameter after each step.

- $\gamma = 0$
  Here one speaks of the simple adaptive cooling scheme. Equation (15.13) is simplified to
  $$\langle \mathcal{H}_{i+1} \rangle_{\mathrm{E}} \geq \langle \mathcal{H}_i \rangle_{\mathrm{E}} \,. \tag{15.15}$$
  The current value of the temperature is kept constant as long as the ensemble average of the energy decreases in a monotonic way and is then decreased if this average increases or stays the same.
- $\gamma > 0$
  This case is called the weakened adaptive scheme. For this scheme, only small values for γ, e. g., $\gamma = 0.5$, are used. If one looks again at a simplified version of Eq. (15.13),
  $$\langle \mathcal{H}_{i+1} \rangle_{\mathrm{E}} \geq \langle \mathcal{H}_i \rangle_{\mathrm{E}} + \varepsilon \,, \tag{15.16}$$
  with
  $$\varepsilon = \gamma \left(\frac{\mathrm{Var}_{\mathrm{E}}(\mathcal{H}_{i+1})}{p-1} \right)^{\nu} \geq 0 \,, \tag{15.17}$$
  then one finds that the temperature is decreased only if the deterioration of the ensemble average of the energy values exhibits this ε.
  This approach must be used instead of the adaptive cooling scheme if the number p of available processors is rather small, e. g., $p < 100$, as in this case the fluctuations during the decrease of the energy are too large, such that the system thinks that it is already equilibrated at the new temperature whereas there was only a fluctuation. This would lead to too rapid a decrease in the temperature. However, one must be careful in the choice of ν in this case: in the literature, ν is usually chosen as 1. However, how large the ensemble variance can become depends on the problem. If it is too large, then the condition is never fulfilled and the algorithm is in an endless loop at some value of the control parameter. For every problem, the appropriateness of the ν value has to be tested. For example, $\nu = \frac{1}{2}$ is a good value for the traveling salesman problem, for which $\nu = 1$ already leads to endless loops at rather high values of the control parameter.

Summarizing, the basic thought of this ensemble based approach is to let the ensemble average of the energy values drop again and again at a fixed temperature. If it does not decrease any further, then the system is believed to have reached equilibrium at the given new temperature. Therefore, the temperature can be decreased again. At the end of the optimization run, all ensemble members get stuck as the control parameter is very small.

This ensemble based approach is defined in the context of the equilibrium properties of SA. However, this approach can also be transferred to other control strategies: for example, ensemble based threshold accepting (EBTA)

works in the same way as EBSA. The temperature is simply replaced by the threshold. Analogously, this approach can be transferred to the temperature in the Tsallis-based methods or to the water level in the GDA.

The bouncing approach is also suited for parallel enablement. For example, one could simply use the best configuration of bouncing iteration i as an input for the next iteration on all p processors. But one could also define an ensemble based bouncing (EBB): in this case, the bouncing temperature T_{B}, up to which the temperature T is increased at the beginning of each bouncing iteration, is not held constant but decreased according to the ensemble based rule, i. e., if using the adaptive scheme, T_{B} is decreased if $\langle \mathcal{H}_{i+1} \rangle_{\mathrm{E}} \geq \langle \mathcal{H}_i \rangle_{\mathrm{E}}$. However, here the ensemble average $\langle \mathcal{H}_i \rangle_{\mathrm{E}}$ is the average over the energies of the final configurations of bouncing iteration i. Note that in contrast to EBSA and EBTA, it is not the temperature and the threshold that are decreased inside a bouncing iteration but the bouncing start temperature T_{B} is decreased according to the ensemble-based rule. Of course, it is also possible to use the ensemble-based approach twice for this bouncing approach: first, it is used for decreasing T_{B}, but secondly one can also decrease the control parameter inside a bouncing iteration according to the ensemble-based rules. This combines EBB and EBSA/EBTA.

It is also interesting to combine the ensemble-based approach with search space smoothing (SSS). Let us use the greedy algorithm as the underlying local search technique inside SSS. As therefore no energy deteriorations are allowed during one α step, the adaptive rule (15.15) of EBSA and EBTA is modified as follows: if $\langle \mathcal{H}_{i+1} \rangle_{\mathrm{E}} = \langle \mathcal{H}_i \rangle_{\mathrm{E}}$, then decrease α, otherwise keep the current value of α. However, the question arises as to which Hamiltonian will be used in this case, such that one can define three different rules for ensemble based search space smoothing (EBSSS):

- "Smoothed" rule:
  The smoothed Hamiltonian $\mathcal{H}^{\alpha}$ is used for calculating the ensemble average $\langle \mathcal{H}_i^{\alpha} \rangle_{\mathrm{E}}$. After changing α, the smoothed Hamiltonian changes, such that sometimes ensemble averages of different Hamiltonians are compared with each other.
- "Original" rule:
  The ensemble average is calculated with the original Hamiltonian $\mathcal{H}^0$.
- "Both" rule:
  One can also consider both Hamiltonians and must therefore change the condition above: α is decreased only if both $\langle \mathcal{H}_{i+1}^{\alpha} \rangle_{\mathrm{E}} = \langle \mathcal{H}_i^{\alpha} \rangle_{\mathrm{E}}$ and $\langle \mathcal{H}_{i+1}^{0} \rangle_{\mathrm{E}} = \langle \mathcal{H}_i^{0} \rangle_{\mathrm{E}}$.

15.4 Simulated Tempering and Parallel Tempering

Simulated tempering (ST) was introduced by Marinari and Parisi [132]. It works like SA, i. e., the simulation starts with an initial configuration and applies a series of moves that are accepted or rejected according to the Metropolis criterion. In contrast to SA, ST does not automatically change the temperature according to a proposed cooling schedule. Instead, the temperature is also seen as a configuration variable and is changed according to the Metropolis criterion. One considers a set of M temperatures T_i with $T_1 < T_2 < \cdots < T_M$, which play the role of the various temperatures in SA. Thus, T_1 must be chosen so small that the system is already frozen, whereas T_M has to be large enough so that the system can explore the whole configuration space. One wants to apply a move $T_i \to T_j$. The question is now how to choose an appropriate transition probability.

For this purpose, one extends the configuration space by a further dimension, in which a variable i takes the discrete values $1, 2, \ldots, M$, denoting the individual temperature steps. The configuration in this joint configuration space has to be considered as a pair $(\sigma, i) \in \Gamma \times \{1, \ldots, M\}$, with σ being the configuration of the original space Γ and i a number between 1 and M. Then one considers the partition sum of this joint system, for which a new parameter g_i is introduced:

$$Z(T_i) = \sum_{\sigma \in \Gamma} \exp(-\beta_i \mathcal{H}(\sigma) + g_i)\,, \tag{15.18}$$

with $\beta_i = 1/(k_\mathrm{B} T_i)$. The g_i parameters are a function of parameter i and thus of the corresponding temperature T_i. They have to be chosen in a way such that

$$Z(T_i) = \text{const} = Z\,, \tag{15.19}$$

i. e., such that the partition sum of the joint system does not depend on the temperature value T_i, as this dependency will cancel out with the dependency of g_i. The equilibrium probability for the state (σ, i) is thus given by

$$\pi(\sigma, i) = \frac{1}{Z} \exp\left(-\frac{\mathcal{H}(\sigma)}{k_\mathrm{B} T_i} + g_i\right)\,. \tag{15.20}$$

There are now two types of moves:

- First, one can apply as usual a move $\sigma \to \tau$, which is now the move $(\sigma, i) \to (\tau, i)$. The detailed balance condition leads to

$$\frac{p((\sigma, i) \to (\tau, i))}{p((\tau, i) \to (\sigma, i))} = \frac{\pi(\tau, i)}{\pi(\sigma, i)} = \exp\left(-\frac{\mathcal{H}(\tau) - \mathcal{H}(\sigma)}{k_\mathrm{B} T_i}\right)\,. \tag{15.21}$$

  Thus, this move type can be accepted with the Metropolis criterion as usual at the given temperature T_i.

- Secondly, one can apply a move to change the temperature $T_i \to T_j$. Applying detailed balance leads to

$$\begin{aligned}\frac{p((\sigma,i)\to(\sigma,j))}{p((\sigma,j)\to(\sigma,i))} &= \frac{\pi(\sigma,j)}{\pi(\sigma,i)} \\ &= \frac{\exp\left(-\mathcal{H}(\sigma)/(k_\mathrm{B}T_j)+g_j\right)}{\exp\left(-\mathcal{H}(\sigma)/(k_\mathrm{B}T_i)+g_i\right)} \\ &= \exp(-(\beta_j-\beta_i)\mathcal{H}(\sigma)+(g_j-g_i))\,.\end{aligned} \tag{15.22}$$

  Thus, the transition probability can be chosen in a Metropolis-like criterion as

$$p((\sigma,i)\to(\sigma,j)) = \begin{cases} 1 & \text{if } \Delta \le 0\,, \\ \exp(-\Delta) & \text{otherwise}\,, \end{cases} \tag{15.23}$$

  with

$$\Delta = (\beta_j-\beta_i)\mathcal{H}(\sigma)-(g_j-g_i)\,. \tag{15.24}$$

  As the acceptance probability decreases exponentially with the difference between the inverse temperatures β_i and β_j, one usually only chooses moves $T_i \to T_{j=i\pm1}$, i. e., to the neighboring temperature values.
- Of course, these two moves can also be combined in one move $(\sigma,i)\to(\tau,j)$. One can derive a Metropolis-like criterion as (15.23) but with

$$\Delta = (\beta_j\mathcal{H}(\tau)-\beta_i\mathcal{H}(\sigma))-(g_j-g_i)\,. \tag{15.25}$$

  Usually, however, one only uses the two moves above.

However, the main question still remains as to how to choose the g_i parameters. They are not a priori known and are usually determined by iterations of simulations that can be rather difficult for complex systems. One of the simplest approaches is first to determine the thermal expectation values $\langle\mathcal{H}\rangle(T_i)$ for each temperature by an ordinary SA run. Then the differences $g_{i\pm1}-g_i$ are given by the differences $\langle\mathcal{H}\rangle(T_{i\pm1})-\langle\mathcal{H}\rangle(T_i)$.

Thus, the outline of ST is as follows:

1. First, determine the g_i parameters.
2. Then start the simulation at some temperature T_i, preferably a large one.
3. Perform a few Monte Carlo sweeps at the given temperature T_i.
4. Then, try a move $T_i \to T_{i\pm1}$ with the transition probability Eq. (15.23).
5. If some final condition is not fulfilled, return to step 3.

Note that in contrast to SA, the system of ST does not need any time to equilibrate at the new temperature as the temperature is also changed with a Metropolis-like criterion.

A related method to overcome the difficulty of determining the g_i parameters is the replica-exchange method (REM) [98, 99], which is also called parallel tempering (PT) [41]. It was developed as an extension of ST. Again a set of M temperatures T_i with $T_1 < T_2 < \cdots < T_M$ is considered. In contrast to SA and ST, one has M different Markov chains, one at each temperature T_i. At each of these constant temperatures, a simulation as with SA at the corresponding temperature T_i is performed; thus each move is accepted according to the Metropolis criterion. The most important point of PT is that these simulations are performed independently of each other.

Thus, after some time, one has M configurations σ_i. The probability $\pi(\sigma)$ is given by the usual Boltzmann weight $\exp(-\mathcal{H}(\sigma_i)/(k_\mathrm{B}T_i))/Z(T_i)$. In contrast to ST, PT considers not the joint configuration space $\Gamma \times \{1, \ldots, M\}$ but the configuration space $\Gamma \times \Gamma \times \cdots \times \Gamma = \Gamma^M$. The probability $\mathcal{P}$ of the product state $\mathcal{S} = \sigma_1 \times \sigma_2 \times \cdots \times \sigma_M$ is given as the product of the Boltzmann weights of the single states due to the independence of the simulations, i. e.,

$$\begin{aligned} \mathcal{P}(\mathcal{S}) &= \mathcal{P}(\sigma_1, \sigma_2, \ldots, \sigma_M) \\ &= \prod_{i=1}^{M} \pi_\mathrm{equ}(\sigma_i) \\ &= \frac{\exp(-\mathcal{H}(\sigma_1)/(k_\mathrm{B}T_1))}{Z(T_1)} \times \cdots \times \frac{\exp(-\mathcal{H}(\sigma_M)/(k_\mathrm{B}T_M))}{Z(T_M)} \end{aligned} \tag{15.26}$$

with $Z(T_i)$ being the partition sum at the temperature T_i.

The new point of PT now is that it introduces a varied type of the move of ST to change the temperature: the temperatures at which two configurations σ_i and σ_j are exchanged by exchanging these two configurations between the corresponding Markov chains at the temperatures T_i and T_j. This means: after having applied a sequence of moves in process 1 at temperature T_1 with the final configuration σ_1, a sequence of moves in process 2 at T_2 with the final σ_2, and analogously in the other processes, one randomly selects two processes i and j and wants to move the configuration σ_i to process j while shifting the configuration σ_j to process i. The question is now with what probability this move

$$\begin{aligned} \mathcal{S}_1 &= (\sigma_1, \ldots, \sigma_i, \ldots, \sigma_j, \ldots, \sigma_M) \\ \rightarrow \mathcal{S}_2 &= (\sigma_1, \ldots, \sigma_j, \ldots, \sigma_i, \ldots, \sigma_M) \end{aligned} \tag{15.27}$$

will be accepted. Of course, again the detailed balance condition will be fulfilled, such that

$$\mathcal{P}(\mathcal{S}_1) \times p(\mathcal{S}_1 \rightarrow \mathcal{S}_2) = \mathcal{P}(\mathcal{S}_2) \times p(\mathcal{S}_2 \rightarrow \mathcal{S}_1)\,. \tag{15.28}$$

Thus, one gets the relation

$$\begin{aligned}
\frac{p(\mathcal{S}_1 \to \mathcal{S}_2)}{p(\mathcal{S}_2 \to \mathcal{S}_1)} &= \frac{\mathcal{P}(\mathcal{S}_2)}{\mathcal{P}(\mathcal{S}_1)} \\
&= \frac{\pi_{\mathrm{equ}}(\sigma_j, T_i) \times \pi_{\mathrm{equ}}(\sigma_i, T_j)}{\pi_{\mathrm{equ}}(\sigma_i, T_i) \times \pi_{\mathrm{equ}}(\sigma_j, T_j)} \\
&= \frac{\exp(-\mathcal{H}(\sigma_j)/(k_{\mathrm{B}}T_i)) \times \exp(-\mathcal{H}(\sigma_i)/(k_{\mathrm{B}}T_j))}{\exp(-\mathcal{H}(\sigma_i)/(k_{\mathrm{B}}T_i)) \times \exp(-\mathcal{H}(\sigma_j)/(k_{\mathrm{B}}T_j))} \\
&= \exp(-\Delta\mathcal{H}/(k_{\mathrm{B}}T_i)) \times \exp(\Delta\mathcal{H}/(k_{\mathrm{B}}T_j)) \\
&= \exp(-\beta_i \Delta\mathcal{H}) \times \exp(\beta_j \Delta\mathcal{H}) \\
&= \exp(-(\beta_i - \beta_j)(\mathcal{H}(\sigma_j) - \mathcal{H}(\sigma_i))) \\
&= \exp(-\Delta\beta \times \Delta\mathcal{H})\,,
\end{aligned} \tag{15.29}$$

with $\Delta\mathcal{H} = \mathcal{H}(\sigma_j) - \mathcal{H}(\sigma_i)$ and $\Delta\beta = \beta_i - \beta_j$.

As in the derivation of the Metropolis criterion, there is some arbitrariness in the explicit choice of the transition probability. Here one can use a Metropolis-like acceptance criterion:

$$p(\mathcal{S}_1 \to \mathcal{S}_2) = \begin{cases} 1 & \text{if } \Delta\beta \times \Delta\mathcal{H} \leq 0\,, \\ \exp(-\Delta\beta \times \Delta\mathcal{H}) & \text{otherwise}\,. \end{cases} \tag{15.30}$$

Thus, if σ_j is better than σ_i, i. e., if $\mathcal{H}(\sigma_j) < \mathcal{H}(\sigma_i)$, these two configurations are always exchanged if $\beta_i > \beta_j$ and thus if $T_i < T_j$. The energetically lower configuration is thus with larger probability put to the smaller temperature and thus cooled down. The various temperatures T_i can stay constant all the time as the cooling can be achieved by the PT exchange mechanism, but one can also lower the M temperatures T_i during the simulation run.

Usually, a restriction similar to the one in ST is applied to this exchange process: as the transition probability decreases exponentially with the difference between the inverse temperatures, only exchanges of configurations are performed between neighboring temperature pairs (T_i, T_{i+1}) [98]. The individual temperature values have to be distributed in such a way that the smallest temperature is in a range where the system is already frozen, whereas the highest temperature should be well above the peak of the specific heat.

Like the problem of determining a good cooling schedule for the ordinary SA, one now has, both for ST and PT, the problem of what temperatures to use. Good results can be achieved by concentrating the temperature values around the peak of the specific heat of the problem.

There are various elaborate ways to determine these temperature values. An iterative way was proposed, e. g., by Kerler and Rehberg [109]: first, they

fix a minimum β_1 and a maximum β_M. Then they start with β_i, which are equally distributed between these marginal values and measure the stay time τ_i at each β_i, which is the time between two accepted exchange moves at the corresponding β_i. The time for β_1 and β_M is divided by two, as these processes have only one neighboring process. Next they calculate the auxiliary variables

$$a_i = (\beta_{i+1} - \beta_i)/(\tau_{i+1} + \tau_i) \tag{15.31}$$

and their sum $A = \sum a_i$. Then they set the β_i to new values,

$$\beta_i^{\text{new}} = \begin{cases} \beta_1 & \text{if } i = 1\,, \\ \beta_{i-1}^{\text{new}} + \dfrac{a_{i-1}}{A}(\beta_M - \beta_1) & \text{otherwise}\,. \end{cases} \tag{15.32}$$

This procedure is iteratively repeated until the values of β_i have converged to some fixed points.

The individual implementations of these algorithms differ not only in the techniques for choosing these inverse temperature values but also in whether the individual values are kept constant during the optimization run or decreased. A cooling schedule for these temperature values can be either rather simple, e. g., the temperatures can be decreased exponentially, but they can also make use of acceptance criteria like those above [98].

The advantage of PT compared with ST is that there are no parameters g_i that have to be determined either beforehand or during the simulation process. Furthermore, one can explore the configuration space in one run much more thoroughly as there are several Markov chains. However, simulating M Monte Carlo walkers walking around obviously costs M times the compute effort of simulating only one. However, this is not necessarily a disadvantage as PT is very well suited for parallel enablement: the single processes can be run on the single nodes without interaction, except for the exchange moves that are sometimes performed.