

1 Parallel tempering

1.1 General considerations

Umbrella sampling and simulated tempering have a serious shortcoming. One needs to optimize the coefficients α_i and this requires the computation of partition function ratios. Here we introduce a different approach that is called in a variety of different ways: *Metropolis-coupled Markov chain Monte Carlo*, *replica-exchange algorithm*, *parallel tempering method*. Mathematicians prefer to indicate it as *swapping algorithm*. *It does not require the computation of partition function ratios.* The parallel-tempering algorithm is an algorithm that works in a state space S' which consists in R replicas of the original state space S : $S' = S \times \dots \times S$. On S' one takes as probability

$$\Pi(x_1, \dots, x_R) = \frac{e^{-\beta_1 H(x_1)}}{Z_{\beta_1}} \frac{e^{-\beta_2 H(x_2)}}{Z_{\beta_2}} \dots \frac{e^{-\beta_R H(x_R)}}{Z_{\beta_R}}$$

The probability Π is simply the product of the canonical-ensemble probabilities at the different temperatures. Obtaining canonical-ensemble averages is therefore straightforward:

$$\langle A(x) \rangle_{\beta_i} = \sum_{x_1, \dots, x_N} A(x_i) \Pi(x_1, \dots, x_R)$$

Stated differently, to compute averages at inverse temperature β_i , we simply consider the configurations x_i that are distributed with the correct canonical distribution. The proof of this relation is trivial. Indeed, the right-hand side is

$$\sum_{x_1, \dots, x_N} A(x_i) \frac{e^{-\beta_1 H(x_1)}}{Z_{\beta_1}} \dots \frac{e^{-\beta_R H(x_R)}}{Z_{\beta_R}} = \left(\sum_{x_1} \frac{e^{-\beta_1 H(x_1)}}{Z_{\beta_1}} \right) \left(\sum_{x_2} \frac{e^{-\beta_2 H(x_2)}}{Z_{\beta_2}} \right) \dots \left(\sum_{x_i} A(x_i) \frac{e^{-\beta_i H(x_i)}}{Z_{\beta_i}} \right) \dots \left(\sum_{x_R} \frac{e^{-\beta_R H(x_R)}}{Z_{\beta_R}} \right). \quad (1)$$

All sums give 1, except the sum over x_i that gives $\langle A(x) \rangle_{\beta_i}$.

The algorithm considers two different types of moves: 1) canonical moves on the systems (we change the configurations x_1 , x_2 , and so on; 2) swapping moves. Note that if we only use canonical moves on the systems (moves of type 1) we would also obtain a valid algorithm. It corresponds to a simulation in which the R systems are simulated together, but independently. This type of simulations are sometimes performed on parallel machines and they correspond to the most trivial parallelization of the simulation.

The swapping moves are the moves that characterize the parallel tempering method. They are implemented as follows:

- a) Choose two replicas i , $i + 1$ randomly (we randomly choose i in $[0, R - 1]$). The proposal matrix is clearly symmetric.
- b) We propose the swap

$$(x_1, \dots, x_i, x_{i+1}, \dots, x_R) \rightarrow (x_1, \dots, x_{i+1}, x_i, \dots, x_R) .$$

Since the proposal matrix is symmetric the acceptance is

$$A_{\text{swap}} = \min \left(1, \frac{e^{-\beta_i E_{i+1} - \beta_{i+1} E_i}}{e^{-\beta_{i+1} E_{i+1} - \beta_i E_i}} \right) = \min \left(1, e^{(\beta_{i+1} - \beta_i)(E_{i+1} - E_i)} \right),$$

where $E_i = H(x_i)$.

As we have already discussed several times, to define the algorithm we consider a random choice for the pair. However, it is also possible to choose the pair sequentially: first, we propose a swap of replicas 1 and 2, then of replicas 2 and 3, and so on. The sequential algorithm is also correct, although it does not satisfy detailed balance (it satisfies the stationarity condition).

As in the US or ST case, in order to perform a PT simulation, one must decide the number R of inverse temperatures and their values. As for the US and ST methods, temperatures should be close enough, so that the typical configuration domains at nearby temperatures overlap.

In the formal setting, the algorithm is defined as an algorithm in which the configurations are swapped. However, it is more natural to think of the algorithm as one in which the temperatures are swapped. We run standard canonical simulations on R different systems and the swapping move corresponds to a temperature swap. In this interpretation, the parallel tempering algorithm represents a generalization of the simulated tempering method. In this framework, temperatures “follow” the configurations. In practice one defines a vector B of length R . At the beginning of the simulation one sets $B = (\beta_1, \beta_2, \beta_3, \dots, \beta_R)$; in C-notation $B[0] = \beta_1$, $H[1] = \beta_2$, and so on. Then, whenever one performs a swap on the pair $i, i + 1$, one does the following:

```
a1 = B[i-1]; a2 = B[i];
B[i-1] = a2; B[i] = a1;
```

At each time the variable B gives the current inverse temperature: $B[0]$ gives the current inverse temperature of system 1; $B[1]$ gives the current temperature of system 2 and so on.

Whenever a PT run is performed, it is important to verify that the algorithm is working correctly. The simplest quantity to measure is the swapping rate $a_{i,i+1}$ between adjacent temperatures, that is the fraction of accepted swaps. The algorithm works efficiently only if, for all i , $a_{i,i+1}$ is not too small. The optimal value for $a_{i,i+1}$ lies between 0.2 and 0.3, but larger, or slightly smaller values, although not optimal, are still acceptable. A reasonable swapping rate is, however, not enough to guarantee that the algorithm is working correctly. Indeed, there are situations in which the swapping rates take the desired values, but the PT simulation is inefficient. This typically occurs when there is a “bottleneck” at a certain temperature β_K (typically, this occurs close to phase transitions). In this case $a_{K-1,K}$ and $a_{K,K+1}$ are both reasonable, but the algorithm is unable to modify a configuration that is typical at β_{K+1} into one that is typical at β_{K-1} . To investigate this issue, it is useful to define the average round-trip time, i.e. the Monte Carlo time between occurrences of $B[i] = \beta_1$ and $B[i] = \beta_R$. If the swapping procedure is working efficiently, the round-trip time should be comparable to the return time of a random walker moving among temperatures with the swapping rates actually measured in the simulation. On the contrary, if the swapping procedure has a bottleneck, then the round-trip time becomes large and is essentially controlled by the time it takes for a replica to go through the bottleneck.