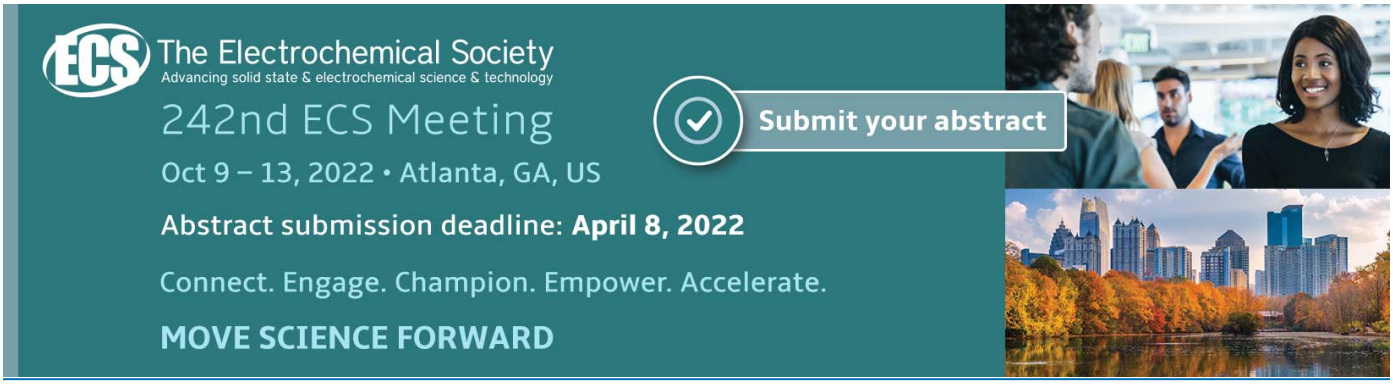
Resampling schemes in population annealing — numerical results



Journal of Physics: Conference Series **2207** (2022) 012012 doi:10.1088/1742-6596/2207/1/012012

Resampling schemes in population annealing – numerical results

Denis Gessert[1,2], **Martin Weigel**[1,3] **and Wolfhard Janke**[2]

[1] Centre for Fluid and Complex Systems, Coventry University, Coventry CV1 5FB, UK
[2] Institut für Theoretische Physik, Universität Leipzig, IPF 231101, 04081 Leipzig, Germany
[3] Institut für Physik, Technische Universität Chemnitz, 09107 Chemnitz, Germany

E-mail: gessertd@uni.coventry.ac.uk

Abstract. Population annealing (PA) is a population-based algorithm that is designed for equilibrium simulations of thermodynamic systems with a rough free energy landscape. It is known to be more efficient in doing so than standard Markov chain Monte Carlo alone. The algorithm has a number of parameters that can be fine-tuned to improve performance. While there is some theoretical and numerical work regarding most of these parameters, there appears to be a gap in the literature concerning the role of resampling in PA. Here, we present a numerical comparison of a number of resampling schemes for PA simulations of the 2D Ising model.

1. Introduction

Systems with a rough free energy landscape, such as spin glasses, structural glasses and some polymers, to name a few, appear in many fields of research and are notoriously hard to simulate [1]. Over the last decades a number of methods have been developed with the aim of making such systems more accessible such as, e.g., multicanonical simulations [2], the Wang-Landau algorithm [3], parallel tempering [4] and the population annealing (PA) algorithm [5, 6] which is the subject of the present work.

While there is no clear-cut answer to the question which of these methods is best and the answer may depend on the system considered, we motivate the use of PA as follows: First, PA is well suited for the highly parallel execution on supercomputers, as the number of replicas is only limited by the underlying computing architecture and not the method itself. In contrast, in parallel tempering the level of parallelization is limited by the number of temperature points considered which is bounded by increasing round-trip times as the number of replicas grows. Additionally, with increasing population size not just the statistical error but also the bias is systematically reduced [7]. Second, as part of the PA algorithm an equilibration routine is run at each temperature, which typically consists of a number of Markov chain Monte Carlo (MCMC) sweeps (MCS). However, the type of update is not specified and can freely be replaced by, e.g., a cluster algorithm, a rejection-free update or even by molecular dynamics [8]. This allows for an easy adaptation of PA to the system under study, and any algorithmic improvement in MCMC immediately translates to PA.

The fundamental idea of PA, first presented by Hukushima and Iba [5, 6], is to perform parallel simulated annealing (SA) simulations (replicas) subject to a population-control step at each temperature. It is worth mentioning that thanks to this control step PA is an equilibrium

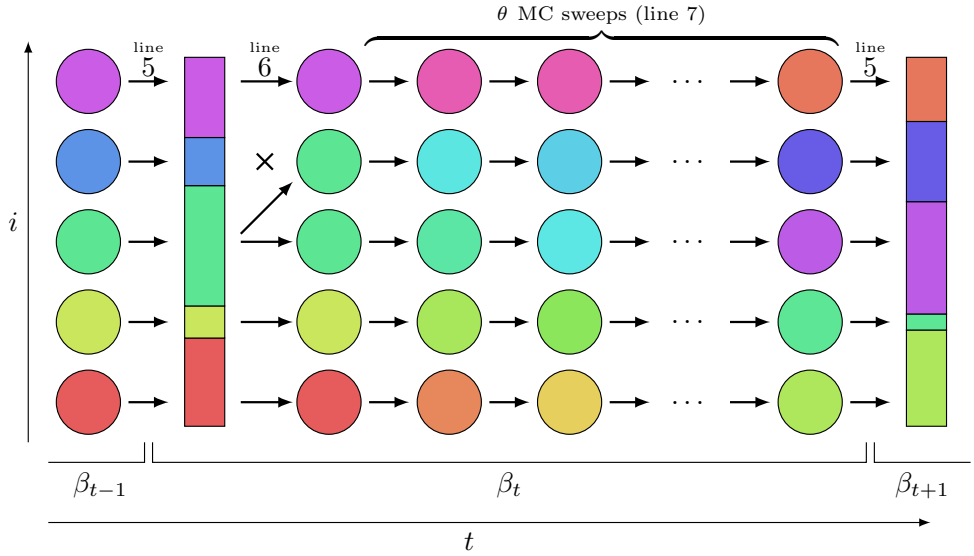


Figure 1. Schematic of the population annealing algorithm. Vertical axis corresponds to R replicas (here five) and horizontal axis to simulation time. Line numbers correspond to algorithm 1 in Appendix A.

method for calculating canonical averages although SA itself by default is not. The population control is implemented by a resampling phase employed at each temperature step (or in principle after M steps, but researchers have focused on $M = 1$) and is crucial to the performance of PA. Despite its importance to the success of the algorithm, it has received little attention in the literature – the effect of different resampling methods has not yet been systematically studied and particularly the fact that resampling introduces additional artificial noise into the simulations has not been discussed.

We use the two-dimensional ferromagnetic Ising model in zero field corresponding to the Hamiltonian

$$\mathcal{H} = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j \tag{1}$$

where the sum is over nearest neighbors $\langle ij \rangle$ only. In our simulations we consider lattices of size 64×64 with periodic boundary conditions. Our implementation of the PA algorithm is based on the Open Source GPU code of Ref. [9] that is available online. The code uses a checkerboard decomposed Metropolis update as an equilibration routine. All resampling methods were also implemented on GPUs.

2. Simulation method

2.1. Population annealing

In PA a population of $R_0 = R$ replicas is initialized at the starting inverse temperature β_0, chosen to be zero if possible; here, R is the target population size. What follows is an iteration scheme (see Figure 1) in which in each iteration the inverse temperature β_t is increased to $\beta_{t+1} = \beta_t + \Delta\beta$. The step $\Delta\beta$ has to be chosen appropriately and can in principle be adaptive [9, 10].

At the end of iteration $t - 1$, the population consists of R_{t-1} replicas at β_{t-1}. Following the line of thought of histogram reweighting [11], if we were asked to estimate the energy histogram at β_t based on the population at β_{t-1}, we would attribute a weight $w_i \propto \exp(-\Delta\beta E_i)$ to each replica i with energy E_i where the weights are normalized such that they sum to one. Next, the population is resampled according to these weights w_i, while attempting to keep the population

size close to the target size R. In practice, the number of descendants r_i of replica i is, on average,

$$\langle r_i \rangle = \hat{\tau}_i = w_i R. \tag{2}$$

Clearly, this only specifies the mean of r_i and the exact way how r_i is determined depends on the used resampling method. Note that it is common that $r_i = 0$ for some replicas, in which case they are culled while others are replicated multiple times.

Immediately after resampling, the population is potentially highly correlated as it may contain identical configurations. To reduce the correlation among replicas, an equilibration routine is used (here a checkerboard decomposed version of the Metropolis algorithm, see Ref. [9] for details). After θ MCS, population averages for observables of interest are calculated. The scheme then proceeds with the iteration for inverse temperature β_{t+1} until the final temperature is reached.

For a pseudocode implementation of the PA algorithm we refer to Appendix A.

2.2. Resampling methods

When the number of copies of one replica r_i is chosen only based on the value of $\hat{\tau}_i$ of the same replica i, the resampling method is completely described by a univariate distribution $P_{\hat{\tau}_i}(r_i = k)$. In this case the sum of all r_i's is a random variable and thus the population size fluctuates with time. In this group we consider "nearest integer" resampling [12] and Poisson resampling [13] given by

$$P_{\hat{\tau}_i}(r_i = k) = \begin{cases} \hat{\tau}_i - \lceil \hat{\tau}_i \rceil & \text{if } k = \lceil \hat{\tau}_i \rceil \\ 1 - (\hat{\tau}_i - \lceil \hat{\tau}_i \rceil) & \text{if } k = \lfloor \hat{\tau}_i \rfloor \\ 0 & \text{else} \end{cases} \tag{3}$$

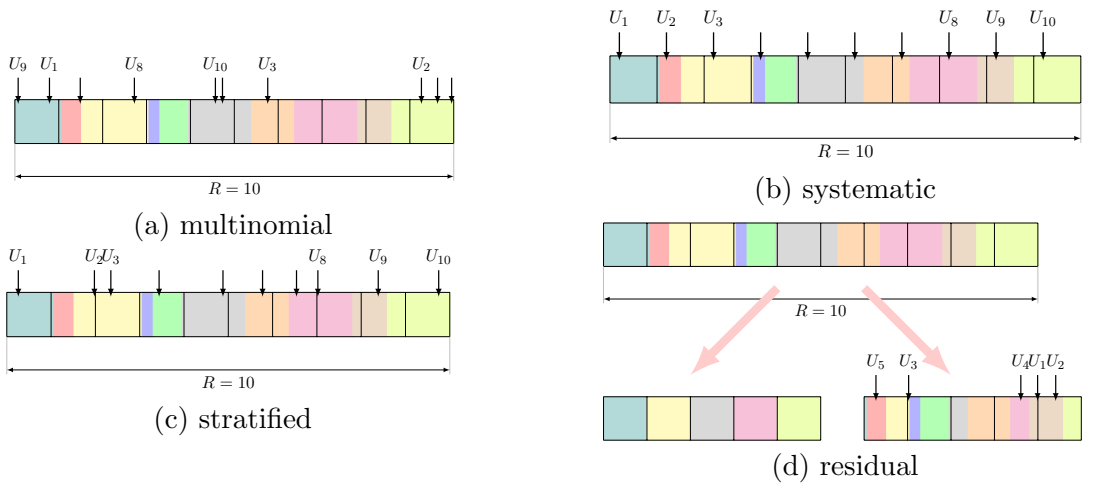


Figure 2. Visualization of various population size preserving resampling methods. Colored boxes correspond to different replicas i and their lengths are proportional to $\hat{\tau}_i$. The number of copies of replica i is determined by the number of arrows in box i. (a) - (c) R arrows are distributed over the interval $[0, R]$. (a) multinomial: Each arrow is uniformly distributed over the interval $[0, R]$. (b) systematic: The first arrow is uniformly distributed within $[0, 1]$. The remaining $R - 1$ arrows are placed with unit spacing. (c) stratified: In each square exactly one arrow is placed with uniform probability, i.e., $U_i \sim \mathcal{U}([i-1, i])$. (d) residual: At first each replica is copied $\lfloor \hat{\tau}_i \rfloor$ times. The population is brought to its original size by multinomially drawing from the residuals, i.e., performing (a) with $\tau_i - \lfloor \tau_i \rfloor$ replacing τ_i.

for nearest-integer and

$$P_{\hat{\tau}_i}(r_i = k) = \frac{\hat{\tau}_i^k}{k!}\, e^{-\hat{\tau}_i} \tag{4}$$

for Poisson, respectively. Here, $\lceil x \rceil$ denotes the smallest integer that is larger than or equal to x, i.e., rounding up, and similarly $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x, i.e., rounding down.

Vice versa, if one requires the population size to be fixed, the set $\{r_i\}$ is drawn at once and the individual r_i's may show some correlation. See Figure 2 for a visualization of the population size preserving resampling methods used here and the caption for explanations. These methods are quite well known in the field of particle filters [14] whereas within the PA community multinomial resampling is the only population size preserving method used so far.

In principle, one might come up with many more methods such as geometric resampling. Here we restrict ourselves to methods with sampling variance less than or equal to that of the widely used multinomial resampling.

2.3. Benchmarking quantities

The most obvious quantities to use for comparing the algorithmic performance are systematic and statistical errors. We looked at various observables and present here the errors measured for the specific heat C_v, as this is where we found the strongest differences among the methods.

Additionally, Ref. [12] introduced population quantities designed to identify the maximal correlation within the population introduced through the resampling step, i.e.,

$$\rho_t = R\sum_{i=1}^{R} \mathfrak{n}_i^2, \qquad \rho_s = R\exp\left(\sum_{i=1}^{R} \mathfrak{n}_i \ln \mathfrak{n}_i\right) \quad \text{and} \quad f = \sum_{i=1} \min\{1, R\mathfrak{n}_i\}, \tag{5}$$

where $\mathfrak{n}_i$ is the fraction of replicas descending from the initial replica i, ρ_t is known as the mean square family size, ρ_s as the entropic family size and f is the number of families. As the authors of Ref. [12] show, these quantities are closely related to each other.

3. Results

The following results were obtained from simulations of the Ising model on a square lattice with periodic boundary conditions of dimensions 64×64. The numerical data presented here was attained through independent PA simulations for each resampling method and was averaged over 5 000 simulations per method. The target population size is $R = 20\,000$ and the annealing schedule was chosen to be $\beta_k = k/300$ with $k \in \{0, \ldots, 299\}$. The choice of $\beta_0 = 0$ allows us to initialize each spin configuration simply by choosing every spin as up or down with equal probabilities which can easily be implemented. At each temperature $\theta = 5$ MCS were carried out.

Figure 3 shows bias and statistical error immediately after resampling (before the equilibration routine) of the specific heat. We calculate the bias by using the known exact solution [15, 16], whereas the statistical errors were obtained by estimating the standard deviation of independent runs. Both quantities peak around the critical temperature. The bias shows little deviation among the methods, although multinomial and Poisson resampling appear to have slightly stronger bias in the vicinity of the critical temperature. Around the critical temperature statistical errors are almost indistinguishable and away from criticality the statistical error using multinomial and Poisson resampling is significantly larger compared to the other methods. The inset, which shows all errors normalized to the nearest-integer errors, illustrates that Poisson and multinomial resampling have statistical errors of up to 25% higher than nearest-integer resampling and outside the depicted near-critical regime these even exceed 30%. We argue that near criticality fluctuations intrinsic to the model are very strong

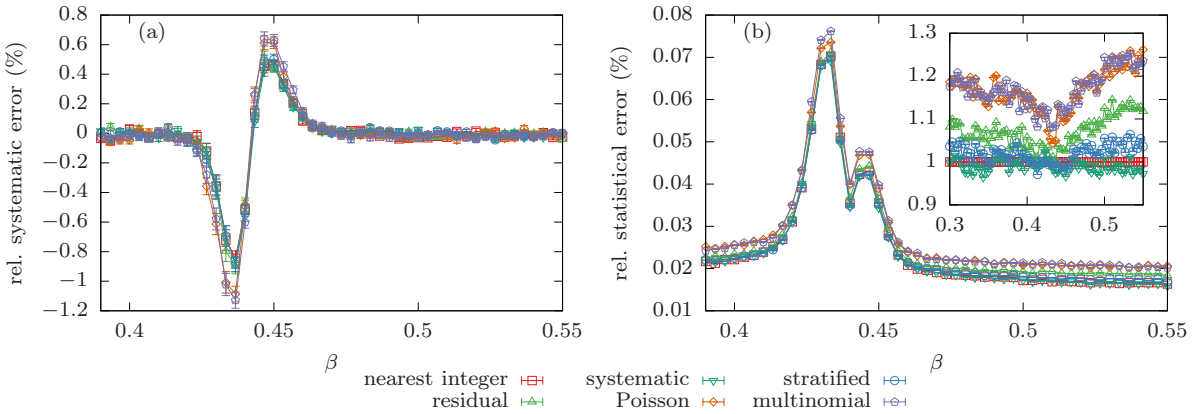


Figure 3. Relative (a) systematic and (b) statistical error of the specific heat measured after the resampling step for various resampling methods. (a) Systematic errors hardly differ between different resampling schemes. (b) The inset shows the statistical error relative to the error using nearest-integer resampling. Away from criticality, the curves differ significantly whereas around $\beta_c \approx 0.44$ the choice of the resampling method appears to have little effect on the statistical error.

and dominate the observed statistical errors whereas further away the resampling noise has a significant effect.

The family quantity ρ_t is shown in Figure 4(a). It increases rapidly around the critical point and slowly otherwise. It agrees with the previous observations in the sense that again multinomial and Poisson resampling stick out and ρ_t consistently takes its highest values for these two methods. Larger average family size typically corresponds to more correlation within the population and thus poorer statistics. The same observations are made for the entropic family size ρ_s and the inverse of the number of families R/f.

4. Conclusion

We have demonstrated that resampling has a significant effect on the data obtained through population annealing, and we see a difference in error bars of up to 30% using standard resampling methods. Furthermore, we identified that resampling matters most away from criticality when it contributes as much or more to the statistical error as the fluctuations intrinsic to the system. Through the study of family quantities we found that higher variance in r_i at

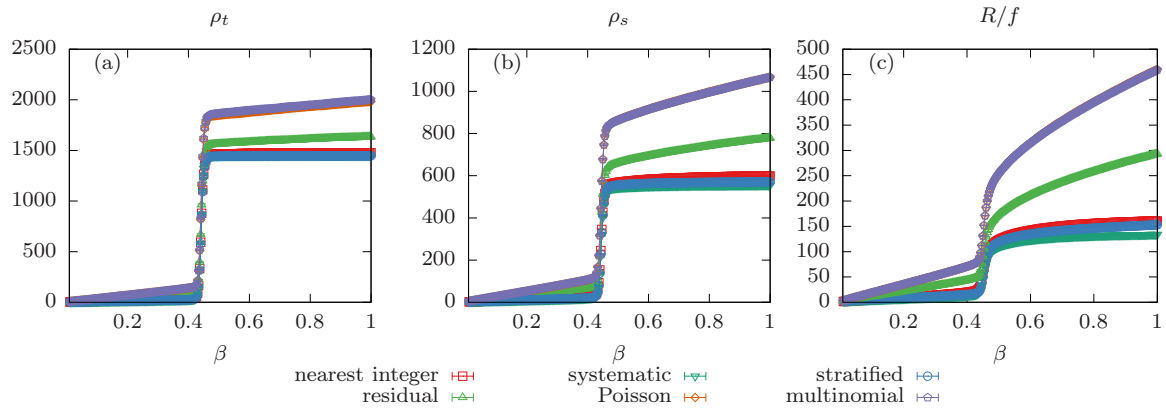


Figure 4. Family quantities (a) ρ_t, (b) ρ_s and (c) R/f for various resampling methods.

high temperatures causes fewer families of larger sizes surviving, thus causing more correlation in the population.

In a quest of finding the best resampling method nearest-integer resampling is the front runner due to it consistently outperforming the other methods and due to its simple implementation. Systematic resampling is also a good choice, in particular if a constant population size is desired, as it provides similar performance although its implementation is slightly more involved. In particular, we note that it performs significantly better than the multinomial method usually used for this purpose.

Acknowledgments

The project was supported by the Deutsch-Französische Hochschule (DFH-UFA) through the Doctoral College "$\mathbb{L}^4$" under Grant No. CDFA-02-07.

Appendix A. Population Annealing algorithm

Algorithm 1 Standard Population Annealing algorithm (introduced in Ref. [6])

Require: R,$\{\beta_k\}$,θ, M

1: $k \leftarrow 0$
2: Initialize population of R independent replicas at β_0
3: **while** $\beta_{k+1} < \beta_{\max}$ **do**
4: $k \leftarrow t + 1$
5: Calculate **Boltzmann weights** w_i of the replicas at β_t
6: **if** $k \equiv 0 (\mathrm{mod}\ M)$ then **resample** population according to weights
(on avg. $\tau_i = Rw_i$ copies of replica i)
7: **Monte Carlo update** of the replicas (θ MC sweeps at β_k)
8: **Measure** observables $\mathcal{O}$

Algorithm 1 shows the standard population annealing algorithm as originally devised by Hukushima and Iba [6]. The parameters are the target population size R (throughout the simulation R_i might fluctuate around R), the annealing schedule $\{\beta_k\}$, the number of equilibration updates θ (typically MCMC) between each temperature step and M, the resampling interval (usually $M = 1$).

References

[1] Janke W (ed) 2007 *Rugged Free Energy Landscapes – Common Computational Approaches to Spin Glasses, Structural Glasses and Biological Macromolecules* (*Lecture Notes in Physics* vol 736) (Berlin: Springer)
[2] Berg B A and Neuhaus T 1991 *Phys. Lett. B* **267** 249–53
[3] Wang F and Landau D P 2001 *Phys. Rev. Lett.* **86** 2050–3
[4] Hukushima K and Nemoto K 1996 *J. Phys. Soc. Jpn.* **65** 1604–8
[5] Iba Y 2001 *Trans. Jpn. Soc. Artif. Intell.* **16** 279–86
[6] Hukushima K and Iba Y 2003 *AIP Conf. Proc.* **690** 200–6
[7] Weigel M, Barash L, Shchur L and Janke W 2021 *Phys. Rev. E* **103** 053301
[8] Christiansen H, Weigel M and Janke W 2019 *Phys. Rev. Lett.* **122** 060602
[9] Barash L Y, Weigel M, Borovský M, Janke W and Shchur L N 2017 *Comput. Phys. Commun.* **220** 341–50
[10] Christiansen H, Weigel M and Janke W 2019 *J. Phys.: Conf. Ser.* **1163** 012074
[11] Ferrenberg A M and Swendsen R H 1988 *Phys. Rev. Lett.* **61** 2635–8
[12] Wang W, Machta J and Katzgraber H G 2015 *Phys. Rev. E* **92** 063307
[13] Machta J 2010 *Phys. Rev. E* **82** 026704
[14] Doucet A and Johansen A M 2011 *Handbook of Nonlinear Filtering* Crisan D and Rozovskii B (eds) (New York: Oxford University Press) chap 8.2, pp 656–704
[15] Beale P D 1996 *Phys. Rev. Lett.* **76** 78–81
[16] Kaufman B 1949 *Phys. Rev.* **76** 1232–43