# New Optimization Algorithms in Physics

Edited by
Alexander K. Hartmann and Heiko Rieger


WILEY-VCH Verlag GmbH & Co. KGaA

# New Optimization Algorithms in Physics

*Edited by*

*Alexander K. Hartmann and Heiko Rieger*

# New Optimization Algorithms in Physics

Edited by
Alexander K. Hartmann and Heiko Rieger


WILEY-VCH Verlag GmbH & Co. KGaA

**Editors**

*Alexander K. Hartmann*
Universität Göttingen, Germany
hartmann@theorie.physik.uni-goettingen.de

*Heiko Rieger*
Universität des Saarlandes, Germany
h.rieger@mx-uni-saarland.de

**Cover Picture**
Short artificial peptide (U. H. E. Hansmann); Dense configuration of polydisperse hard discs (W. Krauth); Cut Polytope for the complete graph with N=3 (F. Liers et al); Factor graph for 3-SAT (R. Zecchina)



**Library of Congress Card No.: applied for**
**British Library Cataloging-in-Publication Data:**
A catalogue record for this book is available from the British Library

**Bibliographic information published by**
**Die Deutsche Bibliothek**
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.



Printed in the Federal Republic of Germany
Printed on acid-free paper

**Composition** Uwe Krieg, Berlin
**Printing** Strauss GmbH, Mörlenbach
**Bookbinding** Litges & Dopf Buchbinderei GmbH, Heppenheim
**ISBN** 3-527-40406-6

# Contents

# List of Contributors


- *Jean-Christian Anglès d'Auriac,* Ch. 6

  Centre de Recherches sur les Tres Basses Temperatures
  BP 166, F-38042 Grenoble, France
  e-mail: dauriac@grenoble.cnrs.fr

- *Stefan Boettcher,* Ch. 11

  Stefan Boettcher
  Physics Department
  Emory University
  Atlanta, GA 30322; USA
  e-mail: sboettc@emory.edu
  www.physics.emory.edu/faculty/boettcher

- *Simona Cocco,* Ch. 8

  Laboratoire de Dynamique des Fluides Complexes
  3 rue de l'Université
  67000 Strasbourg, France
  e-mail: cocco@ldfc.u-strasbg.fr

- *Liat Ein-Dor,* Ch. 8

  Department of Physics of Complex Systems
  Wezmann Institute
  Rehovot 76100, Israel

  and

  Laboratoire de Physique Théorique de l'ENS
  24 rue Lhomond
  75005 Paris, France

- *Ulrich H. E. Hansmann,* Ch. 13

  Department of Physics
  Michigan Technological University
  Houghton
  MI 49931-1295, USA
  e-mail: hansmann@mtu.edu

- *Alexander K. Hartmann,* Ch. 1, 12

  Institut für Theoretische Physik
  Universität Göttingen
  Tammanstraße 1
  D-37077 Göttingen, Germany
  e-mail:
  hartmann@theorie.physik.uni-goettingen.de

- *Michael Jünger,* Ch. 4

  Institut für Informatik
  Universität zu Köln
  Pohligstraße 1
  D-50969 Köln, Germany
  e-mail: mjuenger@informatik.uni-koeln.de

- *Werner Krauth,* Ch. 2

  CNRS-Laboratoire de Physique Statistique
  Ecole Normale Supérieure
  24, rue Lhomond
  F-75231 Paris Cedex 05, France
  e-mail: Werner.Krauth@ens.fr

- *Frauke Liers,* Ch. 4

  Institut für Informatik
  Universität zu Köln
  Pohligstraße 1
  D-50969 Köln, Germany
  e-mail: liers@informatik.uni-koeln.de

• *Olivier C. Martin,* Ch. 3

LPTMS, Université Paris-Sud
Orsay Cedex 91405, France
e-mail: martino@ipno.in2p3.fr

• *A. Alan Middleton,* Ch. 5

Department of Physics
Syracuse University
Syracuse, NY, USA
e-mail: aam@syr.edu

• *Remi Monasson,* Ch. 8

Laboratoire de Physique Théorique de l'ENS
24 rue Lhomond
75005 Paris, France
e-mail: monasson@lpt.ens.fr

and

Laboratoire de Physique Théorique
3 rue de l'Université
67000 Strasbourg, France

• *Károly F. Pál,* Ch. 10

Department of Theoretical Physics
Institute of Nuclear Research of the
Hungarian Academy of Sciences
Bem tér 18/c
H-4026 Debrecen, Hungary
e-mail: kfpal@hal.atomki.hu

• *Gerhard Reinelt,* Ch. 4

Institut für Informatik
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 368
D-69120 Heidelberg, Germany
e-mail:
Gerhard.Reinelt@informatik.uni-heidelberg.de

• *Heiko Rieger,* Ch. 1

Theoretische Physik
Universität des Saarlandes
D-66041 Saarbrücken, Germany
e-mail: rieger@lusi.uni-sb.de

• *Giovanni Rinaldi,* Ch. 4

Istituto di Analisi dei Sistemi ed Informatica
'Antonio Ruberti' - CNR
Viale Manzoni, 30
00185 Roma, Italy
e-mail: rinaldi@iasi.rm.cnr.it

• *Martin Weigt,* Ch. 7

Institut für Theoretische Physik
Universität Göttingen
Tammanstraße 1
D-37077 Göttingen, Germany
e-mail: weigt@theorie.physik.uni-goettingen.de

• *Riccardo Zecchina,* Ch. 9

International Centre for Theoretical Physics (ICTP)
Strada Costiera, 11 P.O.Box 586
I-34100 Trieste, Italy
e-mail: zecchina@ictp.trieste.it

# 1 Introduction

*Alexander K. Hartmann and Heiko Rieger*

Optimization problems occur very frequently in physics. Some of them are easy to handle with conventional methods also used in other areas such as economy or operations research. But as soon as a huge number of degrees of freedom are involved, as is typically the case in statistical physics, condensed matter, astrophysics and biophysics, conventional methods fail to find the optimum in a reasonable time and new methods have to be invented. This book contains a representative collection of new optimization algorithms that have been devised by physicists from various fields, sometimes based on methods developed by computer scientists and mathematicians. However, it is not a mere collection of algorithms but tries to demonstrates their scope and efficiency by describing typical situations in physics where they are useful.

The individual articles of this collections are self-contained and should be understandable for scientists routinely using numerical tools. A more basic and pedagogical introduction into optimization algorithms is our book on *Optimization Algorithms in Physics*, which can serve as an appendix for the newcomer to this field of computational physics or for undergraduate students. The reason why we found it necessary to compose another book in this field with a greater focus is the fact that the application of optimization methods is one of the strongest growing fields in physics. The main reasons for these current developments are the following key factors:

First of all great progress has been made in the development of new combinatorial optimization methods in computer science. Using these sophisticated approaches, much larger system sizes of the corresponding physical systems can be treated. For many models the systems sizes which were accessible before, were too small to obtain reliable and significant data. However, this is now possible. In this way computer science has helped physics.

But knowledge transfer also works the other way round. Physics provides still new insights and methods of treating optimization problems, such as the earlier invention of the simulated annealing technique. Recent algorithmic developments in physics are, e.g., the extremal optimization method or the hysteric optimization approach, both covered in this book.

Moreover, phase transitions were recently found in "classical" optimization problems within theoretical computer science, during the study of suitably parameterized ensembles. These phase transitions very often coincide with peaks of the running time or with changes of the typical-case complexity from polynomial to exponential. As well as the gain from taking the physical viewpoint, by mapping the optimization problems to physical systems and applying methods from statistical physics, it is possible to obtain many results, which have not been found with traditional mathematical techniques. This is true also for the analysis of the typical-case complexity of (random) algorithms.

Finally: All benefit from the increasing power of computers. Despite all predictions, the speed of the hardware still seems to grow exponentially fast, making the application of optimizations methods more and more valuable.

Thus the aim of this book is to promote progress in the fields given above. Physicists will become familiar with the huge progress still taking place in the development of algorithmic techniques. On the other hand, the new developments of physically inspired algorithms can be very useful in computer science as well. In particular the application of physical methods in the field of phase transitions seems to be a very promising field for the next decade.

Currently, the interactions between different communities, namely mathematics, computer science, biology, economy and physics are still too weak. Only by gathering researchers from these different groups and trying to find a common language, can real progress be achieved. All problems, algorithms and results are presented here in a pedagogical way, which makes the information available to a broad audience. This is the main purpose of this collection of papers.

The book contains three main parts. In the first part, we focus on applications of optimization algorithms to problems from physics. The standard way of solving computational problems in statistical physics is to use a Monte Carlo simulation. In his contribution, Werner Krauth shows that by using modern cluster algorithms, many previously inaccessible models can be treated at low temperatures (obtaining low, i.e., minimum energies) or respectively, high densities. He studies as examples the phase separation in binary mixtures and the application of the algorithm to monomer-dimer models. Next, Olivier Martin surveys algorithms for Ising spin-glass ground-state calculations and he explains one new Monte Carlo algorithm in detail. It is a cluster method based on the real-space renormalization group.

Monte Carlo methods, like those shown in the first two contributions, are very efficient and have a wide range of applicability, but they do not guarantee to find a global optimum solution. In contrast, the Branch-and-Cut approach is an exact algorithm. It is presented by Frauke Liers, Michael Jünger, Gerhard Reinelt and Giovanni Rinaldi. They explain the method for an application to the max-cut problem, which is used here for the ground-state calculation of three-dimensional Ising spin glasses.

Another important class of problems in statistical physics is the random-field Ising model. Alan Middleton explains how one can calculate ground states using push/relabel algorithms in polynomial time, how these algorithms perform near phase transitions and how one can use it to characterize the ground-state landscape of the random-field model. In the last chapter of the first part, Jean-Christian Anglès d'Auriac describes a new method for calculating the partition function and other thermodynamic quantities of the infinite-state Potts model with random bonds using a combinatorial optimization algorithm. The latter is based on the concept of submodular functions, which might also prove useful in a number of other applications in the near future.

The second part is dedicated to the study of phase transitions in combinatorial optimization problems. First, Martin Weigt introduces the Satisfiability Problem (SAT), the most fundamental problem in computational complexity theory. He then shows how one can generate large SAT formulas which have a solution but where the solution is hard to find for local algorithms like Walksat. This behavior can be understood by solving the corresponding physical problem analytically by using techniques from statistical mechanics. Simona Cocco, Liat Ein-Dor and Remi Monasson show how one can calculate the typical running time of exact

backtracking algorithms for SAT and for the coloring problem. The basic idea is to investigate the dynamics of the algorithm moving in the phase diagram of the problem. Finally, Riccardo Zecchina presents the currently fastest Algorithm for SAT, the Survey Propagation algorithm, which allows to solve SAT instances near the SAT-UNSAT phase transition of systems having $10^6$ variables. The method is based on the cavity approach, an analytical technique used to study mean-field-like disordered systems in statistical physics. Nevertheless, his presentation is solely based on probability theory, making it also very accessible to non-physicists.

The third part of this book is on new heuristics and interdisciplinary applications. Károly Pál presents an optimization method which is inspired by a physical technique, the measurement of hysteresis in a magnetic system. The basic idea is to demagnetize a system by performing hysteresis loops with continuously decreasing magnitude. He presents the algorithm in a very general style, which in principle allows arbitrary applications. As examples, results for spin glasses and the traveling salesman problem are shown. Stefan Boettcher explains another very general algorithm, the extremal optimization algorithm. Its basic idea is very simple and similar to genetic algorithms. The latter ones usually have many free parameters, which must be tuned to obtain an efficient algorithm. Extremal optimization has the advantage that it is, in the simplest variant, absolutely parameter free. Another major difference in comparison with genetic algorithms is that fitness values are not assigned to different configurations but to different particles of one configuration. Application to graph coloring, spin glasses and image matching are given.

The last two contributions contain applications from Molecular Biology. After providing some biological background, Alexander Hartmann explains alignment algorithms, which are used to compare biological sequences by applying a shortest-path algorithm. As an application, a method to obtain the rare-event tail of the statistics of protein alignments is presented. Finally, Ulrich Hansmann reviews methods used to solve protein-folding problems via energy minimization and in particular explains energy-landscape paving. The basic idea is that one initially modifies the energy landscape such that the global minimum is easier to find. During the simulation, the energy landscape gradually approaches the ordinal one. Furthermore, the algorithm tries to avoid previously visited regions, if the energy is not low enough. Various results for the influence of the temperature on helix formation are also shown.

Compiling this book would not have been possible without the help of many people and various institutions. First of all, we would like to thank all authors for preparing their valuable contributions and also for their helpful cooperation. Furthermore, we are particularly indebted to Vera Palmer, Uwe Krieg, and Cornelia Wanka from Wiley-VCH for the excellent collaboration. Financial support was provided by the VolkswagenStiftung within the Program "Nachwuchsgruppen an Universitäten", by the Deutsche Forschungsgemeinschaft (DFG), by the International Conference and Research Center for Computer Science Schloss Dagstuhl and by the Institute for Scientific Interchange (ISI) Foundation in Turin. The European Community supported this book financially via the Human Potential Program under contract number HPRN-CT-2002-00307 (DYGLAGEMEM), via the High-Level Scientific Conferences (HLSC) program, and via the Complex Systems Network of Excellence "Exystence". This book was prepared in connection with the Dagstuhl Seminar No. 03381 "New Optimization Algorithms in Physics".

# Part I: Applications in Physics

# 2 Cluster Monte Carlo Algorithms

*Werner Krauth*

In recent years, a better understanding of the Monte Carlo method has provided us with many new techniques in different areas of statistical physics. Of particular interest are the so called cluster methods, which exploit the considerable algorithmic freedom given by the detailed balance condition. Cluster algorithms appear, among other systems, in classical spin models, such as the Ising model [14], in lattice quantum models (bosons, quantum spins and related systems) [5] and in hard spheres and other "entropic" systems for which the configurational energy is either zero or infinite [4].

In this chapter, we discuss the basic idea of cluster algorithms with special emphasis on the pivot cluster method for hard spheres and related systems, for which several recent applications are presented. We provide less technical detail but more context than in the original papers. The best implementations of the pivot cluster algorithm, the "pocket" algorithm [10], can be programmed in a few lines. We start with a short exposition of the detailed balance condition, and of "*a priori*" probabilities, which are needed to understand how optimized Monte Carlo algorithms may be developed. A more detailed discussion of these subjects will appear in a forthcoming book [9].

## 2.1 Detailed Balance and a priori Probabilities

In contrast with the combinatorial optimization methods discussed elsewhere in this book, the Monte Carlo approach does not construct a well-defined state of the system – minimizing the energy, or maximizing flow, etc – but attempts to generate a number of statistically independent representative configurations $a$, with probability $\pi(a)$. In classical equilibrium statistical physics, $\pi(a)$ is given by the Boltzmann distribution, whereas, in quantum statistics, the weight is the diagonal many-body density matrix.

In order to generate these configurations with the appropriate weight (and optimal speed), the Monte Carlo algorithm moves (in one iteration) from configuration $a$ to configuration $b$ with probability $P(a \to b)$. This transition probability is chosen to satisfy the fundamental condition of detailed balance

$$\pi(a)P(a \to b) = \pi(b)P(b \to a) \tag{2.1}$$

which is implemented using the Metropolis algorithm

$$P(a \to b) = \min\left(1, \frac{\pi(b)}{\pi(a)}\right) \tag{2.2}$$

or one of its variants.

For the prototypical Ising model, the stationary probability distribution (the statistical weight) of a configuration is the Boltzmann distribution with an energy given by

$$E = -J \sum_{\langle i,j \rangle} S_i S_j \quad L > 0 \tag{2.3}$$

as used and modified in many other places in this book. A common move consists of a spin flip on a particular site $i$, transforming configuration $a$ into another configuration $b$. This is shown in Figure 2.1 (left). In a hard sphere gas, also shown in Figure 2.1 (right), one typically displaces a single particle $i$ from $\mathbf{x}$ to $\mathbf{x} + \boldsymbol{\delta}$. There is a slight difference between these two simple algorithms: by flipping the same spin twice, one goes back to the initial configuration: a spin flip is its own inverse. In contrast, in the case of the hard-sphere system, displacing a particle twice by the same vector $\boldsymbol{\delta}$ does not usually bring one back to the original configuration.


**Figure 2.1:** Two examples of local Monte Carlo algorithms: the two-dimensional Ising model with single-spin flip dynamics (left) and two-dimensional hard disks with a single-particle move (right).

An essential concept is the one of an *a priori* probability: it accounts for the fact that the probability $P(a \to b)$ is a composite object, constructed from the probability of *considering* the move from $a$ to $b$, and the probability of *accepting* it.

$$P(a \to b) = \underbrace{\mathcal{A}(a \to b)}_{\text{consider } a \to b} \times \underbrace{\tilde{P}(a \to b)}_{\text{accept } a \to b}$$

In usual Monte Carlo terminology, if $a \to b$ is rejected (after having been considered), then the "move" $a \to a$ is chosen instead and the system remains where it is.

With these definitions, the detailed balance condition Eq. (2.1) can be written as

$$\frac{\tilde{P}(a \to b)}{\tilde{P}(b \to a)} = \frac{\pi(b)}{\mathcal{A}(a \to b)} \frac{\mathcal{A}(b \to a)}{\pi(a)}$$

and implemented by a Metropolis algorithm generalized from Eq. (2.2):

$$\tilde{P}(a \to b) = \min \left\{ 1, \frac{\pi(b)}{\mathcal{A}(a \to b)} \frac{\mathcal{A}(b \to a)}{\pi(a)} \right\} \tag{2.4}$$

It is very important to realize that the expression "*a priori* probability $\mathcal{A}(a \to b)$" is synonymous to "Monte Carlo algorithm". A Monte Carlo algorithm $\mathcal{A}(a \to b)$ of our own conception must satisfy three conditions:

1. It must lead the state of the system from a configuration $a$ to a configuration $b$, in such a way that, eventually, all configurations in phase space can be reached (ergodicity).

2. It must allow to compute the ratio $\pi(a)/\pi(b)$. This is trivially satisfied, at least for classical systems, as the statistical weight is simply a function of the energy.

3. It must allow, for any possible transition $a \to b$, to compute both the probabilities $\mathcal{A}(a \to b)$ and $\mathcal{A}(b \to a)$. Again, it is the ratio of probabilities which is important.

A trivial application of *a priori* probabilities for hard spheres is given in Figure 2.2. (Suppose that the points $a$ and $b$ are embedded in a large two-dimensional plane.) On the left side of the figure, we see one of the standard choices for the trial moves $\mathbf{x} \to \mathbf{x} + \boldsymbol{\delta}$ of a particle in Figure 2.1: The vector $\boldsymbol{\delta}$ is uniformly sampled from a square centered around the current position. If, however, we decide for some obscure reason to sample $\boldsymbol{\delta}$ from a triangle, we realize that in cases such as the one shown in Figure 2.2 (right), the *a priori* probability for the return move vanishes. It is easy to see from Eq. (2.4) that, in this case, both $P(a \to b)$ and $P(b \to a)$ are zero.

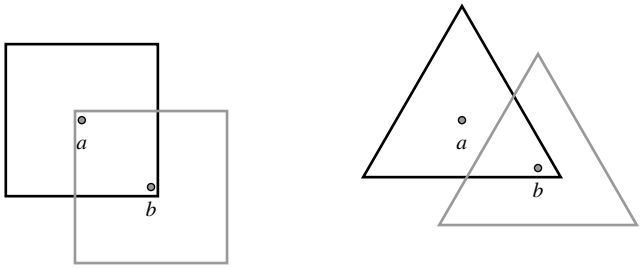


**Figure 2.2:** A priori probabilities for the hard-sphere system. Left: "square" – $\mathcal{A}(a \to b)$ is constant within the square boundary, and zero outside. By construction, $\mathcal{A}(a \to b) = \mathcal{A}(b \to a)$. Right: "triangle" – for the analogous (if hypothetical) case of a triangle, there are pairs $a, b$, where $\mathcal{A}(a \to b)$ is finite, but $\mathcal{A}(b \to a) = 0$. Both rates $P(a \to b)$ and $P(b \to a)$ vanish.

Notwithstanding its simplicity, the triangle "algorithm" illustrates that *any* Monte Carlo method $\mathcal{A}(a \to b)$ can be made to comply with detailed balance, if we feed it through Eq. (2.4). The usefulness of the algorithm is uniquely determined by the speed with which it moves through configuration space, and is highest if no rejections at all appear. It is to be noted, however, that, even if $\tilde{P}(a \to b)$ is always 1 (no rejections), the simulation *can* remain rather difficult. This happens, for example, in the two-dimensional $XY$-model and in several examples treated below.

Local algorithms are satisfactory for many problems but fail whenever the typical differences between relevant configurations are much larger than the change that can be achieved by one iteration of the Monte Carlo algorithm. In the Ising model at the critical point, for example, the distribution of magnetizations is wide, but the local Monte Carlo algorithm implements a change of magnetization of only $\pm 2$. This mismatch lies at the core of critical slowing down in experimental systems and on the computer.

In liquids, modeled e.g. by the hard-sphere system, another well-known limiting factor is that density fluctuations can relax only through local diffusion. This process generates slow hydrodynamic modes, if the overall diffusion constants are small.

Besides these slow *dense* systems, there is also the class of highly *constrained* models, of which binary mixtures will be treated later. In these systems, the motion of some degrees of freedom naturally couple to many others. In a binary mixture, e. g., a big colloidal particle is surrounded by a large number of small particles, which are influenced by its motion. This is extremely difficult to deal with in Monte Carlo simulations, where the local moves $\mathbf{x} \to \mathbf{x} + \boldsymbol{\delta}$ are essentially the unconstrained motion of an isolated particle.

## 2.2 The Wolff Cluster Algorithm for the Ising Model

The local spin-flip Monte Carlo algorithm not being satisfactory, it would be much better to move large parts of the system, so called clusters. This cannot be done by a blind flip of one or many spins (with $\mathcal{A}(a \to b) = \mathcal{A}(b \to a)$), which allows unit acceptance rate both for the move $a \to b$ and its reverse $b \to a$ only if the energies of both configurations are the same. One needs an algorithm whose *a priori* probabilities $\mathcal{A}(a \to b)$ and $\mathcal{A}(b \to a)$ soak up any differences in statistical weight $\pi(a)$ and $\pi(b)$.

This can be done by starting the construction of a cluster with a randomly sampled spin and by iteratively adding neighboring sites of the same magnetization with a probability $p$. To be precise, one should speak about "links": if site $i$ is in the cluster and a neighboring site $j$ is not, and if, furthermore, $S_i = S_j$, then one should add link $\langle i, j \rangle$ with probability $p$. A site is added to the cluster if it is connected by at least one link. In configuration $a$ of Figure 2.3, the cluster construction has stopped in the presence of 9 links "$--$" across the boundary. Each of these links could have been accepted with probability $p$, but has been rejected. This gives a term $(1-p)^9$ in the *a priori* probability. Flipping the cluster brings us to configuration $b$. The construction of the cluster for configuration $b$ would stop in the presence of 19 links "$++$" across the boundary (*a priori* probability $\propto (1-p)^{19}$)).

This allows us to compute the *a priori* probabilities

$$\begin{aligned}
\mathcal{A}(a \to b) &= \mathcal{A}_{\text{interior}} \times (1-p)^9 \\
\mathcal{A}(b \to a) &= \mathcal{A}_{\text{interior}} \times (1-p)^{19} \\
E_a &= E_{\text{interior}} + E_{\text{exterior}} - 9 \times J + 19 \times J \qquad (\pi_a = \exp[-\beta E_a]) \\
E_b &= E_{\text{interior}} + E_{\text{exterior}} - 19 \times J + 9 \times J \qquad (\pi_b = \exp[-\beta E_b])
\end{aligned}$$

In these equations, the "interior" refers to the part of the cluster which does not touch the boundary. By construction, the "interior" and "exterior" energies and *a priori* probabilities are the same for any pair of configurations $a$ and $b$ which are connected through a single cluster flip.

We thus dispose of all the information needed to evaluate the acceptance probability $\tilde{P}$ in Eq. (2.4), which we write more generally in terms of the number of "same" and of "different" links in the configuration $a$. These notions are interchanged for configuration $b$ (in Figure 2.3,

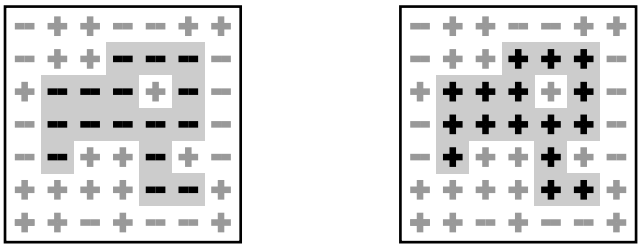


**Figure 2.3:** The Wolff cluster algorithm for the Ising model adds, with probability $p$, a link connecting a site outside the cluster to a site already in the cluster (thereby adding the site). In the configuration $a$, construction of the cluster (as shown) stopped with 9 links "$--$", corresponding to an *a priori* probability $\mathcal{A}(a \to b) = \mathcal{A}_{\text{interior}} \times (1-p)^9$. The return move stops with probability $\mathcal{A}(b \to a) = \mathcal{A}_{\text{interior}} \times (1-p)^{19}$, as there are 19 links "$++$" across the boundary in configuration $b$.

we have $n_{\text{same}} = 9$, $n_{\text{diff}} = 19$). With the energy scale $J$ set to 1, we find

$$\begin{aligned}\tilde{P}(a \to b) &= \min\left\{1, \frac{e^{\beta n_{\text{diff}}} e^{-\beta n_{\text{same}}}}{(1-p)^{n_{\text{same}}}} \frac{(1-p)^{n_{\text{diff}}}}{e^{-\beta n_{\text{diff}}} e^{\beta n_{\text{same}}}}\right\} \\ &= \min\left\{1, \left[\frac{e^{-2\beta}}{1-p}\right]^{n_{\text{same}}} \left[\frac{1-p}{e^{-2\beta}}\right]^{n_{\text{diff}}}\right\} \end{aligned} \tag{2.5}$$

Once the cluster construction stops, we know the configuration $b$, may count $n_{\text{same}}$ and $n_{\text{diff}}$, and evaluate $\tilde{P}(a \to b)$. Of course, a lucky coincidence[1] occurs for $p = 1 - \exp[-2J\beta]$. This special choice yields a rejection-free algorithm whose acceptance probability is unity for all possible moves and is implemented in the celebrated Wolff cluster algorithm [14], the fastest currently known simulation method for the Ising model. The Wolff algorithm can be programmed in a few lines, by keeping a vector of cluster spins, and an active frontier, as shown below. The algorithm below presents a single iteration $a \to b$. The function ran$[0,1]$ denotes a uniformly distributed random number between 0 and 1, and $p$ is set to the magic value $p = 1 - \exp[-2J\beta]$. The implementation uses the fact that a cluster can grow only at its frontier (called the "old" frontier $\mathcal{F}_{\text{old}}$, and generating the new one $\mathcal{F}_{\text{new}}$). It goes without saying that for the magic value of $p$ we do not have to evaluate $\tilde{P}(a \to b)$ in Eq. (2.5), as it is always 1. Any proposed move is accepted.

[1] This accident explains the deep connection between the Ising model and percolation.

```
algorithm wolff-cluster
begin
   i := random particle;
   C := {i};
   F_old := {i};
   while F_old ≠ {} do
   begin
      F_new := {};
      for ∀ i ∈ F_old do
      begin
         for ∀ j neighbor of i with S_i = S_j, j ∉ C do
         begin
            if ran[0,1] < p then
            begin
               F_new := F_new ∪ {j};
               C := C ∪ {j};
            end
         end
      end
      F_old := F_new;
   end
   for ∀ i ∈ C do
   S_i := −S_i;
end
```

## 2.3 Cluster Algorithm for Hard Spheres and Related Systems

We want to further exploit the analogy between the spin model and the hard-sphere system. As the spin-cluster algorithm constructs a cluster of spins which *flip* together, one might think that a cluster algorithm for hard spheres should identify "blobs" of spheres that *move* together. Such a macroscopic ballistic motion would replace slow diffusion.

To see that this strategy cannot be successful, it suffices to look at the generalized detailed balance condition in the example shown in Figure 2.4: any reasonable algorithm $\mathcal{A}$ would have less trouble spotting the cluster of dark disks in configuration $a$ than in $b$. This means that $\mathcal{A}(a \to b) \gg \mathcal{A}(b \to a)$ and that the acceptance rate $\tilde{P}(a \to b)$ would be very small.

The imbalance between $\mathcal{A}(a \to b)$ and $\mathcal{A}(b \to a)$ can, however, be avoided if the two transition probabilities are protected by a symmetry principle: the transformation $T$ producing $b$ from $a$ must be the same as the one producing $a$ from $b$. Thus, $T$ should be its own inverse.

In Figure 2.5, this program is applied to a hard disk configuration using, as transformation $T$, a rotation by an angle $\pi$ around an arbitrarily sampled pivot (denoted by $\oplus$, for each iteration a new pivot is used). Notice that for a symmetric particle, the rotation by an angle $\pi$ is identical to the reflection around the pivot. It is useful to transform not just a single particle,

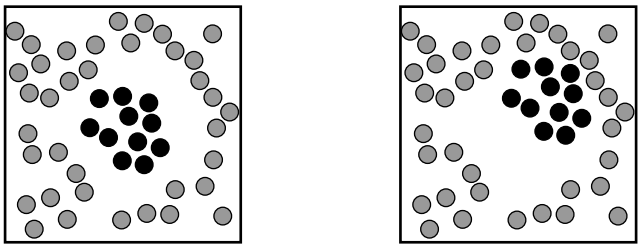


**Figure 2.4:** The dark disks are easier to identify as a cluster in configuration $a$ than in $b$, where they are fused into the background. This means that, for the configurations $a$ and $b$ shown in this figure, $\mathcal{A}(a \to b) \gg \mathcal{A}(b \to a)$ for any generic Monte Carlo algorithm. As $\pi(a) = \pi(b)$, the acceptance probability $\tilde{\mathcal{P}}(a \to b)$ in Eq. (2.4) will be extremely small. The problem can be avoided [4] if the transformation $a \to b$ is protected by a symmetry principle: it must be its own inverse.

but the whole original configuration $a$ yielding the "copy". By overlaying the original with its rotated copy, we may identify the invariant sub-ensembles (clusters) which transform independently under $T$. For example, in Figure 2.5, we may rotate the disks numbered $6$, $8$, and $9$, which form a cluster of overlapping disks in the ensemble of overlayed original and copy.

In Figure 2.5, there are the following three invariant clusters:

$$\{6, 8, 9\}, \{2, 3, 4, 7\}, \{1, 5\} \tag{2.6}$$

The configuration $b$ in Figure 2.5 shows the final positions after rotation of the first of these clusters. By construction, $\mathcal{A}(a \to b) = \mathcal{A}(b \to a)$ and $\pi(a) = \pi(b)$. This perfect symmetry ensures that detailed balance is satisfied for the non-local move. Notice that moving the cluster $\{1, 5\}$ is equivalent to exchanging the labels of the two particles and performing two local moves. Ergodicity of the algorithm follows from ergodicity of the local algorithm, as a local move $\mathbf{x} \to \mathbf{x} + \boldsymbol{\delta}$ can always be disguised as a cluster rotation around the pivot $\mathbf{x} + \boldsymbol{\delta}/2$.

Figure 2.5 indicates the basic limitation of the pivot cluster approach: if the density of particles becomes too large, almost all particles will be in the same cluster, and flipping it will essentially rotate the whole system. Nevertheless, even though above the percolation threshold in the thermodynamic limit there exists a large cluster containing a finite fraction of all particles, the remaining particles are distributed among a distribution of small clusters. This means that finite clusters of various sizes will be produced. These may give rise to useful moves, for example in the case of dense polydisperse disks discussed below. Even small clusters provide non-diffusive mass transport if they contain an odd number of particles (cf. the example in Figure 2.5) or particles of different type.

It is also useful to discuss what will happen if the "copy" does not stem from a symmetry operation, for example, if the copy is obtained from the original through a simple translation with a vector $\boldsymbol{\delta}$. In this case, there would still be clusters, but they no longer appear in pairs. It would still be possible to flip individual clusters, but not to conserve the number of particles

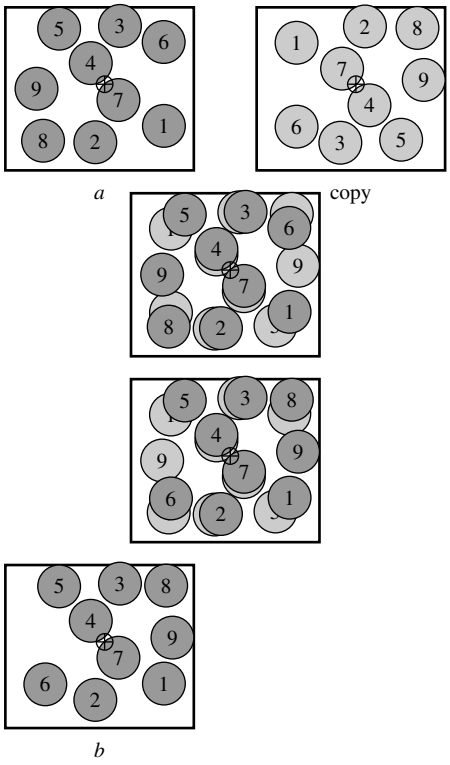


**Figure 2.5:** The pivot cluster algorithm performs a symmetry operation which is its own inverse. In this system of hard disks (with periodic boundary conditions), a rotation by an angle $\pi$ around an arbitrarily sampled pivot ($\oplus$) is shown: $a$ is the original configuration, $b$ the rotated copy. The intermediate pictures show the superposed system of original and copy before and after the flip. The final configuration, $b$, is also shown. Notice that the transformation maps the simulation box (with periodic boundary conditions) onto itself. If this is not the case, the treatment of boundary conditions becomes more involved, and generates rejections.

on each plate. This setting can also have important applications, it is very closely related to Gibbs ensemble simulations and provides an optimal way of exchanging particles between two plates. The two plates would no longer describe the same system but would be part of a larger system of coupled plates.

```
algorithm pocket-cluster
begin
   r_pivot := random point in box;
   i := random particle;
   P := {i};
   O := {all particles} \ {i};
   while P ≠ {} do
   begin
      i := any element of P;
      P := P \ {i};
      r(i) := reflection of r(i) around r_pivot;
      for ∀ j ∈ O do
      if j ∩ i then
      begin
         O := O \ {j};
         P := P ∪ {j};
      end
   end
end
```

Having discussed the conceptual underpinnings of the pivot cluster algorithm, it is interesting to understand how it can be made into a working program. Figure 2.5 suggests that one should use a representation with two plates, and perform cluster analyses, very similar to what is done in the Wolff algorithm.

However, it is not necessary to work with two plates. The transformation can be done on the system itself and does not even have to consider a cluster at all. This ultimately simple solution is achieved in the "pocket" algorithm [10]: it merely keeps track of particles which eventually have to be moved in order to satisfy all the hard-core constraints. After sampling the pivot (or another symmetry operation), one chooses a first particle, which is put into the pocket. At each stage of the iteration, one particle is taken from the pocket, and the transformation is applied to it. At the particle's new position, the hard-core constraint will probably be violated for other particles. These have simply to be marked as "belonging to the pocket". One single "move" of the cluster algorithm consists of all the stages until the pocket is empty or, equivalently, of all the steps leading from frame $a$ to frame $e$ in Figure 2.6. The inherent symmetry guarantees that the process will end with an empty pocket, and detailed balance will again be satisfied as the output is the same as in the two-plate version.

In the printed algorithm, $\mathcal{P}$ stands for the "pocket", and $\mathcal{O}$ is the set of "other" particles that currently do not have to be moved to satisfy the hard-core constraints. The expression $j \cap i$ is "true" if the pair $i, j$ violates the hard-core constraint.

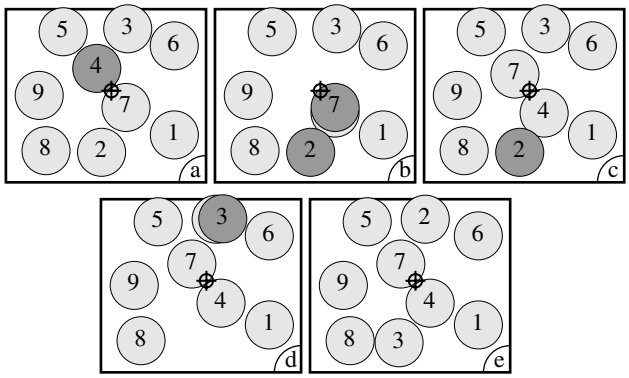


**Figure 2.6:** One iteration of the pocket algorithm ("pocket" ≡ "dark disks"). Initially (frame $a$), a pivot is chosen and a starting disk (here disk 4) is put into the pocket. At each subsequent step, a disk is removed from the pocket and transformed with respect to the pivot. Any overlapping disks are added to the pocket. For example, in frame $b$, overlaps exist between disk 4 (which has just been moved) and disks 2 and 7. Only one of these disks is transformed in frame $c$. The pocket algorithm is guaranteed to move from a valid hard-disk configuration to another one, and to respect detailed balance. It can be implemented in a few lines of code, as shown in the algorithm on page 15.

## 2.4 Applications

### 2.4.1 Phase Separation in Binary Mixtures

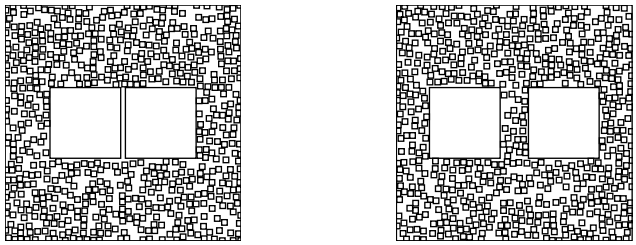

**Figure 2.7:** Entropic interaction between two colloids (squares of edge length $d_{\text{large}}$) in a sea of small particles (of size $d_{\text{small}}$). Left: Small particles cannot penetrate into the slit between the large particles. The concentration difference leads to an effective entropic interaction between colloids, which is attractive at small separation. Right: At large distances between colloids, the effective interaction vanishes.

The depletion force – one of the basic interactions between colloidal particles – is of purely entropic origin. It is easily understood for a system of large and small squares (or cubes): In the left picture of Figure 2.7, the two large squares are very close together so that no small