THE AMERICAN MATHEMATICAL MONTHLY

Volume 117, Number 5 May 2010

THE AMERICAN MATHEMATICAL
# MONTHLY

Volume 117, Number 5 May 2010

# NOTICE TO AUTHORS

The MONTHLY publishes articles, as well as notes and other features, about mathematics and the profession. Its readers span a broad spectrum of mathematical interests, and include professional mathematicians as well as students of mathematics at all collegiate levels. Authors are invited to submit articles and notes that bring interesting mathematical ideas to a wide audience of MONTHLY readers.

The MONTHLY's readers expect a high standard of exposition; they expect articles to inform, stimulate, challenge, enlighten, and even entertain. MONTHLY articles are meant to be read, enjoyed, and discussed, rather than just archived. Articles may be expositions of old or new results, historical or biographical essays, speculations or definitive treatments, broad developments, or explorations of a single application. Novelty and generality are far less important than clarity of exposition and broad appeal. Appropriate figures, diagrams, and photographs are encouraged.

Notes are short, sharply focused, and possibly informal. They are often gems that provide a new proof of an old theorem, a novel presentation of a familiar theme, or a lively discussion of a single issue.

Articles and notes should be sent to the Editor:

DANIEL J. VELLEMAN
American Mathematical Monthly
Amherst College
P. O. Box 5000
Amherst, MA 01002-5000
*mathmonthly@amherst.edu*

For an initial submission, please send a pdf file as an email attachment to: mathmonthly@amherst.edu. (Pdf is the only electronic file format we accept.) Please put "Submission to the Monthly" in the subject line, and include the title of the paper and the name and postal address of the corresponding author in the body of your email. If submitting more than one paper, send each in a separate email. In lieu of a pdf, an author may submit a single paper copy of the manuscript, printed on only one side of the paper. Manuscript pages should be numbered, and left and right margins should be at least one inch wide. Authors who use LaTeX are urged to use article.sty, or a similar generic style, and its standard environments with no custom formatting. See recent articles in the MONTHLY for the style of citations for journal articles and books. Follow the link to *Electronic Publication Information* for authors at http://www.maa.org/pubs/monthly.html for information about figures and files as well as general editorial guidelines.

Letters to the Editor on any topic are invited. Comments, criticisms, and suggestions for making the MONTHLY more lively, entertaining, and informative are welcome.

The online MONTHLY archive at *www.jstor.org* is a valuable resource for both authors and readers; it may be searched online in a variety of ways for any specified keyword(s). MAA members whose institutions do not provide JSTOR access may obtain individual access for a modest annual fee; call 800-331-1622.

See the MONTHLY section of MAA Online for current information such as contents of issues and descriptive summaries of forthcoming articles:

http://www.maa.org/

Proposed problems or solutions should be sent to:

DOUG HENSLEY, MONTHLY Problems
Department of Mathematics
Texas A&M University
3368 TAMU
College Station, TX 77843-3368

In lieu of duplicate hardcopy, authors may submit pdfs to monthlyproblems@math.tamu.edu.

Advertising Correspondence:
MAA Advertising
1529 Eighteenth St. NW
Washington DC 20036

Phone: (866) 821-1221
Fax: (866) 387-1208
E-mail: advertising@maa.org

Further advertising information can be found online at www.maa.org

Change of address, missing issue inquiries, and other subscription correspondence:
MAA Service Center, *maahq@maa.org*

All at the address:

The Mathematical Association of America
1529 Eighteenth Street, N.W.
Washington, DC 20036

Recent copies of the MONTHLY are available for purchase through the MAA Service Center.
maahq@maa.org, 1-800-331-1622

Microfilm Editions: University Microfilms International, Serial Bid coordinator, 300 North Zeeb Road, Ann Arbor, MI 48106.

The AMERICAN MATHEMATICAL MONTHLY (ISSN 0002-9890) is published monthly except bimonthly June-July and August-September by the Mathematical Association of America at 1529 Eighteenth Street, N.W., Washington, DC 20036 and Hanover, PA, and copyrighted by the Mathematical Association of America (Incorporated), 2010, including rights to this journal issue as a whole and, except where otherwise noted, rights to each individual contribution. Permission to make copies of individual articles, in paper or electronic form, including posting on personal and class web pages, for educational and scientific use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear the following copyright notice: [Copyright the Mathematical Association of America 2010. All rights reserved.] Abstracting, with credit, is permitted. To copy otherwise, or to republish, requires specific permission of the MAA's Director of Publications and possibly a fee. Periodicals postage paid at Washington, DC, and additional mailing offices. ***Postmaster***: Send address changes to the American Mathematical Monthly, Membership/Subscription Department, MAA, 1529 Eighteenth Street, N.W., Washington, DC, 20036-1385.

# The Evolution of Markov Chain Monte Carlo Methods

**Matthew Richey**

**1. INTRODUCTION.** There is an algorithm which is powerful, easy to implement, and so versatile it warrants the label "universal." It is flexible enough to solve otherwise intractable problems in physics, applied mathematics, computer science, and statistics. It works in both probabilistic and deterministic situations. Best of all, because it was inspired by Nature, it is blessed with extreme elegance.

This algorithm is actually a collection of related algorithms—Metropolis-Hastings, simulated annealing, and Gibbs sampling—together known as *Markov chain Monte Carlo* (MCMC) methods. The original MCMC method, the Metropolis algorithm, arose in physics, and now its most current variants are central to computational statistics. Along the way from physics to statistics the algorithm appeared in—and was transformed by—applied mathematics and computer science. Perhaps no other algorithm has been used in such a range of areas. Even before its wondrous utility had been revealed, its discovers knew they had found

> . . . a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. [**48**]

This is the story of the evolution of MCMC methods. It begins with a single paper, one with no antecedent. The original idea required the right combination of place, people, and perspective. The place was Los Alamos right after World War II. The people included the familiar—von Neumann, Ulam, Teller—along with several less familiar. The perspective was that randomness and sampling could be used to circumvent insurmountable analytic roadblocks. There was also one last necessary ingredient present: a computer.

The evolution of MCMC methods is marked by creative insights by individuals from seemingly disparate disciplines. At each important juncture, a definitive paper signaled an expansion of the algorithm into new territory. Our story will follow the chronological order of these papers.

1. *Equations of State Calculations by Fast Computing Machines*, 1953, by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller [**48**], which introduced the Metropolis algorithm.
2. *Optimization by Simulated Annealing*, 1983, by Kirkpatrick, Gelatt, and Vecchi [**45**], which brought simulated annealing to applied mathematics.
3. *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*, 1984, by Geman and Geman [**28**], which introduced Gibbs sampling.

doi:10.4169/000298910X485923

4. *Sampling-Based Approaches to Calculating Marginal Densities*, 1990, by Gelfand and Smith [**25**], which brought the power of MCMC methods to the statistics community.

It is difficult to overstate the impact of these papers and the importance of MCMC methods in modern applied mathematics. Collectively these four papers have been cited almost 40,000 times. The original Metropolis algorithm has been called one of the ten most important algorithms of the twentieth century [**19**].[1]

The goal here is not to provide a tutorial on how to use MCMC methods; there are many resources for this purpose [**8, 54, 27, 29, 10, 65, 61**]. Rather, the goal is to tell the story of how MCMC methods evolved from physics into applied mathematics and statistics. Parts of this story have already been told. Much has been written about the research at Los Alamos that led to the Metropolis algorithm; see, for example, the Los Alamos publications [**2, 46, 47, 32**]. A very nice overview of the connections between the Metropolis algorithm and modern statistics can be found in [**36, 10, 63**]. An unpublished work by Robert and Casella [**55**] also focuses on the history of MCMC methods, mostly from a statistician's perspective. Less has been said about the history of simulated annealing and the role of MCMC methods in image restoration.

**2. THE BEGINNING: METROPOLIS ET AL., 1953.** Our story begins with the Metropolis algorithm, the original Markov chain Monte Carlo method. But first we take a brief look at the history of Monte Carlo methods and also recall some facts about Markov chains.

**2.1. Monte Carlo Methods.** Shortly after World War II, Los Alamos was a hotbed of applied mathematics and theoretical physics. Much of the work was motivated by the intense focus on developing nuclear weapons. One particularly difficult problem was to estimate the behavior of large (e.g., $10^{23}$) collections of atomic particles. The physical laws governing their behavior—thermodynamics, statistical physics, and quantum mechanics—were inherently probabilistic and so complicated that traditional methods were not sufficient for the sort of detailed analysis needed. In this setting a new idea took hold; instead of searching for closed form, analytic solutions, one could *simulate* the behavior of the system in order to estimate the desired solution. Producing simulations was a challenge. Before the late 1940s no device existed that could quickly and accurately carry out large-scale random simulations. By the end of World War II, things were different. Researchers at Los Alamos had access to such a device, the ENIAC (Electronic Numerical Integrator and Computer) at the University of Pennsylvania.

The use of probabilistic simulations predated the existence of a computer. The (perhaps apocryphal) case of the 18th century Buffon's needle is one example of how a "manual" simulation can be used to estimate a parameter of interest, in this case $\pi$. A more interesting, and better documented, example of simulation appears in Lord Kelvin's 1901 paper *Nineteenth Century Clouds Over the Dynamical Theory of Heat and Light*[2] [**43**], in which he described a method to estimate velocity distributions using simulated values obtained from a carefully constructed set of cards. There is also

[1] Others listed include the QR algorithm for eigenvalue calculation, the simplex method, quicksort, and the fast Fourier transform.

[2] Aside from describing Monte Carlo-like simulations, this paper had a significant role in the history of modern physics. Kelvin outlines the central problems of physics at the beginning of the twentieth century, namely the so-called "ultraviolet catastrophe" and the Michelson-Morely "anomaly" regarding the speed of light.

evidence that Enrico Fermi [**46, 58**] used manual Monte Carlo-like methods during the 1930s in his early work on nuclear fission.[3]

At Los Alamos credit for introducing probabilistic simulation goes to Stanislaw Ulam. As the story goes [**20, 64**], in 1946 Ulam was confined to bed to recover from an illness. To pass the time he played Canfield Solitaire, a form of solitaire for which the outcome is determined once the cards are shuffled and dealt. As he played, Ulam wondered how to determine the probability of winning a game. Clearly, this was an intractable calculation, but he imagined programming the ENIAC to simulate a random shuffle and then apply the rules of the game to determine the outcome. Repeating this a large number of times would give an empirical estimate of the probability of winning. Analyzing solitaire did not justify using Los Alamos's precious computing resources, but Ulam saw that this new approach could be used in realistic and important settings. He conveyed the idea to his friend John von Neumann.

In 1947, von Neumann and others were working on methods to estimate neutron diffusion and multiplication rates in fission devices (i.e., nuclear bombs) [**20**]. Following Ulam's suggestion, von Neumann proposed a simple plan: create a relatively large number of "virtual" neutrons and use the computer to randomly simulate their evolution through the fissionable material. When finished, count the number of neutrons remaining to estimate the desired rates. In modern terms, the scale was extremely modest: a simulation of just 100 neutrons with 100 collisions each required about five hours of computing time on the ENIAC. Nonetheless, the utility of this approach was immediately apparent. From this point forward, randomized simulations—soon to be called *Monte Carlo methods*—were an important technique in physics.

Apparently, Nicholas Metropolis was responsible for the name *Monte Carlo methods*.

> . . . I suggested an obvious name for the statistical method—a suggestion not unrelated to the fact that Stan (Ulam) had an uncle who would borrow money from relatives just because he had to go to Monte Carlo. The name seems to have endured. [**46**]

This new approach first appeared in Ulam and Metropolis's 1949 paper *The Monte Carlo Method* [**49**].

> We want to now point out that modern computing machines are extremely well suited to perform the procedures described. In practice, a set of values of parameters characterizing a particle is represented, for example, by a set of numbers punched on a card. . . . It may seem strange that the machine can simulate the production of a series of random numbers, but this is indeed possible. In fact, it suffices to produce a sequence of numbers between 0 and 1 which have a uniform distribution . . . [4]

[3] In 1955, fellow Nobel laureate Emilio Segrè recalled

> . . . Fermi acquired, by the combined means of empirical experience, Monte Carlo calculation, and more formal theory, that extraordinary feeling for the behavior of slow neutrons . . . [**58**]

[4] A standard approach of the era was the so-called "middle third" method. Let $r_k$ be an $n$-digit random number. Square it and extract the middle $n$ digits to form $r_{k+1}$. Linear congruential methods would be developed shortly thereafter by Lehmer and others.

From our perspective, it is perhaps difficult to appreciate the revolutionary nature of simulation as an alternative to analytical methods. But at the time, few mathematicians or physicists had any experience with the computer, much less simulation.

In addition to sampling from the uniform distribution, there soon emerged ways to sample from other probability distributions. For many of the standard distributions (e.g., the normal distribution), mathematical transformations of the uniform distribution sufficed. For more general distributions, in particular ones arising from physical models, more sophisticated techniques were needed.[5] One early method, also due to von Neumann, became what we now call *acceptance-rejection* sampling. These methods were far from universal and not well suited for higher-dimensional probability distributions. MCMC methods overcame these limitations. The key was the use of a Markov chain, which we now briefly review.

**2.2. Markov Chains.** Given a finite state (configuration) space $\mathbb{S} = \{1, 2, \ldots, N\}$, a *Markov chain* is a stochastic process defined by a sequence of random variables, $X_i \in \mathbb{S}$, for $i = 1, 2, \ldots$ such that

$$\text{Prob}(X_{k+1} = x_{k+1} \mid X_1 = x_1, \ldots, X_k = x_k) = \text{Prob}(X_{k+1} = x_{k+1} \mid X_k = x_k).$$

In other words, the probability of being in a particular state at the $(k+1)$st step only depends on the state at the $k$th step. We only consider Markov chains for which this dependence is independent of $k$ (that is, time-homogeneous Markov chains). This gives an $N \times N$ *transition matrix* $\mathbf{P} = (\mathbf{p}_{ij})$ defined by

$$\mathbf{p}_{ij} = \text{Prob}(X_{k+1} = j \mid X_k = i).$$

Note that for $i = 1, 2, \ldots, N$,

$$\sum_{j=1}^{N} \mathbf{p}_{ij} = 1.$$

The $(i, j)$-entry of the $K$th power of $\mathbf{P}$ gives the probability of transitioning from state $i$ to state $j$ in $K$ steps.

Two desirable properties of a Markov chain are:

- It is *irreducible*: for all states $i$ and $j$, there exists $K$ such that $(P^K)_{i,j} \neq 0$.
- It is *aperiodic*: for all states $i$ and $j$, $\gcd\{K : (P^K)_{i,j} > 0\} = 1$.

An irreducible, aperiodic Markov chain must have a unique distribution $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ on the state space $\mathbb{S}$ ($\pi_i =$ the probability of state $i$) with the property that

$$\pi = \pi \mathbf{P}.$$

We say that the Markov chain is *stable on the distribution* $\pi$, or that $\pi$ is the *stable distribution* for the Markov chain.

MCMC methods depend on the observation:

> If $\pi$ is the stable distribution for an irreducible, aperiodic Markov chain, then *we can use the Markov chain to sample from* $\pi$.

[5] See [**15**] for a thorough overview of modern sampling methods.

To obtain a sample, select $s_1 \in \mathbb{S}$ arbitrarily. Then for any $k > 1$, if $s_{k-1} = i$, select $s_k = j$ with probability $\mathbf{p}_{ij}$. The resulting sequence $s_1, s_2, \ldots$ has the property that as $M \to \infty$,

$$\frac{|\{k : k \leq M \text{ and } s_k = j\}|}{M} \to \pi_j \tag{1}$$

with probability one.

Any large (but finite) portion of this sequence approximates a sample from $\pi$. Often, one discards the first $m$ terms of the sequence, and uses the "tail" of the sequence

$$s_{m+1}, s_{m+2}, \ldots, s_M$$

as the sample.

However they are obtained, samples from $\pi$ provide a way to approximate properties of $\pi$. For example, suppose $f$ is any real-valued function on the state space $\mathbb{S}$ and we wish to approximate the expected value

$$E[f] = \sum_{i=1}^{N} f(i)\pi_i.$$

To do so, select a sample $s_1, s_2, \ldots, s_M$ from $\pi$ and the ergodic theorem guarantees that

$$\frac{1}{M}\sum_{i=1}^{M} f(s_i) \to E[f] \tag{2}$$

as $M \to \infty$ with the convergence $O(M^{-1/2})$ [**34**].

Given the transition matrix for an irreducible, aperiodic Markov chain, it is a standard exercise to determine its stable distribution. We are keenly interested in the inverse problem:

> Given a distribution $\pi$ on a finite state space, find an irreducible, aperiodic Markov chain which is stable on $\pi$.

The solution is the Metropolis algorithm.

**2.3. Statistical Mechanics and the Boltzmann Distribution.** The Metropolis algorithm was motivated by the desire to discern properties of the *Boltzmann distribution* from statistical mechanics, the branch of physics concerned with the average behavior of large systems of interacting particles. Let us briefly develop some of the fundamental ideas behind the Boltzmann distribution.

A state of the particles is described by a *configuration* $\omega$ taken from the *configuration space* $\Omega$. The configuration space can be infinite or finite, continuous or discrete. For example, we might start with $N$ interacting particles, each described by its position and velocity in three-dimensional space. In this case $\Omega$ is an infinite, continuous subset of $\mathbb{R}^{6N}$. Alternatively, $\Omega$ could be described by taking a bounded subset, $\Lambda$, of the integer lattice in the plane and to each site attaching a value, say $\pm 1$. The value at a site might indicate the presence of a particle there, or it might indicate an orientation (or "spin") of a particle at the site. If $|\Lambda| = N$, then the configuration space consists of all $2^N$ possible assignments of values to sites in $\Lambda$.

The physics of a configuration space $\Omega$ is described by an *energy function* $E : \Omega \to \mathbb{R}^+$. We say that $E(\omega)$ is the energy of a configuration $\omega$. For the continuous example above, energy could reflect the sum of gravitational potential energies. For the discrete example, the energy could reflect the total influence that neighboring particles exert on each other, as in the Ising model, which we will look at shortly.

A fundamental principle of statistical physics is that Nature seeks low-energy configurations. The random organization of molecules in a room is governed by this principle. Rarely observed configurations (e.g., all of the molecules gathering in a corner of the room) have high energies and hence very low probabilities. Common configurations (e.g., molecules isotropically distributed throughout the room) have low energies and much higher probabilities, high enough so that they are essentially the only configurations ever observed.

For a system at equilibrium, the relative frequency of a configuration $\omega$ is given by its *Boltzmann weight*,

$$e^{-E(\omega)/kT}, \tag{3}$$

where $T$ is the temperature and $k$ is Boltzmann's constant.

For any $\omega \in \Omega$, its *Boltzmann probability*, $\text{Boltz}(\omega)$, is

$$\text{Boltz}(\omega) = \frac{e^{-E(\omega)/kT}}{Z}. \tag{4}$$

The denominator

$$Z = \sum_{\omega' \in \Omega} e^{-E(\omega')/kT}$$

is called the *partition function*. In any realistic setting, the partition function is analytically and computationally intractable. This intractability single-handedly accounts for the dearth of analytic, closed-form results in statistical mechanics.

The relationship between energy and probability leads to expressions for many interesting physical quantities. For example, the total energy of the system, $\langle E \rangle$, is the expected value of the energy function $E(\omega)$ and is defined by

$$\langle E \rangle = \sum_{\omega \in \Omega} E(\omega)\text{Boltz}(\omega) = \frac{\sum_{\omega \in \Omega} E(\omega) e^{-E(\omega)/kT}}{Z}. \tag{5}$$

Many other physical quantities are defined similarly. In each case there is no avoiding the partition function $Z$.

Expressions such as (5) could be naively approximated using Monte Carlo sampling. To do so, generate a sample $\omega_1, \omega_2, \ldots, \omega_M$ *uniformly* from $\Omega$, and estimate both the numerator and denominator of (5) separately, resulting in

$$\langle E \rangle \approx \frac{\sum_{i=1}^{M} E(\omega_i) e^{-E(\omega_i)/kT}}{\sum_{i=1}^{M} e^{-E(\omega_i)/kT}}.$$

Metropolis et al. understood the limitations of sampling uniformly from the configuration space and proposed an alternative approach.

> This method is not practical ... since with high probability we choose a configuration where $\exp(-E/kT)$ is very small; hence a configuration with very

low weight. The method we employ is actually a modified Monte Carlo scheme where, instead of choosing configurations randomly, then weighting them with $\exp(-E/kT)$, we choose configurations with probability $\exp(-E/kT)$ and weight them evenly. [**48**]

In other words, it would be much better to sample from $\Omega$ so that $\omega$ is selected with probability Boltz($\omega$). If this can be done, then for any such sample $\omega_1, \omega_2, \ldots, \omega_M$,

$$\frac{1}{M}\sum_{i=1}^{M} E(\omega_i) \to \langle E \rangle$$

with, as noted earlier, $O(M^{-1/2})$ convergence. The challenge is to sample from the Boltzmann distribution.

**2.4. The Metropolis Algorithm.** The genius of the Metropolis algorithm is that it creates an easily computed Markov chain which is stable on the Boltzmann distribution. Using this Markov chain, a sample from the Boltzmann distribution is easily obtained. The construction requires only the Boltzmann weights (3), not the full probabilities (4), hence avoiding the dreaded partition function. To appreciate the motivation for the Metropolis algorithm, let's recreate Metropolis et al.'s argument from their 1953 paper.

The setting for the Metropolis algorithm includes a large but finite configuration space $\Omega$, an energy function $E$, and a fixed temperature $T$. Let $\tilde{\Omega}$ be any sample of configurations selected *with replacement* from $\Omega$. It is possible, even desirable, to allow $\tilde{\Omega}$ to be larger than $\Omega$. By adding and removing configurations, we want to modify $\tilde{\Omega}$ so that it becomes (approximately) a sample from the Boltzmann distribution. Suppose $|\tilde{\Omega}| = \tilde{N}$ and let $N_\omega$ denote the number of occurrences of $\omega$ in $\tilde{\Omega}$. To say that the sample perfectly reflects the Boltzmann distribution means

$$\frac{N_\omega}{\tilde{N}} \propto e^{-E(\omega)/kT},$$

or equivalently, for any two configurations $\omega$ and $\omega'$,

$$\frac{N_{\omega'}}{N_\omega} = \frac{e^{-E(\omega')/kT}}{e^{-E(\omega)/kT}} = e^{-\Delta E/kT}, \tag{6}$$

where $\Delta E = E(\omega') - E(\omega)$. Notice that this ratio does not depend on the partition function.

To get from an arbitrary distribution of energies to the desired Boltzmann distribution, imagine applying our yet-to-be-discovered Markov chain on $\Omega$ to all of the configurations in $\tilde{\Omega}$ simultaneously. Start by selecting a *proposal transition*: any irreducible, aperiodic Markov chain on $\Omega$. Denote the probability of transitioning from a configuration $\omega$ to a configuration $\omega'$ by $P_{\omega,\omega'}$. As well, assume that the proposal transition is symmetric, that is, $P_{\omega,\omega'} = P_{\omega',\omega}$.

Consider configurations $\omega$ and $\omega'$ where $E(\omega) < E(\omega')$. Allow transitions from configurations with high energy $E(\omega')$ to configurations with low energy $E(\omega)$ whenever they are proposed; the number of times this occurs is simply

$$P_{\omega',\omega} N_{\omega'}.$$

By itself, this is just a randomized version of the steepest descent algorithm; any "downhill" transition is allowed.

In order to have any hope of reaching equilibrium, we must occasionally allow "uphill" transitions from configurations with low energy $E(\omega)$ to ones with high energy $E(\omega')$, that is, with some probability $\text{Prob}(\omega \to \omega')$. The number of times such a move is proposed is $P_{\omega,\omega'}N_\omega$ and hence the number of moves that actually occur is

$$P_{\omega,\omega'}N_\omega \text{Prob}(\omega \to \omega').$$

Since $P_{\omega,\omega'} = P_{\omega',\omega}$, the net flux between configurations with energy $E(\omega)$ and those with energy $E(\omega')$ is

$$P_{\omega,\omega'}\big[N_{\omega'} - N_\omega \text{Prob}(\omega \to \omega')\big]. \tag{7}$$

If (6) holds, that is, if the distribution of energies in $\tilde{\Omega}$ perfectly reflects the Boltzmann distribution, then the flux (7) should be zero. The result is what physicists call the *detailed balance*. This implies that the uphill probability must be

$$\text{Prob}(\omega \to \omega') = e^{-\Delta E/kT}.$$

This choice of occasional "uphill" transitions provides the magic of the Metropolis algorithm.

This process will also drive an arbitrary distribution of energies toward the Boltzmann distribution. Suppose there are too many configurations with high energy $E(\omega')$ relative to configurations with the low energy $E(\omega)$, that is,

$$\frac{N_{\omega'}}{N_\omega} > e^{-\Delta E/kT}.$$

In this case, the flux (7) is positive and there will be more transitions from configurations with energy $E(\omega)$ to those with energy $E(\omega')$ than in the other direction. Accordingly, the distribution of energies in $\tilde{\Omega}$ will move toward the Boltzmann distribution. Repeating this process a large number of times will produce a set of configurations whose distribution of energies approximates the Boltzmann distribution.

Based on this argument and physical intuition, Metropolis et al. were satisfied that their algorithm would produce samples from the Boltzmann distribution. More mathematically rigorous proofs of the convergence to the stable distribution would soon appear [**34, 35**]. Other important practical considerations, particularly understanding the rate of convergence, would have to wait longer.[6]

We now formally state the Metropolis algorithm. Assume a suitable proposal transition has been selected. For an arbitrary $\omega \in \Omega$ define the transition to a configuration $\omega^*$ as follows.

**Step 1.** Select $\omega'$ according to the proposal transition.
**Step 2A.** If $E(\omega') \le E(\omega)$, or equivalently, $\text{Boltz}(\omega') \ge \text{Boltz}(\omega)$, let $\omega^* = \omega'$. In other words, always move to lower energy (higher probability) configurations.
**Step 2B.** If $E(\omega') > E(\omega)$, or equivalently, $\text{Boltz}(\omega') < \text{Boltz}(\omega)$, let $\omega^* = \omega'$ with probability

$$\frac{\text{Boltz}(\omega')}{\text{Boltz}(\omega)} = e^{-\Delta E/kT}. \tag{8}$$

Otherwise, $\omega^* = \omega$.

[6]Metropolis et al. knew the rate of convergence was an open question: "[Our] argument does not, of course, specify how rapidly the canonical distribution is approached." [**48**]

Several observations are in order:

- This process defines an irreducible, aperiodic Markov chain on the configuration space $\Omega$.
- The ratio (8) is crucial to the computational utility of the Metropolis algorithm in that it avoids the intractable partition function.
- The steps in the chain are easily computable, or at least as easily computable as the proposal transition, $E(\omega)$, and, most importantly, $\Delta E = E(\omega') - E(\omega)$. In many settings, $\Delta E$ is extremely simple to compute; often it is independent of $|\Omega|$.
- The Markov chain defined by the Metropolis algorithm can be implemented without knowing the entire transition matrix.

The first application of the Metropolis algorithm in [**48**] was to analyze the so-called *hard spheres* model, a simple model of nonoverlapping molecules (e.g., a gas). Despite its apparent simplicity, the hard spheres model has proven to be a rich source of insight for statistical physicists. Using their new algorithm on 224 two-dimensional discs, Metropolis et al. allowed the system to evolve from an ordered state to a state close to equilibrium. The results were encouraging; the physical values they estimated agreed nicely with estimates obtained by traditional analytic methods. Best of all, the calculation times were reasonable. A single data point (of which there were hundreds) on a curve representing information about the hard spheres model only took about four or five hours of computing time on Los Alamos's MANIAC (Mathematical Analyzer, Numerator, Integrator, and Calculator).

**2.5. The Metropolis Algorithm and the Ising Model.** For a more illustrative application of the Metropolis algorithm, consider the two-dimensional *Ising model*. The Ising model has been extensively studied in both physics and mathematics. For more on the history and features of the Ising model, see [**6, 11**]. In addition to illustrating the effectiveness of the Metropolis algorithm, the Ising model plays a fundamental role in Geman and Geman's work on image reconstruction.

The Ising model can be thought of as a simple model of a ferromagnet in that it captures the tendency for neighboring sites to align with each other or with an external magnetic field. Formally, the two-dimensional Ising model is defined on a bounded planar lattice with $N$ sites. At each lattice site, there is a "spin" represented by $\pm 1$. A configuration is given by $\omega = (\omega_1, \omega_2, \ldots, \omega_N)$, where $\omega_i = \pm 1$ is the spin at the $i$th site; hence $|\Omega| = 2^N$. The energy of a configuration is defined as

$$E_{\text{ising}}(\omega) = -J \sum_{\langle i,j \rangle} \omega_i \omega_j - H \sum_{i=1}^{N} \omega_i \tag{9}$$

where $J > 0$ represents the nearest-neighbor affinity, $H > 0$ represents the external field, and $\langle i, j \rangle$ indicates that sites $i$ and $j$ are nearest neighbors, that is, sites that share either a horizontal or vertical bond. We will assume there is no external field (i.e., $H = 0$) and that $J = 1$.

One reason that the Ising model has long interested statistical physicists is that it exhibits a *phase transition*. Mathematically, a phase transition occurs when a quantity undergoes a dramatic change as a parameter passes through a *critical value*. The most familiar example of a phase transition occurs in water as it freezes or boils; in this case, the density of water changes dramatically as the temperature $T$ passes through the critical value of $T_c = 0$ (or $T_c = 100$).

An important phase transition for the two-dimensional Ising model occurs in the *magnetization*. For a configuration $\omega$, define

$$M(\omega) = \sum_{i=1}^{N} \omega_i.$$

The magnetization $\langle M \rangle_T$ at a temperature $T$ is the expected value of $M(\omega)$:

$$\begin{aligned} \langle M \rangle_T &= \sum_{\omega \in \Omega} M(\omega) \text{Boltz}(\omega) \\ &= \frac{1}{Z} \sum_{\omega \in \Omega} M(\omega) e^{-E_{\text{ising}}(\omega)/kT}. \end{aligned} \tag{10}$$

At high temperatures, states are essentially uniformly distributed and hence $\langle M \rangle_T$ is zero; in particular, there is almost no correlation between sites. However, as the temperature is lowered, *spontaneous magnetization* occurs: there is a critical temperature, $T_c$, below which sites influence each other at long ranges. One of the most celebrated results of statistical physics is Osager's exact calculation of the critical temperature for the two-dimensional Ising model:[7]

$$kT_c/J = \frac{2}{\ln(1+\sqrt{2})} \approx 2.269.$$

Let's use the Metropolis algorithm to visualize the phase transition in the magnetization for an Ising lattice with $N$ sites. To implement **Step 1**, we need a proposal transition process between configurations $\omega$ and $\omega'$. A simple way to do this is to pick a lattice site $i$ uniformly from $1, 2, \ldots, N$. At site $i$, with probability $1/2$, flip the spin $\omega_i$ to its opposite value; otherwise keep its current value. Notice that $\omega'_j = \omega_j$ for all $j \neq i$; that is, only the one site, $\omega_i$, is affected. This proposal transition between configurations is irreducible, aperiodic, and symmetric.

For **Step 2**, we must decide whether to keep the proposed $\omega'$. The important quantity is the change in energy:

$$\begin{aligned} \Delta E &= E_{\text{ising}}(\omega') - E_{\text{ising}}(\omega) \\ &= (\omega'_i - \omega_i) \sum_{\langle i,j \rangle} \omega_j, \end{aligned}$$

where the sum is over the four nearest neighbors of the $i$th site. Hence $\Delta E$ only depends on *the spins at the four sites neighboring the affected site* and therefore the computational cost of updating a site is both small and independent of the size of the lattice. This dependence on the local structure, the so-called *local characteristics*, is a recurring part of the Metropolis algorithm and MCMC methods in general.

Another recurring—but vexing—theme of MCMC methods is convergence. In general, it is extremely hard to determine how many iterations of the algorithm are required to reasonably approximate the target distribution. Also, an unavoidable feature of a Markov chain is sequential correlation between samples. This means it can take

[7] Surprisingly, there is no phase transition for the Ising model in one dimension. For a purely combinatorial argument for the existence of a phase transition for the two-dimensional Ising model, see [**44**].

a long time to traverse the configuration space, especially near the critical temperature where things are most interesting.[8] See Diaconis [**17**] for a survey of convergence results related to Ising-like models both near and far from the critical temperature.

**2.6. An Application of the Metropolis Algorithm.** Figure 1 shows two snapshots of a $200 \times 200$ Ising lattice; black indicates a spin of $+1$ and white a spin of $-1$. The lattice on the left is above the critical temperature for a phase transition, while the lattice on the right is below it. In each case, the Metropolis algorithm ran long enough so that the resulting sequence of states represented a sample from the Boltzmann distribution. On the left it is visually evident that there is little correlation of spin values of sites located some distance from each other. On the right there is a clear long-range correlation between spins. This qualitative difference reflects two distinct phases of the Ising model.

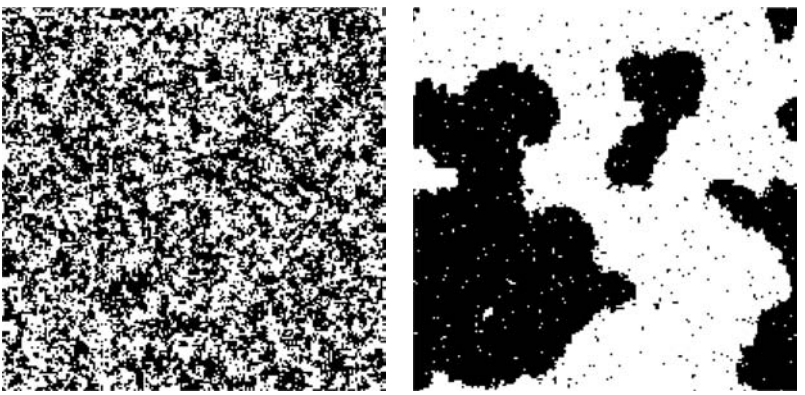

**Figure 1.** A $200 \times 200$ Ising model simulated using the Metropolis algorithm. The image on the left is above $kT_c/J \approx 2.269$ and exhibits very little long-range correlation between sites. The image on the right is below $kT_c/J \approx 2.269$ and shows a clear long-range correlation between sites.

**2.7. Attribution.** So who is responsible for the Metropolis algorithm? That there are five co-authors of [**48**] (including two husband-and-wife teams) makes any claim of "ownership" murky. Given the passage of time, the answer will probably never be known. Perhaps the final word on the subject belongs to Marshall Rosenbluth, the last survivor of the five co-authors, who commented on this question a few months before his death in 2003. At a conference celebrating the 50th anniversary of the Metropolis algorithm, he stated "Metropolis played no role in the development other than providing computer time" [**32**]. Rosenbluth went on to say that the idea to use Monte Carlo methods came from conversations with Ulam and von Neumann and that Teller made useful suggestions that led to replacing ordinary averages with averages weighted by the Boltzmann distribution. As well, he claimed that he and his wife and co-author, Arianna, did the important work.

There is evidence that essentially the same idea was independently proposed by Bernie Alder, Stan Frankel, S. G. Lewinson, and J. G. Kirkwood working at California

[8]The Swendsen-Wang algorithm [**60**] provides a significant, and elegant, solution to the second problem. This version of the Metropolis algorithm updates entire clusters of like spins. As a result, it traverses the configuration space much more rapidly, even at and below the critical temperature. The Swendsen-Wang algorithm is particularly interesting in that it introduces "bond" variables, an idea that appears in the image reconstruction work of Geman and Geman and also in Bayesian hierarchical models in statistics.

Institute of Technology and the Lawrence Livermore National Laboratories in the late 1940s. When asked, Alder is quite clear about their role.

> My guess is we did it first at Cal Tech. It's not that difficult to come up with that algorithm, which, by the way, I think is one of, if not THE, most powerful algorithms.... The fact is, we never published—you can't publish something your boss doesn't believe in! [**50**]

Support from their boss was not the only thing Alder and his colleagues lacked. They also did not have easy access to the tool most crucial to an implementation of the algorithm: a computer.

Interestingly, in 1947 Frankel and Metropolis had co-authored one of the first papers demonstrating the usefulness of the computer to perform numerical (not Monte Carlo) integration [**22**]. No doubt, the interplay between all these individuals during this era makes a strong argument that credit for what we now call the Metropolis algorithm should be shared among many.

**2.8. Interlude.** From the 1950s to the 1980s, most of the interest in the Metropolis algorithm came from the physics community. One exception was Hammersley and Handscomb's 1964 classic *Monte Carlo Methods* [**34**]. This delightful—and still relevant—monograph describes that era's state of the art of Monte Carlo methods, including a short survey of Markov chain Monte Carlo methods in statistical mechanics.

Physicists of this era were busy developing generalizations of the Metropolis algorithm, many of which were applied to spin models such as the Ising model. One of the first of these was the "heat bath" proposed by Glauber[9] in 1963 [**31**]. Glauber's algorithm moves through the lattice sites sequentially. At the $i$th site, the spin $\omega_i$ is assigned according to the local Boltzmann weight

$$\text{Prob}(\omega_i = s) = e^{-(s\sum_{\langle i,j\rangle}\omega_j)/kT},$$

where the sum, as usual, is over the nearest neighbor sites of $i$. Interestingly, Glauber's motivation was to understand analytical properties of the time-dependent (nonequilibrium) dynamics of spin models, not to develop a new computational tool. A decade later, Flinn [**21**] described a similar "spin-exchange" algorithm to computationally investigate phase transitions in the Ising model. Creutz in 1979 [**13**] showed how single-site updates could be applied to $SU(2)$ (special unitary group of degree 2) gauge theories.

Another generalization appeared in 1965 when A. A. Barker [**3**] introduced an alternative to the Metropolis construction, resulting in one more Markov chain with the Boltzmann distribution as its stable distribution. The existence of these variants raised the questions: How many Metropolis-like algorithms are there? Among all the variants, which one, if any, is best?

Answers to these questions appeared in the 1970s, starting with the work of the statistician W. K. Hastings. He was the first to see that the Metropolis algorithm was a (perhaps, *the*) general-purpose sampling algorithm. In an interview, Hastings recalled how he came across the algorithm.

[9]Glauber is also known for his contributions to the quantum theory of optical coherence, work for which he shared the 2005 Nobel Prize.

> [The chemists] were using Metropolis's method to estimate the mean energy of a system of particles in a defined potential field. With six coordinates per particle, a system of just 100 particles involved a dimension of 600. When I learned how easy it was to generate samples from high dimensional distributions using Markov chains, I realised how important this was for Statistics and I devoted all my time to this method and its variants which resulted in the 1970 paper. [**57**]

This 1970 paper was *Monte Carlo Sampling Methods using Markov Chains and Their Applications* [**35**] in which Hastings was able to distill the Metropolis algorithm down to its mathematical essentials. He also demonstrated how to use the Metropolis algorithm to generate random samples from a variety of standard probability distributions, as well as in other settings, such as from the group of orthogonal matrices. The importance of this paper was not immediately understood; recognition would have to wait for Gelfand and Smith's work twenty years later. In statistical circles, the Metropolis algorithm is now often referred to as the *Metropolis-Hastings algorithm.*

Let's briefly look at Hastings's generalization of the Metropolis algorithm. Given a distribution $\pi$ from which we want to sample, select any Metropolis-like proposal transition, $\mathbf{Q} = (\mathbf{q}_{ij})$, on the state space $\mathbb{S}$. Unlike in the original Metropolis algorithm, *it does not need to be symmetric*. Define the transition matrix $\mathbf{P} = (\mathbf{p}_{ij})$ by

$$\mathbf{p}_{ij} = \begin{cases} \mathbf{q}_{ij}\alpha_{ij} & \text{if } i \neq j, \\ 1 - \sum_{k \neq i} \mathbf{p}_{ik} & \text{if } i = j, \end{cases} \tag{11}$$

where $\alpha_{ij}$ is given by

$$\alpha_{ij} = \frac{s_{ij}}{1 + \frac{\pi_i}{\pi_j}\frac{q_{ij}}{q_{ji}}}.$$

The values $s_{ij}$ can be quite general, so long as (i) $s_{ij} = s_{ji}$ for all $i$, $j$ and (ii) $\alpha_{ij} \in [0, 1]$. For any such choice of $s_{ij}$, it is easy to verify that $\pi$ is the unique stable distribution for $\mathbf{P}$. For a symmetric $\mathbf{Q}$, a simple choice of $s_{ij}$ recovers the original Metropolis algorithm.

For a given distribution $\pi$, different choices of the $s_{ij}$ lead to qualitatively different Metropolis-like algorithms, all of which produce a Markov chain stable on $\pi$. Why does only the original Metropolis(-Hasting) algorithm live on? The reason was provided by Hastings's student, P. H. Peskun. Peskun [**52**] showed that among all choices of the $s_{ij}$, the variance of the estimate given in (2) is asymptotically minimal for the choice that leads to the Metropolis algorithm. Whether by luck or intuition, the first example of a Markov chain Monte Carlo method proved to be the best.

**3. SIMULATED ANNEALING AND COMBINATORIAL OPTIMIZATION: KIRKPATRICK ET AL., 1983.** Despite the popularity of the Metropolis algorithm in statistical physics and Hastings's observation of its potential as a general-purpose sampling tool, before 1980 the algorithm was little known in other circles. The situation changed with the appearance of *Optimization by Simulated Annealing* [**45**] by Scott Kirkpatrick, C. D. Gelatt, and M. P. Vecchi in 1983. At almost the same time V. Černý, a Czech applied mathematician, independently developed equivalent ideas in his 1985 paper *Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm* [**9**]. Kirkpatrick et al.'s work is better known and

rightfully so; they did more to develop the mathematics of annealing and applied it to a larger collection of problems. Although we will focus primarily on their work, Černý's paper is significant in its own right and deserves to be more widely read.

Kirkpatrick et al. were part of IBM's Thomas Watson Research Center. They were working on problems in *combinatorial optimization*, a type of deterministic problem for which the Metropolis algorithm was unexpectedly effective. Combinatorial optimization problems share two features:

1. An objective (cost) function for which a global minimum value is sought.
2. A discrete (often finite, but large) search space in which one looks for the global minimum. In practice, approximations to the global minimum are the best one can expect.

A standard example of a combinatorial optimization problem is the *traveling salesman problem* (TSP) where the goal is to minimize the distance of a tour through a set of vertices. The search space consists of possible tours and the objective function is the total distance of a tour. Like many combinatorial optimization problems, the TSP (when recast as a decision problem) is NP-complete.

Kirkpatrick and the other authors were trained as statistical physicists, so it was natural for them to think of the objective function as an energy function. Knowing that Nature seeks low energy configurations, they considered ways to use the Metropolis algorithm to select low energy configurations from the search space. The challenge, they discovered, was to find a way to properly utilize temperature $T$, a quantity for which there is no natural analog in a combinatorial optimization setting. For large values of $T$, the Metropolis algorithm produced an essentially uniform distribution, hence was nothing more than a random search. For small values of $T$, the Metropolis algorithm was susceptible to becoming trapped near local minima far removed from the desired global minimum. An understanding of how to properly utilize $T$ required insights from statistical mechanics. We will construct Kirkpatrick et al.'s original argument to see how this is done.

**3.1. Circuit Design and Spin Glasses.** The motivating question for Kirkpatrick et al. was not the TSP, but how to place circuits (i.e., transistors) on computer chips efficiently. Circuits on the same chip communicate easily, but there is a substantial communication penalty for signals connecting circuits on different chips. The goal is to place the circuits in a way that minimizes the total communication cost with the constraint that there must be a (roughly) equal number of circuits on each chip.

To formulate this problem mathematically, suppose there are $N$ circuits that must be placed on two separate chips. A configuration $\omega$ is given by the $N$-tuple

$$\omega = (\omega_1, \omega_2, \ldots, \omega_N),$$

where $\omega_i = \pm 1$ indicates the chip on which the $i$th circuit is placed. The value $a_{ij}$ indicates the number of signals (connections) between circuits $i$ and $j$.

Following Kirkpatrick et al., represent the between-chip communication cost as

$$\sum_{i>j} \frac{a_{ij}}{4}(\omega_i - \omega_j)^2. \tag{12}$$

The cost of an imbalance between the number of circuits on each of the two chips can be expressed as

$$\lambda \left( \sum_i \omega_i \right)^2, \tag{13}$$

where $\lambda > 0$ is the imbalance "penalty."

Expanding (12), combining it with (13), and dropping all constant terms results in an objective function

$$C(\omega) = \sum_{i>j} \left( \lambda - \frac{a_{ij}}{2} \right) \omega_i \omega_j. \tag{14}$$

By the early 1980s, researchers at IBM had developed various algorithms to (approximately) minimize $C(\omega)$. As the number of transistors $N$ grew from several hundred to thousands (and beyond), these methods were proving less viable. As Kirkpatrick recalls,

> Previous methods were arcane, if you looked at them carefully they involved solving for conflicting objectives one after another. We knew we could do better than that. (Scott Kirpatrick, personal communication)

Fortunately, Kirkpatrick et al. knew of a model in statistical mechanics whose energy function bore a striking resemblance to (14). This model is called a *spin glass*.

Spin glasses are much like the Ising model but with a slightly different energy function

$$E(\omega) = \sum_{i>j} \left( U - U_{ij} \right) \omega_i \omega_j.$$

The analogy to (14) is immediate. The values of $U_{ij}$ represent local attractive (ferromagnetic) forces between neighboring states. These are in competition with long-range repulsive (anti-ferromagnetic) interactions represented by $U$. Spin glasses are called *frustrated* because they cannot have configurations which simultaneously satisfy both the attractive and repulsive requirements. As a result, the low energy ground states do not have extended regions of pure symmetry.

For spin glasses and other frustrated systems, Kirkpatrick knew that the Metropolis algorithm had to be carefully applied in order to identify low-temperature ground states. If the system is *quenched*, that is, the temperature is lowered too quickly, then it can settle into a state other than a ground state. A better approach is to *anneal*, that is, to slowly lower the temperature so the system can evolve gently to a ground state. This observation led Kirkpatrick et al. to *simulated annealing*.

> Using the cost function in place of the energy and defining configurations by a set of parameters $\{x_{ij}\}$, it is straightforward with the Metropolis procedure to generate a population of configurations of a given optimization problem at some effective temperature. This temperature is simply a control parameter in the same units as the cost function. The simulated annealing process consists of first "melting" the system being optimized at a high effective temperature, then lowering the temperature by slow stages until the system "freezes" and no further changes occur. At each temperature, the simulation must proceed long enough for the system to reach steady state. The sequence of temperatures and number of rearrangements of the $\{x_{ij}\}$ attempted to reach equilibrium at each temperature can be considered an annealing schedule. [**45**]

Kirkpatrick et al. applied this new technique to several realistic problems in circuit design, along with a demonstration of its effectiveness on the TSP. The results were impressive—clearly simulated annealing worked. As well, around the same time Černý produced similar results applying his version of simulated annealing to the TSP.

**3.2. After Kirkpatrick et al.** Since 1983 simulated annealing has become a standard technique in the applied mathematician's toolbox. The range of problems to which it has been applied is staggering. It works, to some degree, in both discrete and continuous settings. It has been used in almost every area of applied mathematics, including operations research, biology, economics, and electrical engineering. Combinatorial optimization is replete with algorithms that solve particular problems—or even special cases of particular problems—quite well. However, most are customized to fit the particular nuances of the problem at hand. Simulated annealing's popularity is due to a combination of its effectiveness and ease of implementation: given an objective function and a proposal transition, one can almost always apply simulated annealing.

Perhaps because of the lack of a strong mathematical framework, it took some time for simulated annealing to become accepted in the applied mathematics community. The first thorough empirical analysis of simulated annealing appeared in 1989 in a series of papers by Johnson, Aragon, McGeoch, and Schevon [**41, 42**]. See [**66, 65**] for an excellent discussion of some of the theoretical issues, a survey of the applications of simulated annealing (especially of the type considered by Kirkpatrick et al.), and more analysis of its performance relative to other algorithms. For an accessible description, along with a simple example of an application of simulated annealing, see [**1**].

The computational efficiency of simulated annealing depends on the relationship between the proposal transition and the energy function. A good proposal transition changes the energy function as little as possible, that is, $\Delta E$ is easily computed, often in a manner that is independent of the problem size. In the original circuit design problem the proposal transition consists of randomly selecting a circuit and moving it to the other chip. The cost of computing the change in energy is therefore independent of the problem size. The advantage of these efficient, local changes was demonstrated in the work of Geman and Geman, who used ideas from both the Metropolis algorithm and simulated annealing to attack problems in digital image reconstruction.

**3.3. An Application of Simulated Annealing.** The importance of local changes can be seen in an application of simulated annealing to the traveling salesman problem. In this setting a configuration $\omega$ is a tour of the $n$ vertices and is specified by a permutation of $(1, 2, \ldots, n)$.

A simple proposal transition is defined by randomly selecting two vertices $1 \leq i < j \leq n$ and reversing the direction of the path between them. This means if

$$\omega = (a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{j-1}, a_j, a_{j+1}, \ldots, a_n)$$

then

$$\omega' = (a_1, \ldots, a_{i-1}, a_j, a_{j-1}, \ldots, a_{i+1}, a_i, a_{j+1}, \ldots, a_n).$$

The change in distance (energy) is easily computed:

$$\Delta E = (|a_{i-1} - a_j| + |a_i - a_{j+1}|) - (|a_{i-1} - a_i| + |a_j - a_{j+1}|).$$

Figure 2 illustrates this implementation of simulated annealing on a TSP graph consisting of 500 vertices which were randomly placed in the unit square.