Abstract

The field of randomized algorithms has benefitted greatly from insights from statistical physics. We give examples in two distinct settings. The first is in the context of Markov chain Monte Carlo algorithms, which have become ubiquitous across science and engineering as a means of exploring large configuration spaces. One of the most striking discoveries was the realization that many natural Markov chains undergo phase transitions, whereby they are efficient for some parameter settings and then suddenly become inefficient as a parameter of the system is slowly modified. The second is in the context of distributed algorithms for programmable matter. Self-organizing particle systems based on statistical models with phase changes have been used to achieve basic tasks involving coordination, movement, and conformation in a fully distributed, local setting. We briefly describe these two settings to demonstrate how computing and statistical physics together provide powerful insights that apply across multiple domains.

1 Introduction

Statistical physics employs probabilistic techniques to study systems consisting of large populations. The underlying principles explain numerous physical phenomena, such as magnetism, changes in states of matter, thermal radiation, noise in electronic devices, and more (see, e.g., [16]). In addition, these scientific insights help explain collective behavior across disciplines, including interacting biological systems [22], colloidal mixtures from chemistry [3, 21], segregation models from economics [5, 25], and random graph models in combinatorics [9].

Throughout theoretical computer science, we also find many examples where a statistical physics perspective has enriched the design and analysis of algorithms. A significant example concerns the role of phase transitions, showing how micro-scale behavior can induce global, macro-scale changes to a system (see, e.g., [4, 7, 26]). For example, phase transitions in random structures allow us to identify emergent characteristics of a configuration space, such as the birth of the giant component [19]. Moreover, Markov chains have been shown to undergo phase changes in their convergence times, transitioning from disordered phases, where they converge to (near) stationarity in polynomial time, to ordered phases that require exponential time [7, 10, 26]. More recently, algorithms exhibiting particular phase changes from disordered gaseous or liquid phases to ordered solid phases have proven effective for the design of distributed algorithms for robot swarms and active matter, where we seek collective organization achieving certain tasks [1, 11, 12].
Physical systems define probability measures favoring configurations that minimize energy. Each configuration $\sigma$ has energy determined by a Hamiltonian $H(\sigma)$ and a corresponding weight $w(\sigma) = e^{-B H(\sigma)}$, where $B = 1/T$ is inverse temperature. The Gibbs (or Boltzmann) distribution assigns probabilities proportional to their weight $w(\sigma)$, where configurations with the least energy $H(\sigma)$ have the highest weight and are most likely. However, if there are few of these higher weight configurations, the sample space may be dominated instead by those with small weight, simply because there are many more of them (i.e., there is higher entropy), giving these configurations much higher probability overall, even if they are individually less probable. The thermodynamic properties of a physical system, such as specific heat and free energy, are derived from statistical properties of these distributions, and discontinuities in any of these quantities indicate a phase transition between states of matter.

We will provide a small window into the rich marriage between statistical physics and algorithms in the context of Markov chains and programmable matter. Markov chains can become prohibitively slow once the energy from the Hamiltonian outweighs the effects of entropy and the system transitions to an ordered state. In contrast, for programmable matter, we purposefully design algorithms that achieve distinct collective behaviors in their disordered and ordered phases, leading to robust distributed algorithms for self-organizing particle systems.

## 2 Local Markov Chains

Markov chain Monte Carlo (MCMC) algorithms are ubiquitous throughout science and engineering, providing useful tools for approximate counting, combinatorial optimization and modeling. The main idea is to perform a random walk among a set of configurations so that samples drawn from the limiting distribution are meaningful. For this to be useful, these algorithms need to be efficient, and indeed bounding the convergence time of a Markov chain is often the critical step in establishing the efficiency of approximation algorithms based on random sampling. For example, if $G = (V_1, V_2, E)$ is a bipartite graph with $E \subseteq V_1 \times V_2$, then sampling perfect matchings on $G$ allows us to estimate the permanent of the adjacency matrix [20]. Calculating the permanent of a matrix was shown by Valiant to be #P-complete [27], or as hard as counting solutions to any NP-complete problem, so solutions that efficiently produce estimates approximating the exact count are the best we can expect.

Markov chains based on local moves, known as Glauber dynamics, are common in practice, primarily because of their simplicity. As an example, consider the following chain that can be used to sample from the set of independent sets in a given graph, known in statistical physics as the hard-core lattice gas model. Given a graph $G$, the state space $\Omega$ is the set of independent sets. We are also given an input parameter $\lambda$, known as the fugacity (or activity). Our goal is to sample from the Gibbs distribution

$$\pi(I) = \lambda^{|I|}/Z,$$

where $|I|$ is the size of independent set $I$ and $Z = \sum_{J \in \Omega} \lambda^{|J|}$ is the normalizing constant known as the partition function. We define the Glauber dynamics so that we can move between pairs of configurations that differ by a single vertex, and the celebrated Metropolis Algorithm tells us how to implement these moves so that we converge to the Gibbs distribution $\pi$, as follows. Starting at any configuration $\sigma \in \Omega$, say the empty independent set (with no vertices), we repeat the following: choose a vertex $v$ at random; if $v$ is in the current independent set, remove it with probability $\min(1, \lambda^{-1})/2$; if it is not in the independent
set, add it with probability $\min(1, \lambda)/2$, if possible; in all other cases, the independent set remains unchanged. It is simple to show that this chain is ergodic and converges to $\pi$, so our goal is to determine if it is efficient.

An interesting phenomenon occurs as $\lambda$ is varied. For small values of $\lambda$, Glauber dynamics converge quickly to stationarity, while for large values it is prohibitively slow. To see why, imagine the underlying graph $G$ is an $n \times n$ region of $\mathbb{Z}^2$. Large independent sets dominate the stationary distribution $\pi$ when $\lambda$ is sufficiently large and lie on one of the two sublattices (corresponding to each of the two colors of the checkerboard coloring on the dual lattice). When $\lambda$ is large, it will take exponential time to move from an independent set that lies mostly on the odd sublattice to one that is mostly even. Currently, the best known rigorous bounds verify that the Markov chain converges in polynomial time whenever $\lambda < 2.538$ [26] and requires exponential time when $\lambda > 5.365$ [7, 8].

This type of dichotomy is well known in the statistical physics community, where many models have been shown to abruptly transition from a disordered state to a predominantly ordered one. Physicists observe phase transitions when extending Gibbs distributions to infinite lattices and studying whether there is a unique limiting Gibbs measure, known as a Gibbs state (see, e.g., [14]). For the hard-core model on $\mathbb{Z}^2$, it is believed that there exists a critical value $\lambda_c$ such that for $\lambda < \lambda_c$ there is a unique Gibbs state, while for $\lambda > \lambda_c$ there are multiple Gibbs states. This has been verified for small and large values of $\lambda$ bounded away from the conjectured critical point $\lambda_c \approx 3.79$ in both the computational and physics settings [7, 26].

Fortunately, insights from statistical physics can also allow us to design alternative approaches to sampling in the slow regimes, in some cases. One approach that has proven fruitful far below the critical point (in the slow regime) is based on the cluster expansion [18]: at sufficiently low temperatures, configurations have long-range order, and can be precisely defined as small, randomized perturbations from some ground state, or highest probability state. Then configurations can be sampled by first randomly picking a ground state, and then inserting random defects with the appropriate conditional probabilities. A second approach uses simulated tempering or parallel tempering to sample at low temperatures by dynamically adjusting temperatures up and down during each simulation. These algorithms can be effective when we can (i) generate random samples from a family of temperatures so that low temperature configurations of interest arise often enough, and with the correct conditional probabilities, and (ii) the composite Markov chain on the larger state space (including configurations at all temperatures) converges quickly, even if it is prohibitively slow at low temperatures [6]. Finally, in some contexts it may be possible to rewrite the partition function as a sum over a different family of configurations, and this new representation may suggest alternative Markov chains that are quickly converging, even at low temperatures (see, e.g., [17, 28]).

## 3 Programmable Matter

Systems of programmable matter can be viewed as collections of simple interacting components with constant-size memory and limited computational capacity. We are interested in how these systems can be made to self-organize to produce emergent behaviors, such as coordination and collective movement.

Using a stochastic approach based on Markov chains, we can design rigorous and robust distributed algorithms for programmable matter exhibiting various desirable properties. For example, for the compression problem, our goal is to design an algorithm that allows an
interacting particle system to self-organize and gather together compactly. We say a connected
particle system on a planar lattice is $\alpha$-compressed if the perimeter of the ensemble is at
most $\alpha$ times the minimum perimeter possible for the $n$ particles, $p_{\min} = \Theta(\sqrt{n})$. In [12], we
gave a distributed, local Markov chain-based algorithm that solves the compression problem
for connected particle systems under the geometric amoebot model [13], a formal distributed
model in which particles move on the triangular lattice.

Our approach to this and other basic tasks proceeds as follows. We first choose a Hamilto-
nian $H(\sigma)$ over particle configurations that assigns lower values to preferable (compressed)
configurations. The transitions of the Markov chain are then defined to favor configurations
with small Hamiltonians. For compression, we let $H(\sigma) = -e(\sigma)$, where $e(\sigma)$ is the number
of edges induced by configuration $\sigma$, i.e., the number of lattice edges with both endpoints
occupied. Setting $\lambda = e^B$, we get $w(\sigma) = \lambda^{e(\sigma)}$. It is easy to verify that the number of
induced edges negatively correlated with the size of the perimeter, so the more induced edges,
the more compressed a configuration will be.

Using a Metropolis filter, we design a Markov chain $M$ that performs local moves and
converges to a distribution that generates configurations proportional to their weight $w(\sigma)$. In
particular, the probability of a configuration $\sigma$ is $w(\sigma)/Z$, where $Z = \sum_{\sigma} w(\sigma')$ is the
normalizing constant known as the partition function. Using tools from both statistical
physics and Markov chain analysis, we prove that, if we wait long enough, non-compressed
configurations occupy an exponentially small fraction of probability distribution when $\lambda$ is
sufficiently large.

The Markov chain $M$ for compression is defined as follows. Starting with an arbitrary
configuration $\sigma_0$ of $n$ simply connected particles, we define local rules that maintain con-
nectivity throughout the algorithm. There is a bias parameter $\lambda$ given as input, where
$\lambda > 1$ corresponds to particles preferring more neighbors and $\lambda < 1$ corresponds to particles
preferring fewer neighbors. The moves of the Markov chain $M$ are carefully designed so that the
particle system always remains simply connected, preventing the chain to disconnect or
form holes, which still keeping the state space connected via allowable transitions (so $M$ is
ergodic). Moreover, the moves are defined locally so that they can be implemented in a fully
distributed setting. Maintaining connectivity makes the analysis of the limiting distribution
simpler, but showing ergodicity is more challenging.

Particles individually execute a distributed algorithm defined by $M$, using Poisson clocks
to define when to attempt local moves. We prove that for all $\lambda > 2 + \sqrt{2}$, there is a constant
$\alpha = \alpha(\lambda) > 1$ such that at stationarity, with all but exponentially small probability, the
particle system will be $\alpha$-compressed. In fact, we show that for any $\alpha > 1$, there exists $\lambda$
such that our algorithms achieve $\alpha$-compression. Moreover, when $\lambda$ is small we achieve the
inverse property of expansion. For all $0 < \lambda < 2.17$, there is a constant $\beta < 1$ such that at
stationarity, with all but exponentially small probability, the perimeter will be $\beta$-expanded, i.e., the perimeter will be within a $\beta$ fraction of the maximum perimeter $p_{\text{max}} = \Theta(n)$. This implies that for any $0 < \lambda < 2.17$, the probability that the particle system is $\alpha$-compressed is exponentially small for any constant $\alpha > 1$.

The key ingredient used to establish compression and expansion is a careful Peierls
argument, used in statistical physics to study non-uniqueness of limiting Gibbs measures
and in computer science to establish slow mixing of Markov chains. Because we enforce
connectivity throughout the Markov process, our Peierls arguments are significantly simpler
than many standard arguments on configurations that are not required to be connected. In
subsequent work, we extended these results to the disconnected setting where, in contrast,
verifying ergodicity becomes trivial but analyzing the stationary distribution requires more
sophisticated tools [15].
One appeal of such a stochastic, distributed algorithm is its robustness. The system can recover from deviations in Poisson clocks waking particles to perform moves, anomalies in our individual particle’s movements, and even some particle failures. Moreover, this stochastic approach provides a general framework that is applicable beyond compression – it has the potential to solve any problem where the objective can be described in terms of minimizing some energy function, provided changes in that energy function can be calculated using only local information. One example is an optimization problem inspired by ant behavior [23] known as shortcut bridging where particles maintain bridge structures that balance a efficiency-cost tradeoff [1, 2]. A second example is a self-organizing system achieving separation, where particles of different colors can be shown to either intermingle or segregate depending on the settings of parameters [11]. Distributed algorithms based on Markov chains also have provided a theoretical explanation of phototaxing, or directed collective motion towards or away from a light source, in an experimental system of swarm robots [24]. Finally, we have promising directions for alignment and flocking, where oriented particles coordinate to determine a preferred direction of movement. In many of these cases, the collective behavior can be controlled by adjusting whether a physical system is in a disordered (gaseous) or an ordered (solid) state by exploring the physical properties of these systems.

References


