Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Designing Fast Absorbing Markov Chains

Stefano Ermon, Carla P. Gomes
Department of Computer Science
Cornell University, Ithaca, USA
{ermonste,gomes}@cs.cornell.edu

Ashish Sabharwal
IBM Watson Research Center
Yorktown Heights, NY, USA
ashish.sabharwal@us.ibm.com

Bart Selman
Department of Computer Science
Cornell University, Ithaca, USA
selman@cs.cornell.edu

Abstract

Markov Chains are a fundamental tool for the analysis of real world phenomena and randomized algorithms. Given a graph with some specified sink nodes and an initial probability distribution, we consider the problem of designing an absorbing Markov Chain that minimizes the time required to reach a sink node, by selecting transition probabilities subject to some natural regularity constraints. By exploiting the Markovian structure, we obtain closed form expressions for the objective function as well as its gradient, which can be thus evaluated efficiently without any simulation of the underlying process and fed to a gradient-based optimization package. For the special case of designing reversible Markov Chains, we show that global optimum can be efficiently computed by exploiting convexity. We demonstrate how our method can be used for the evaluation and design of local search methods tailored for certain domains.

Introduction

Markovian processes are an important modeling tool for the analysis of natural phenomena (for example, diffusion processes and Brownian motion), human made systems (e.g., telecommunication infrastructure and games), and also algorithms, most prominently methods for sampling from large state spaces (Madras 2002), graph-based learning (Page et al. 1999; Fouss et al. 2007; Wu et al. 2012) and local search techniques for combinatorial search and optimization (Aarts and Lenstra 2003; Hoos and Stützle 2004). Understanding statistical properties of such stochastic processes is of paramount importance, as they characterize the resulting behavior and often reflect performance metrics. For instance, mixing times are a key property of Markov Chain Monte Carlo methods (Jerrum and Sinclair 1997; Madras 2002), one of the most widely used tools to approximate #P problems which arise, e.g., in many probabilistic inference applications (Neal 1993). Since we can often interact with and influence the behavior of such processes, a natural question to consider is: *How can we optimally design such a system?*

Interestingly, the Markovian nature of these processes imposes a rich structure that can be exploited. For example, Boyd, Diaconis, and Xiao (2004) showed that, under certain conditions, the problem of designing a Markov Chain on a graph such that it has the *fastest mixing time* is convex, which allows it to be solved to optimality using standard optimization techniques.

Many real world processes, however, have a transient nature: for example, a game will eventually end and a local search algorithm (e.g., stochastic hill climbing) for a combinatorial search problem will eventually find a solution and stop. In such scenarios, we are typically interested in the final outcome, and the time required for the process to reach that outcome and terminate. In other words, the metric of interest is the *absorption time* of the chain, rather than its mixing time. Markov Chain theory has been extensively used to study such properties of specific, pre-defined processes. For example, in the context of local search, analytic results have been obtained for the probability of success (Papadimitriou 1991; Schoning 1999; 2007) and expected runtime (Zhou, He, and Nie 2009) of specific algorithms. However, to the best of our knowledge, there has been no attempt to formalize the general problem of designing in a principled way an optimal Markov Chain for a given transient processes.

We fill this gap by formalizing the problem of designing, given a state space, an initial probability distribution, and natural regularity constraints, a Markov Chain that minimizes the expected absorption time. By exploiting the Markovian structure, we obtain a closed form expression for the objective function—the expected absorption time—as well as its gradient, and can thus avoid the pitfalls of estimating it using Monte Carlo sampling methods. This allows us to use standard optimization methods from the literature to find locally optimal solutions. When restricting the design to the space of *reversible* Markov Chains, we prove that this optimization problem is convex and therefore a globally optimum can be efficiently found.

As an application of our methodology, we use it to understand the behavior of local search methods for certain domains. While our study is limited by the exponential growth of the underlying state space, we give an example of how insights used from this process can be used to enhance performance even on larger problems.

Markov Chains and Absorption Times

A discrete Markov chain (Grinstead and Snell 1997) M is a stochastic process defined on a finite set $\mathcal{X}$ of states. The process (or the "chain") starts in a state X^0 chosen from an initial probability distribution $P_0 \in [0,1]^{|\mathcal{X}|}$. If the chain is currently in state $x_i \in \mathcal{X}$, in the next step it moves to $x_j \in \mathcal{X}$ with a *transition probability* dependent not on the history of states visited previously but only on x_i and x_j and denoted by $p_{i\to j}$. Hence the process is Markovian. The *transition probability matrix* $P \in [0,1]^{|\mathcal{X}|\times|\mathcal{X}|}$ is defined by $P_{i,j} = p_{j\to i} = p_{ji}$. P is stochastic, i.e., $\sum_i P_{i,j} = 1$. For $k \geq 0$, we denote by X^k the state of the process at step k. X^k is distributed according to $P^k P_0$.

Definition 1. *A state $s \in \mathcal{X}$ is called **absorbing** if $p_{s\to s} = 1$, i.e. once entered is never left. A state $x \in \mathcal{X}$ is called **transient** if there is a non-zero probability of reaching an absorbing state from x (not necessarily in one step). M is called **absorbing** if it has at least one absorbing state and all its non-absorbing states are transient.*

Given an absorbing Markov Chain, let $S \subseteq \mathcal{X}$ be the set of absorbing states or "solutions". $\mathcal{X}$ is then the disjoint union $\mathcal{X} = S \sqcup \mathcal{T}$, where $\mathcal{T}$ is the set of transient states. We can write the transition matrix in canonical form[1]

$$P = \left(\begin{array}{c|c} Q & 0 \\ \hline R & I_{|S|} \end{array}\right) \tag{1}$$

by rearranging the order of the states so that the $|S|$ solutions appear last. Let $x_i, x_j \in \mathcal{T}$. By the structure of P, $\Pr[X^k = x_j \mid X^0 = x_i] = Q^k_{j,i}$, where $Q^k_{j,i}$ denotes the i,j-th entry of the k-th power matrix of Q. Hence, the expected number of times the chain visits state x_j during the first n steps given that it starts in state x_i is $\sum_{k=0}^{n} Q^k_{j,i}$. Let $\mathrm{E}[t_j \mid X^0 = x_i]$ be the expected number of visits to state x_j before the chain is absorbed, given that the chain starts in state x_i.

$$\mathrm{E}[t_j \mid X^0 = x_i] = \lim_{n\to\infty} \sum_{k=0}^{n} Q^k_{j,i}$$

This limit exists because the sequence is monotonically non-decreasing, but it might not be finite. The expected number of steps in state x_j before the chain is absorbed, given an initial probability distribution $\boldsymbol{\pi}_0 \in [0,1]^{|\mathcal{T}|}$, is therefore

$$\mathrm{E}[t_j] = \sum_{x_i \in \mathcal{T}} \mathrm{E}[t_j \mid X^0 = x_i]\pi_0(x_i) = \sum_{x_i} \sum_{k=0}^{\infty} Q^k_{j,i}\pi_0(x_i)$$

The expected number of steps before the chain is absorbed (i.e, the expected time spent in $\mathcal{T}$) is thus

$$\sum_{x_j \in \mathcal{T}} \mathrm{E}[t_j] = \sum_{x_i, x_j \in \mathcal{T}} \sum_{k=0}^{\infty} Q^k_{j,i} P_0(x_i) = \mathbf{1}^T \left(\sum_{k=0}^{\infty} Q^k\right) \boldsymbol{\pi}_0$$

It is known that this infinite sum can be evaluated by matrix inversion:

[1] Q is not necessarily irreducible, as there could be more than one transient class.

Lemma 1 ((Grinstead and Snell 1997; Meyer 2000)). *For an absorbing Markov chain M defined by eqn. (1),*

1. *$Q^n \to \mathbf{0}$;*
2. *$I - Q$ is invertible; and*
3. *$N = (I-Q)^{-1} = \sum_{k=0}^{\infty} Q^k$.*

This allows us to evaluate the *expected absorption time* of M, denoted $\mathrm{E}[t_{\text{absorb}}(M)]$, without having to actually simulate the process or compute an infinite number of power matrices:

$$\mathrm{E}[t_{\text{absorb}}(M)] = f(P, \boldsymbol{\pi}_0) = \mathbf{1}^T (I-Q)^{-1} \boldsymbol{\pi}_0 \tag{2}$$

As a special case, let $\boldsymbol{\pi}_u$ be the uniform distribution over $\mathcal{T}$. Then, $f(P, \boldsymbol{\pi}_u) = \frac{1}{|\mathcal{T}|} \mathbf{1}^T (I-Q)^{-1} \mathbf{1}$.

Note that we need not explicitly invert $(I-Q)$. The expected absorption time can be calculated more efficiently in practice as $\mathbf{1}^T \mathbf{x}$ where $\mathbf{x}$ is the solution of the system of linear equations $(I-Q)\mathbf{x} = \boldsymbol{\pi}_0$. Further, we obtain a closed form expression for the gradient of the expected absorption time $\nabla \mathrm{E}[t_{\text{absorb}}(M)]$ from the identity $\frac{\partial \mathbf{Y}^{-1}}{\partial x} = -\mathbf{Y}^{-1} \frac{\partial \mathbf{Y}}{\partial x} \mathbf{Y}^{-1}$ (Petersen and Pedersen 2008). Even in this case, explicit matrix inversion is not required.

Optimizing Absorption Time

Let $\mathbb{I}$ denote the set of all closed intervals within $[0,1]$.

Definition 2 (Constrained Fastest Absorbing Markov Chain Problem). *Given a set $\mathcal{X} = S \sqcup \mathcal{T}$ of states with $S \subseteq \mathcal{X}$ as the absorbing states, intervals $\{I\}_{ij} \in \mathbb{I}^{|\mathcal{X}|\times|\mathcal{X}|}$, equivalence classes given by $\mathbb{D} \subseteq \mathcal{T}^4$, and an initial distribution $\boldsymbol{\pi}_0$ over $\mathcal{T}$, design a Markov chain M, as defined by a transition matrix P structured as in eqn. (1), such that the expected absorption time of M is minimized and P satisfies non-negativity, stochasticity, interval, and equivalence constraints:*

$$\begin{aligned}
\text{minimize} \quad & \mathbf{1}^T (I-Q)^{-1} \boldsymbol{\pi}_0 && (3)\\
\text{subject to} \quad & 0 \leq P \leq 1 \ , \ \mathbf{1}^T P = \mathbf{1}^T && (4)\\
& p_{ij} \in I_{ij}, \forall i,j \in \{1,\ldots,|\mathcal{X}|\} && (5)\\
& p_{ij} = p_{i'j'}, \forall (ij, i'j') \in \mathbb{D} && (6)
\end{aligned}$$

If constraints (5) and (6) are not imposed, the optimal solution is to simply go from every transient state to an absorbing state in one step, i.e., setting $Q = 0$ and fixing R arbitrarily so as to satisfy (4). This is, of course, not very interesting. We thus work under the assumption that is it not always possible to jump from any transient state always to an absorbing state, i.e., Q may not be 0. This is the case in several Markov chains defined on graphs which impose a neighborhood structure and p_{ij} must be 0 whenever i is not a neighbor of j, a restriction specified by constraint (5). For local search algorithms in discrete domains, the neighborhood may be defined by the Hamming metric over configurations. In wildlife conservation, nodes represent geographic locations which impose a natural neighborhood structure. We will see examples of these later in the paper.

Even when a neighborhood structure imposes $p_{ij} = 0$ for all non-neighbors, the optimal solution is to define a *deterministic* Q that corresponds to a network of shortest paths from all transient states to absorbing states. In the case of local search algorithms over combinatorial spaces, however, the number of states is exponential and the Markov chain must be represented implicitly in a succinct way, thereby eliminating the possibility of simply using shortest paths. We formalize this restriction as (6). Again, $\mathbb{D}$ itself must have succinct representation to be useful. In our use case of optimizing local search, this can be done by defining certain "features" ϕ associated with pairs (x_i, x_j) of states such that the feature space is much smaller than the state space and the transition probabilities are completely characterized by the features. We will discuss this in detail later.

Optimization algorithm

We propose to solve the optimization problem in Definition 2 using the L-BFGS-B algorithm proposed by Byrd et al. (1995), which extends L-BFGS to handle box constraints (5) on the variables. The algorithm is designed for large scale optimization problems and it is based on the gradient projection method, with a limited memory BFGS matrix used to approximate the Hessian of the objective function. The gradient of the objective function is evaluated at each step using the closed form derived earlier. This choice is motivated by the fact that the optimization problem in Definition 2 is in general non-convex, as we will discuss shortly. This also means that L-BFGS, although likely to produce good solutions, will generally not provide any global optimality guarantees.

Efficiently Designing Globally Optimal Chains

We explore conditions that guarantee an efficient design of globally optimal Markov Chains. Let $\mathbb{W}$ be the set of transition matrices of absorbing Markov Chains with state space $\mathcal{X} = S \sqcup \mathcal{T}$.

Proposition 1. *The set $\mathbb{W}$ is convex.*

Proof. Let $P_1, P_2 \in \mathbb{W}$. Consider $P = \lambda P_1 + (1-\lambda)P_2$ for $\lambda \in [0,1]$. Then P is stochastic. Further, from any state $i \in \mathcal{T}$, there must exist paths ℓ_1 and ℓ_2 of length at most $|\mathcal{T}|$ in the two chains, respectively, to the sinks S and with non-zero probabilities p_1 and p_2, resp., of being taken. The probability of eventually reaching S from i under P is therefore at least $\max\{p_1\lambda^{|\mathcal{T}|}, p_2(1-\lambda)^{|\mathcal{T}|}\}$, which is strictly positive. Hence, P is also absorbing, proving convexity of $\mathbb{W}$. □

The objective function $f(P, \boldsymbol{\pi}_0)$ in (2) is, however, in general not convex over $\mathbb{W}$. As an example, consider P_1 such that its Q submatrix in decomposition (1) is $Q_1 = [0, 1; 0, 0]$, and P_2 such that $Q_2 = [0, 0; 1, 0]$. Then, $f(P_1, \boldsymbol{\pi}_u) = f(P_2, \boldsymbol{\pi}_u) = 3/2$, but $f(\frac{1}{2}P_1 + \frac{1}{2}P_2, \boldsymbol{\pi}_u) = 4/2 > 3/2$.

Interestingly, if we restrict the focus on the design of *reversible* (relative to Q) Markov Chains, convexity does hold, as we show below. The complete Markov Chain, represented by P, clearly cannot be reversible because of the sinks. However, the part relative to Q can still be. For example, Simulated Annealing at fixed temperature forms a reversible chain over the transient states.

Theorem 1. *Let $\boldsymbol{\pi} > 0$ and $\mathbb{S}_{\boldsymbol{\pi}} \subseteq \mathbb{W}$ be the set of stochastic matrices defining absorbing Markov chains with a finite state space $\mathcal{X} = S \sqcup \mathcal{T}$ such that $\pi_i q_{ji} = \pi_j q_{ij}$, for all transient states $i, j \in \mathcal{T}$. Then $\mathbb{S}_{\boldsymbol{\pi}}$ is convex, $f(P, \boldsymbol{\pi}) = \mathbf{1}^T (I - Q)^{-1} \boldsymbol{\pi}$ is convex over $\mathbb{S}_{\boldsymbol{\pi}}$, and the optimization problem in Definition 2 can be solved efficiently under the additional constraint $P \in \mathbb{S}_{\boldsymbol{\pi}}$.*

Proof. Letting $\Pi = diag(\boldsymbol{\pi})$, the assumption in the theorem can be compactly written as $\Pi Q^T = Q\Pi$ which implies that $A \triangleq \Pi^{-1/2} Q \Pi^{1/2}$ is symmetric. Further, $A^k = \Pi^{1/2} Q^k \Pi^{-1/2}$, and so

$$\begin{aligned}(I-Q)^{-1} = \sum_{k=0}^{\infty} Q^k &= \Pi^{1/2} (\sum_{k=0}^{\infty} A^k) \Pi^{-1/2} \\ &= \Pi^{1/2} (I - A)^{-1} \Pi^{-1/2}\end{aligned}$$

Using the fact that $\boldsymbol{\pi} = \Pi \mathbf{1}$, we have

$$\begin{aligned}f(P, \boldsymbol{\pi}) &= \mathbf{1}^T \Pi^{1/2} (I-A)^{-1} \Pi^{-1/2} \boldsymbol{\pi} \\ &= \mathbf{1}^T \Pi^{1/2} (I-A)^{-1} \Pi^{1/2} \mathbf{1} \\ &= h(\Pi^{1/2}\mathbf{1}, (I-A))\end{aligned}$$

where $h(x, Y) = x^T Y^{-1} x$ is convex over the domain $\mathbb{R}^n \times S^n_{++}$ (the set of symmetric positive definite matrices) (Boyd and Vandenberghe 2004). Note that A has the same eigenvalues as Q because these matrices are similar. Since the Markov Chain defined by P is absorbing, its eigenvalue with the largest modulus has modulus (i.e., spectral radius) $\rho(Q) < 1$. Thus, we also have $\rho(A) < 1$, implying that the eigenvalues of $(I - A)$ are positive, which in turn implies that $(I - A)$ is positive definite. Using the convexity of h and observing that $(I - A)$ is an affine transformation of P proves the result.

Since $\mathbb{W}$ is convex and $\pi_i q_{ji} = \pi_j q_{ij}$ is a linear constraint, $\mathbb{S}_{\boldsymbol{\pi}} \subseteq \mathbb{W}$ is also convex. Since constraints (4) and (5) are also linear, the optimization problem in Definition 2 is convex and can be solved efficiently under the additional constraint $P \in \mathbb{S}_{\boldsymbol{\pi}}$. □

In particular, when $\boldsymbol{\pi} = \boldsymbol{\pi}_u$, reversibility simply means that the transition matrix P is symmetric (relative to Q). Hence, globally optimal symmetric Markov chains can also be designed efficiently.

Enhancing Local Search Methods

A Stochastic Local Search method for a discrete decision or optimization problem may be viewed as a "recipe" for mapping a problem instance to a transition probability matrix P, which defines a finite state Markov Chain over the entire search space (e.g., one state for each possible assignment to the variables), and then simulating the chain until it reaches a solution.[2] The resulting algorithm can therefore be seen as

[2]Time-varying processes, such as Simulated Annealing with a changing temperature or "restarts" in local search methods, may also be modeled this way with an appropriately extended P.

simulating an absorbing Markov Chain, where the absorbing states are the solutions. Its expected runtime is thus simply the expected absorption time of this chain.

Since the search space is typically exponentially large, the transition probability matrix is often defined only implicitly. Pairs of source-destination states are mapped to *features* of interest, such as a fitness function (e.g., number of constraints violated by each configuration) in the case of a stochastic hill climbing algorithm. The transition probability for that state pair is then defined as a succinctly represented function of these features. E.g., the algorithm may always accept downhill moves and accept uphill moves with a certain probability.

While the feature engineering part is often problem specific and designed by domain experts, how these features are combined in the algorithm to produce transition probabilities opens up a rich and interesting space for the analysis and improvement of such algorithms. Given a stochastic local search algorithm with a set of features it uses, does it combine the information from these features in a way that optimizes its expected runtime? If not, can we improve upon it? We next explore how our technique can address these questions. Although our method is clearly limited by the exponential growth of the state space for combinatorial search problems, we will show that insights obtained on small problems can be applied to larger scale problems as well.

Formally, let $\mathcal{X}^{\mathcal{I}}$ be the set of all configurations (i.e., variable assignments) for an instance $\mathcal{I}$ of a combinatorial search problem $\mathcal{P}$ (e.g, SATisfiability testing (Biere 2009)). Let $\mathbb{S}_{|\mathcal{X}^{\mathcal{I}}|} \subset \mathbb{R}^{|\mathcal{X}^{\mathcal{I}}| \times |\mathcal{X}^{\mathcal{I}}|}$ denote the set of all stochastic matrices of order $|\mathcal{X}^{\mathcal{I}}|$. Every local search method provides a function $g : \mathcal{P} \to \mathbb{S}_{|\mathcal{X}^{\mathcal{I}}|}$ which is a "recipe" to associate with each instance $\mathcal{I}$ of $\mathcal{P}$ a transition probability matrix $g(\mathcal{I})$.

Typically, the i, j entry of $g(\mathcal{I})$ depends only on a small set of *features* of the corresponding states (configurations) x_i, x_j. Such features could be the number of constraints they violate (called *energy*) and their Hamming distance. Let $\phi : \mathcal{X}^{\mathcal{I}} \times \mathcal{X}^{\mathcal{I}} \to \mathcal{F}$ be a function mapping configuration-pairs to a feature vector. We consider a class of stochastic local search methods whose transition probability matrix P has entries $p_{ij} = h(\phi(x_i, x_j))$ that depend only on some well-defined set of features. For example, consider a Metropolis algorithm for a Boltzmann distribution, i.e. a fixed "temperature" Simulated Annealing algorithm (SA) (Kirkpatrick, Gelatt Jr., and Vecchi 1983), for a Constraint Satisfaction problem (CSP) (Russell et al. 2010). Here, the only features considered are the Hamming distance $d_H(x_i, x_j)$ and the difference in the number of violated constraints $E_i - E_j$, and transition probabilities are:

$$p_{ij} = \frac{1}{n} \min \left\{ 1, \exp\left(\frac{E_i - E_j}{T} \right) \right\} 1_{d_H(x_i,x_j)=1} \quad (7)$$

where T is a formal parameter called "temperature". A greedy algorithm might include as a feature the lowest energy among the configurations in some neighborhood of x_i, and would set the transition probability to zero unless the destination configuration has minimal energy in the neighborhood. For structured problems like SAT, features typically also include information based on the constraint structure (e.g., the so-called makecount and breakcount of each variable (Selman, Kautz, and Cohen 1993)).

These features can be thought of as defining *equivalence classes* among all possible transitions, because transitions with the same features will be given the same transition probabilities. This is the key property that allows local search algorithms to specify in a succint way an exponentially large number of transition probabilities. More formally, there exists an equivalence classes given by $\mathbb{D} \subseteq \mathcal{X}^{\mathcal{I}^4}$ such that $p_{ij} = p_{i'j'}, \forall (ij, i'j') \in \mathbb{D}$. Notice this limitation imposed by the succint representation is given by (6) in the CONSTRAINED FASTEST ABSORBING MARKOV CHAIN PROBLEM definition.

Parameterized Local Search: Given this insight, it is natural to consider a more general class of **parameterized** local search methods $g' : \Theta \times \mathcal{P} \to \mathbb{S}_{|\mathcal{X}^{\mathcal{I}}|}$ where g' is a function mapping parameters $\theta \in \Theta$ and instances to transition probability matrices. Each parameter can be thought of as a possible way of combining the information contained in the features. For example, if we use as features the energies E_i, E_j of the source and destination states, we could define a very general version of SA where

$$p_{ij} = \frac{1}{n} \theta_{E_i,E_j} 1_{d_H(x_i,x_j)=1} \quad (8)$$

i.e., there is one parameter for each possible energy transition (E_i, E_j), not restricted to have the specific form given in (7).

Given this flexibility and a defined set of features, it is natural to seek a parameter $\theta \in \Theta$ that optimizes the algorithm's performance, measured as the expected time required to find a solution (assuming the instance has a solution), which is the expected absorption time, as defined in (2), of the Markov Chain defined by $g'(\theta, \mathcal{I})$. This problem in turn can be casted as a CONSTRAINED FASTEST ABSORBING MARKOV CHAIN PROBLEM, where constraint (6) enforces the restriction on the transition probabilities given by the *equivalence classes* associated with the features considered by a certain algorithm. In other words, this guarantees that the resulting Markov Chain can be specified in a succint way. Constraint (5) is used to directly enforce the limitations imposed by underlying metric, i.e. the local search algorithm is only allowed local moves in a neighborhood of a certain size (in our experiments, a Hamming ball of radius 1). Specifically, we can use the L-BFGS-B based optimization algorithm described earlier to find the parameter θ that optimizes the absorption time, subject to the additional constraints we imposed.

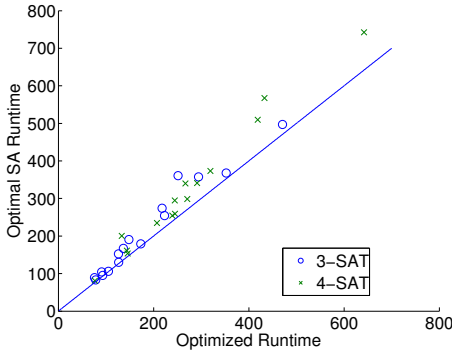
Optimizing Local Search Performance

We consider a local search scheme for a binary CSP defined over n variables and with m constraints. The state space $\mathcal{X} = \{0, 1\}^n$ corresponds to the 2^n possible configurations or variable-value assignments, and solutions $S \subseteq \mathcal{X}$ are configurations that satisfy all the constraints. We define

the energy E_i of a configuration i as the number of constraints of the CSP violated by i. For this study, we focus on a parameterized local search scheme based on energy as the only feature (and only allowing moves of up to Hamming distance 1). Specifically, we have a parameter $\theta \in [0,1]^{m\times m}$. The transition probabilities are given by (8) for $i \neq j$, and $p_{ii} = 1 - \sum_{j'} p_{ij'}$. This corresponds to a sampling scheme where from state i (with energy E_i) we select uniformly at random a neighbor j' at Hamming distance 1, and accept the transition to j' with probability $\theta_{E_i,E_{j'}}$. This corresponds to a fixed temperature SA if θ_{E_i,E_j} is set to $\exp((E_i - E_j)/T)$. Our formulation, however, allows for much more flexibility. We explore whether the expected runtime of the algorithm can be improved by altering the way energy feature is utilized when computing transition probabilities. Specifically, given a CSP instance we use the L-BFGS-B based optimization algorithm described earlier to solve the corresponding CONSTRAINED FASTEST ABSORBING MARKOV CHAIN PROBLEM, i.e. to look for the optimal parameter $\theta \in [0,1]^{m\times m}$ that optimizes the expected absorption time, subject to the additional constraints we imposed.

Random k-SAT Ensemble: To evaluate the above approach, we generated 250 satisfiable instances from the random k-SAT ensemble (Selman, Mitchell, and Levesque 1996) for $k = 3, 4$. For each individual instance, we optimized the expected runtime (2) for a uniform initial distribution $\boldsymbol{\pi}_u$ as a function of the parameter $\theta \in [0,1]^{m\times m}$.When we compare with the expected absorption time achieved by setting optimally the only parameter T in the transition probabilities defined as in (7), we get improvements ranging from 18% to 49% for 3-SAT instances, and ranging from 18% to 81% for the harder 4-SAT instances (optimizing over MC with 2^7 to 2^{13} states). A scatter plot for the runtime improvement for several instances is reported in Figure 2 (left). We see that the additional flexibility allowed by having a different acceptance probability per energy-pair allows us to greatly outperform SA. Notice that this comparison is fair because the two schemes use the same set of features, Hamming distance limitations, and neighbor selection scheme. The improvement can thus be attributed entirely to a more effective way of exploring the search space.

As a second experiment, we used our method to optimize the parameter θ for the expected absorption time *jointly* for an ensemble of random k-SAT problems, seeking a *single* parameter that performs well on average across the entire family of instances. In this case, we found that the average performance of the best parameter θ^* is essentially identical to that one of a SA scheme tuned with the best temperature parameter T. For instance, on 3-SAT problems with 10 variables and 42 clauses we get 106.4 vs 107.9 of the best SA. On 5-SAT with 10 variables and 212 clauses, we get 305.2 vs 307.6 of SA. An optimal parameter θ^* represented as heat-map is shown in the left panel of Figure 1. Notice that it is quite similar to SA, and differences mostly occur for transitions that do not frequently happen in the data, and thus don't affect the result much. These results suggest that the strategy employed by SA is actually good (when the temperature is tuned well), at least for these small problems, and for the average ensemble one can't actually improve much if the only feature we have is the energy (clearly, with more features one can do much better than SA for random k-SAT).



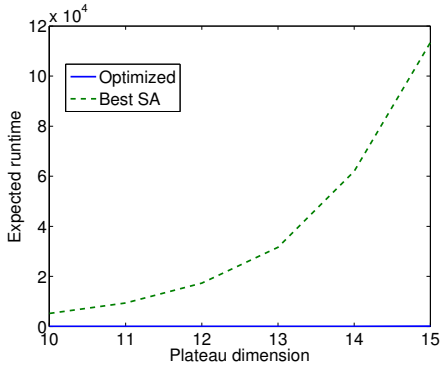


Figure 2: Runtime improvements. Top: random instances. Bottom: hard energy landscape instances.

Hard energy landscapes: We next consider an ensemble defining a challenging energy landscape. These are SAT problems over binary variables $x_1, y_1, \cdots, y_b, r_1, \cdots, r_p$ with constraints

$$(x_1 \Rightarrow y_1) \wedge (x_1 \Rightarrow y_2) \wedge \cdots \wedge (x_1 \Rightarrow y_b) \wedge$$
$$(\neg x_1 \Rightarrow \neg y_1) \wedge (\neg x_1 \Rightarrow \neg y_2) \wedge \cdots \wedge (\neg x_1 \Rightarrow \neg y_b))$$
$$\wedge (x_1 \vee r_1) \wedge \cdots \wedge (x_1 \vee r_p) \wedge (\neg x_1 \vee z_1) \wedge (\neg x_1 \vee \neg z_1)$$

The effect of the first line of constraints is to create an "energy barrier" between $(x_1, y_1, \cdots, y_b) = (0 \ldots 0)$ and $(x_1, y_1, \cdots, y_b) = (1 \cdots 1)$. In order to go from one to the other by flipping one variable at a time, one has to violate at least $b/2$ constraints. The last line creates a large "energy plateau" corresponding to assignments with $x_1 = 1$ that are close to solutions because they satisfy all

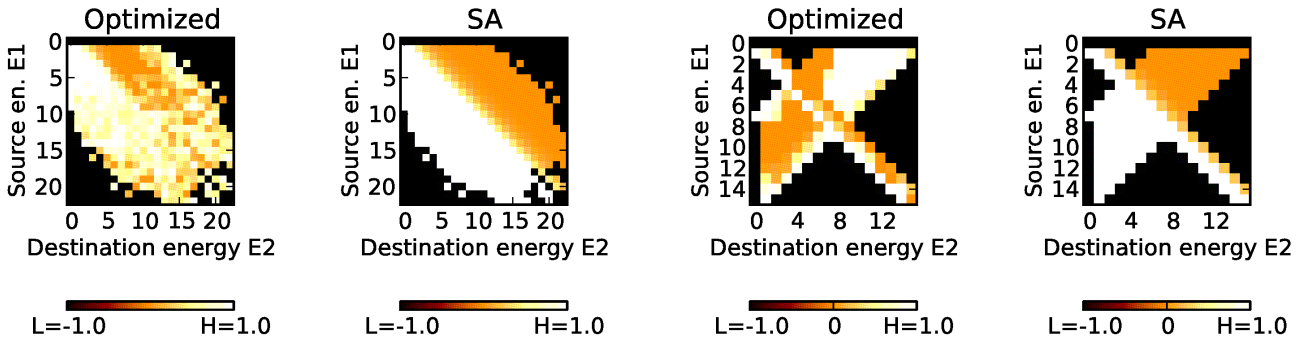


Figure 1: Acceptance probabilities for transitions from energy E_1 to E_2 for optimized local search vs. Simulated Annealing. Black indicates energy-pair was not used. Left: 5-SAT with 10 variables, trained on 10000 instances. Right: hard energy landscape instance with $p = 10, b = 6$.

the $(x_1 \vee r_1) \wedge \cdots \wedge (x_1 \vee r_p)$ constraints (but all solutions have $x_1 = 0$).Intuitively, this instance is difficult for SA because in order to reach one of the solutions, SA needs to set $x_1 = 0$. Setting $x_1 = 0$ is likely to violate several of the first b clauses, making the uphill move unlikely to be accepted.

For this experiment, we optimized $\theta \in [0, 1]^{m \times m}$ for an instance with $b = 6, p = 8$. The optimal parameter θ^* is shown in the right panel of Figure 1 as a heatmap, and compared with the corresponding acceptance probabilities computed according to Eq. (7) for the optimal parameter T. Note that the two schemes look extremely different. E.g., according to θ^* we are accepting uphill moves with a much higher probability. We then run the chain on other instances generated with different values of p, which controls the size of the plateau. When we encounter a previously unseen parameter in (8) (black in the heatmap), we set it to a random value from $[0, 1]$.

As shown in Figure 2 (bottom), there is a dramatic improvement in performance when using the algorithm defined by θ^* as opposed to SA, whose time complexity, measured here as the number of steps in the Markov Chain, increases exponentially with p. While the former often takes only of the order of 1,000 steps irrespective of the plateau dimension (the line labeled "Optimzied" stays barely above the horizontal axis), SA quickly increases to over 100,000 steps as p is increased to 15.

Generalization to Larger Instances: Perhaps most interestingly, *the parameters learned from smaller instances generalize well to larger ones*. For example, using the parameters learned for the rather small instance with $b = 6, p = 8$, instances with a much larger plateau dimension such as $b = 6, p = 45$ are solved easily in 22,898 steps. The learned parameters also generalize in terms of the barrier size b, up to $b = 40, p = 8$ with an expected runtime of 28,571 steps, and also to instances with both a larger barrier size and a larger plateau size (e.g., $p = 20, b = 20$ is solved on average in 7931 steps). These results demonstrate that the optimized parameters can generalize to new, previously unseen instances of a larger size, which are all well out of reach of SA.

Conclusion

We introduced and formalized the problem of designing a Markov Chain that minimizes the expected absorption time. Our method allows efficiently finding locally optimal solutions thanks to a closed form expression for the objective function as well as its gradient, which can thus be evaluated without simulation. The design of an optimal reversible chain turns out to be a convex optimization problem and thus solvable optimally for reasonably sized state spaces. As an application, we used our method to optimize transition probabilities in local search algorithms.

Acknowledgments

This research is funded by NSF Expeditions in Computing grant #0832782 and Computing Research Infrastructure grant #1059284.

References

Aarts, E., and Lenstra, J. 2003. *Local search in combinatorial optimization*. Princeton University Press.

Biere, A. 2009. *Handbook of satisfiability*, volume 185. Ios Press-Inc.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Boyd, S.; Diaconis, P.; and Xiao, L. 2004. Fastest mixing Markov chain on a graph. *SIAM review* 46(4):667–689.

Byrd, R.; Lu, P.; Nocedal, J.; and Zhu, C. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16(5):1190–1208.

Fouss, F.; Pirotte, A.; Renders, J.-M.; and Saerens, M. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *Knowledge and Data Engineering, IEEE Transactions on* 19(3):355–369.

Grinstead, C., and Snell, J. 1997. *Introduction to probability*. Amer Mathematical Society.

Hoos, H., and Stützle, T. 2004. *Stochastic local search: Foundations & applications*. Morgan Kaufmann.

Jerrum, M., and Sinclair, A. 1997. The Markov chain Monte Carlo method: an approach to approximate counting and integration. *Approximation algorithms for NP-hard problems* 482–520.

Kirkpatrick, S.; Gelatt Jr., D.; and Vecchi, M. 1983. Optimization by simmulated annealing. *science* 220(4598):671–680.

Madras, N. 2002. *Lectures on Monte Carlo Methods*. American Mathematical Society.

Meyer, C. 2000. *Matrix analysis and applied linear algebral*. Society for Industrial Mathematics.

Neal, R. 1993. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank citation ranking: bringing order to the web. Technical report, Stanford InfoLab.

Papadimitriou, C. 1991. On selecting a satisfying truth assignment. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, 163–169. IEEE.

Petersen, K., and Pedersen, M. 2008. The matrix cookbook. *Technical University of Denmark*.

Russell, S.; Norvig, P.; Davis, E.; Russell, S.; and Russell, S. 2010. *Artificial intelligence: a modern approach*. Prentice hall Upper Saddle River, NJ.

Schoning, T. 1999. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, 410–414. IEEE.

Schoning, U. 2007. Principles of stochastic local search. In *Unconventional Computation*, volume 4618 of *Lecture Notes in Computer Science*, 178–187.

Selman, B.; Kautz, H.; and Cohen, B. 1993. Local search strategies for satisfiability testing. *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge* 26:521–532.

Selman, B.; Mitchell, D.; and Levesque, H. 1996. Generating hard satisfiability problems. *Artificial intelligence* 81(1):17–29.

Wu, X.-M.; Li, Z.; So, A. M.-C.; Wright, J.; and Chang, S.-F. 2012. Learning with partially absorbing random walks. In *Advances in Neural Information Processing Systems 25*, 3086–3094.

Zhou, Y.; He, J.; and Nie, Q. 2009. A comparative runtime analysis of heuristic algorithms for satisfiability problems. *Artificial intelligence* 173(2):240–257.