

Time Lower Bounds for the Metropolis Process and Simulated Annealing

Zongchen Chen ✉ 🏠 


Georgia Institute of Technology, Atlanta, GA, USA

Dan Mikulincer ✉ 🏠 

University of Washington, Seattle, WA, USA

Daniel Reichman ✉ 🏠 

Worcester Polytechnic Institute, MA, USA

Alexander S. Wein ✉ 🏠 

University of California, Davis, CA, USA

Abstract

The Metropolis process (MP) and Simulated Annealing (SA) are stochastic local search heuristics that are often used in solving combinatorial optimization problems. Despite significant interest, there are very few theoretical results regarding the quality of approximation obtained by MP and SA (with polynomially many iterations) for NP-hard optimization problems.

We provide rigorous lower bounds for MP and SA with respect to the classical maximum independent set problem when the algorithms are initialized from the empty set. We establish the existence of a family of graphs for which both MP and SA fail to find approximate solutions in polynomial time. More specifically, we show that for any $\varepsilon \in (0, 1)$ there are n -vertex graphs for which the probability SA (when limited to polynomially many iterations) will approximate the optimal solution within ratio $\Omega\left(\frac{1}{n^{1-\varepsilon}}\right)$ is exponentially small. Our lower bounds extend to graphs of constant average degree d , illustrating the failure of MP to achieve an approximation ratio of $\Omega\left(\frac{\log(d)}{d}\right)$ in polynomial time. In some cases, our lower bounds apply even when the temperature is chosen adaptively. Finally, we prove exponential-time lower bounds when the inputs to these algorithms are bipartite graphs, and even trees, which are known to admit polynomial-time algorithms for the independent set problem.

2012 ACM Subject Classification Theory of computation \rightarrow Complexity classes

Keywords and phrases Metropolis Process, Simulated Annealing, Independent Set

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2025.47

Category RANDOM

Related Version *Full Version:* <https://arxiv.org/abs/2312.13554> [11]

Funding *Dan Mikulincer:* DM was partially supported by the Brian and Tiffnie Pang Faculty Fellowship.

Alexander S. Wein: ASW was partially supported by an Alfred P. Sloan Research Fellowship.

Acknowledgements We would like to thank Eric Vigoda for his useful feedback regarding this work.

1 Introduction

Simulated Annealing [42, 60] (SA) is a family of randomized local search heuristics, widely applicable for combinatorial optimization problems. In the maximization version we are given a finite search space \mathcal{C} of feasible solutions, and a cost $f(x)$, for every solution $x \in \mathcal{C}$. Additionally, for every solution $x \in \mathcal{C}$ there is a set $N(x) \subseteq \mathcal{C}$ of *neighboring solutions* accessible via a single move of the search algorithm. In contrast to hill-climbing methods that consistently choose an element $y \in N(x)$ with $f(y) \geq f(x)$, SA may choose a neighboring



© Zongchen Chen, Dan Mikulincer, Daniel Reichman, and Alexander S. Wein;
licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025).

Editors: Alina Ene and Eshan Chattopadhyay; Article No. 47; pp. 47:1–47:22



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

solution y satisfying $f(y) < f(x)$ with probability $e^{-\frac{\Delta}{T}}$ where $\Delta := f(x) - f(y)$ and $T > 0$ is a *temperature parameter*, that governs the behavior of the algorithm. Typically one gradually reduces the temperature over time using a predetermined cooling schedule allowing for a more exploratory algorithm in the early stages. The idea is that since the algorithm is allowed to accept downhill moves it should be able to escape local maxima and find (hopefully in polynomial time) near-optimal solutions. The case where the temperature T is fixed throughout the algorithm has received attention as well [51]: in this case, the algorithm is called the Metropolis process (MP).

Since its inception in the 1980s SA was found empirically to be highly effective for numerous optimization problems in diverse fields such as VLSI design, pattern recognition, and quantum computing. The great popularity of SA is acknowledged in several dedicated books, articles, surveys, and textbooks concerned with algorithm design [43, 14, 1, 4, 39, 40].

Considering the wide applicability of SA for optimization problems, one may wonder what rigorous results can be obtained regarding the algorithm's performance. It is well-known [26] that for a suitable cooling schedule, SA, if run for sufficiently many iterations, will almost surely converge to a global optimizer. However, there is no guarantee that the running time will be polynomial in the size of the input. This begs the question of what can be said with respect to SA when it is constrained to run for *polynomially many steps*. This question is explicitly mentioned as challenging in several papers [2, 4, 25, 37]. For example, it is mentioned in [2] that "The polynomial time behavior of simulated annealing is notoriously hard to analyze rigorously". In the field of approximation algorithms for NP-hard optimization problems not much seems to be known with respect to upper and lower bounds regarding the *approximation factor* that can be achieved efficiently with SA. The situation for MP is similar: Little is known about the approximation ratio achievable by MP (when run for polynomially many steps) for NP-hard optimization problems. As stated by [37], "Rigorous results...about the performance of the Metropolis algorithm on non-trivial optimization problems are few and far between". Despite some recent developments [13, 12, 47], the literature on rigorous results for MP and SA remains sparse and experts have noticed the "gap between theory and practice...for Simulated Annealing" [15].

The lack of runtime complexity lower bounds for SA and MP is not a coincidence, since some natural approaches run into difficulties. One direction to prove time lower bounds for MP and SA is to rely on known bounds for the mixing time of the relevant Markov chains. For such bounds, there is a wealth of existing established techniques [46, 52, 20, 9, 16, 6, 54]. In particular, it is well known that for a fixed fugacity parameter λ , the Metropolis process for independent sets converges to a stationary distribution known as the hardcore model. This distribution assigns to each independent set I the value $\lambda^{|I|}/Z$ where Z is the normalizing constant. There are numerous lower bounds for the mixing time of Markov chains converging to the hardcore model [22, 23, 55, 58, 54]. However, slow mixing does not necessarily imply anything about the efficiency of MP as an approximation algorithm. For example, a simple conductance argument shows that MP has exponential mixing time, with any temperature parameter, when searching for the maximum independent set in the complete bipartite graph $K_{n,n}$. On the other hand, it is easily seen that with an appropriate temperature, MP will find an optimal independent set in $K_{n,n}$ in polynomial time. Furthermore, lower bounds on mixing times imply the existence of a "bad" initial state from which the expected time of the chain to mix is super-polynomial. These kinds of statements do not usually carry information about initialization from specific states, as done in practice. For example, as observed in [35] and further elaborated in [12], conductance lower bounds on the mixing time do not imply comparable lower bounds on the time it takes MP for the independent

set problem to find an optimal or even near-optimal solution when using the natural empty state initialization. Finally, as noted in [35], common techniques to prove mixing times lower bounds for homogeneous Markov chains do not generalize in a straightforward way to inhomogeneous chains such as SA.

A brief summary of our results

The overarching aim of this paper is to advance our understanding of the theoretical guarantees afforded by the above-mentioned algorithms and investigate the possible limitations and hard instances. Specifically, we focus on the *maximum independent set* problem. Recall that given a graph G , an independent set in G is a subset of vertices that spans no edge. The cardinality of a maximum independent set in G is denoted by $\alpha(G)$. Computing $\alpha(G)$ exactly or approximately is a classical NP-hard problem. We establish super polynomial lower bounds on the number of iterations required by MP or SA to obtain a reasonable approximation for $\alpha(G)$ for several different families of graphs: dense graphs, graphs of bounded average degree, bipartite graphs, and trees (for more details please see Section 3). For each such class, we prove corresponding lower bounds, that differ in the obtained approximation ratio and allowed cooling schedule, used in SA. Notably, for dense graphs, which represent the most general category, we present particularly robust results that extend beyond SA, applying to *any* cooling schedule, even adaptive ones. While the specifics of our results, and their proofs, differ across graph classes, the common thread is this: We establish the existence of graph families where either MP or SA must run for an exponentially large number of steps to approximate the maximum independent set. The only exception is our lower bound for trees where we study the time to find the *optimal* solution (as we will see in this case that MP does succeed in efficiently finding an *approximate* solution for tree instances). Finally, we observe that our results imply that SA cannot sample a uniformly random independent set efficiently when the input is a bipartite graph, answering a recent question raised in [32].

Proving lower bounds for MP and SA has proven difficult to achieve even for instances where the independent set problem (or equivalently the clique problem) is believed to be intractable such as sparse random graphs [13] and the planted clique problem [12, 35]. Despite significant effort, it is not known whether MP and SA in their full generality can find efficiently the optimal solution in these instances. Tackling first the easier challenge of proving the limitations of these algorithms for worst-case instances could be instrumental in proving superpolynomial-time lower bounds for MP for finding nearly optimal solutions in sparse random graphs or random graphs containing a planted clique.

It is well known that the independent set problem is NP-hard not only to compute exactly but also to approximate [30, 63, 5]. Nevertheless, the study of *unconditional* limitations for the quality of approximate solutions to NP-hard problems that can be achieved efficiently by specific algorithmic methods such as linear and semi-definite programming has received significant attention [45, 21, 62]. We believe that proving unconditional limitations for the quality of approximation that can be efficiently achieved by widely used algorithmic methods such as SA and MP is of interest as well. One takeaway of the current work is that rigorous mathematical study of these algorithms can be achieved from first principles. We hope that this will encourage the mathematical study of the performance of SA and MP for additional NP-hard optimization problems.

The exact dynamics of the Metropolis process, Simulated Annealing, and the various variants we consider for the independent set problem are introduced in Section 2, and we refer the reader there for exact definitions. Let us mention that the main differences between the different algorithms lie in how the temperature, also called (inverse) fugacity, changes

Algorithm 1 Universal Metropolis Process (UMP).

Input: $G = (V, E)$ a graph, T the number of steps, $\{\lambda_t\}_{t=1}^T$ a fugacity schedule

- 1: $I_0 \leftarrow \emptyset;$ ▷ Empty IS initialization
- 2: **for** $t \leftarrow 1$ to T **do**
- 3: Sample a vertex $v_t \in V$ u.a.r.;
- 4: Sample a real $\zeta_t \in [0, 1]$ u.a.r.;
- 5: $I_t \leftarrow \text{UPDATE}(I_{t-1}, v_t, \zeta_t, \lambda_t);$ ▷ Algorithm 2
- 6: **end for**

Output: I_{t^*} where $t^* = \arg \max_{t \in \{1, \dots, T\}} |I_t|$

Algorithm 2 $\text{UPDATE}(I, v, \zeta, \lambda)$.

Input: I an IS, v a vertex, $\zeta \in [0, 1]$, $\lambda \in [1, \infty]$

- 1: **if** $v \notin I$ **then**
- 2: $I' \leftarrow \begin{cases} I \cup v, & \text{if } I \cup v \text{ is an IS;} \\ I, & \text{otherwise.} \end{cases}$
- 3: **else** (i.e., $v \in I$)
- 4: $I' \leftarrow \begin{cases} I \setminus v, & \text{if } \zeta \leq \frac{1}{\lambda}; \\ I, & \text{otherwise.} \end{cases}$
- 5:
- 6: **end if**

Output: I'

over the execution of the algorithm. In the classical Metropolis process, the temperature is fixed and does not change, while for Simulated Annealing there is some fixed schedule for decreasing the temperature over time. Some of our results also apply to a more general class of algorithms where the temperature can be chosen adaptively during the algorithm's execution. In the sequel, for simplicity we shall colloquially refer to all these algorithms as the Metropolis process (MP) and make sure to mention the different temperature schedules when relevant.

2 The Metropolis Process

In this section, we introduce the different variants of the Metropolis process for which we prove lower bounds. Our description differs slightly from the algorithm described in the introduction; we will work on a logarithmic scale for the temperature and parametrize the algorithm by the inverse of the temperature, also called *fugacity*. Ultimately this is a choice for convenience and the two descriptions are equivalent.

All considered algorithms for finding (random) large independent sets of a given input graph fall under one very general stochastic process, called the *Universal Metropolis Process (UMP)*. Among others, UMP incorporates the Randomized Greedy algorithm, the Metropolis process, and the Simulated Annealing process. Let $G = (V, E)$ be an input graph, and $T \in \mathbb{N}$ be the number of steps of the process. For each $t \geq 1$, let $\lambda_t \in [1, \infty]$ be the fugacity, or inverse-temperature, at time t . We refer to the collection $\{\lambda_t\}_{t=1}^\infty$ of all fugacities as the *fugacity schedule* of the process. With these definitions, the Universal Metropolis Process is described by Algorithms 1 and 2.

We observe that:

- If $\lambda_t \equiv \infty$ for all $t > 0$, then UMP corresponds to the Randomized Greedy algorithm where in each iteration a vertex is chosen randomly and added to the maintained independent set if it is not a neighbor of a previously added vertex. As the deletion probability is zero, no deletions of added vertices are possible.
- If $\lambda_t \equiv \lambda$ for all $t > 0$ and for some $\lambda \in [1, \infty)$, independent of t , then UMP corresponds to the Metropolis process with fugacity λ , which is a Markov chain whose stationary distribution is the Gibbs distribution μ of the hardcore model on G (i.e., $\mu(I) \propto \lambda^{|I|}$ for each independent set I of G).
- If $\{\lambda_t\}$ forms a predetermined non-decreasing sequence, i.e. $\lambda_1 \leq \lambda_2 \leq \dots$, then UMP corresponds to the Simulated Annealing algorithm with fugacity schedule $\{\lambda_t\}$.

Of course, UMP goes beyond these three well-known cases and allows for, say, non-monotone fugacities schedules (that could depend on the input graph G in complicated ways). In general, if $\{\lambda_t\}$ is some arbitrary deterministic sequence or if it is random, but independent from the randomness of $\{I_t\}$, we shall call it a *non-adaptive* schedule. On the other hand, if λ_t can depend on $\{I_s\}_{s=1}^{t-1}$ (and so it is necessarily random), we call the schedule *adaptive*. As an example for an adaptive schedule, one may set $\lambda_t = \frac{1}{|I_t|+1}$ where I_t is the currently maintained independent set, resulting with decreased deletion probability as the cardinality of I_t increases.

3 Our results

Our main result consist of exponential lower bounds on the time complexity of MP when approximating the value of $\alpha(G)$. We construct infinite families of graphs $G_1, G_2, \dots, G_n, \dots$ (with G_n having n vertices) such that the following holds. There is a function $p: \mathbb{N} \rightarrow (0, 1)$ and a constant $\eta > 0$, such that if MP runs for fewer than e^{n^η} iterations the probability it will find an independent set in G_n larger than $p(n)\alpha(G_n)$ is at most e^{-n^η} . In other words, when run for less than an exponential number of steps, MP gives a multiplicative approximation of at most $p(n)$ to $\alpha(G_n)$. As an instructive case, for general graphs, we show that one can take $p(n) = \frac{1}{n^{1-2\varepsilon}}$, while $\alpha(G_n) = n^{1-\varepsilon}$, for some $\varepsilon > 0$ arbitrarily small. Thus, even though G_n contains an independent set of nearly linear size, MP may struggle to even find an independent set of size n^ε . All of our results hold when MP is initialized from the empty set. As previously noted, proving lower bounds for these algorithms when starting from a given state has proven challenging.

Results for general graphs

Our first main result is rather general and establishes lower bounds for the classical Metropolis process (with constant temperature) on graphs parametrized by their average degree. In Section 3.1 we outline the key ideas used in the proof and the complete proof can be found in [11].

► **Theorem 1.** *Let $\{d_n\}_{n=1}^\infty$ satisfy $d_n \geq C$ for some large enough constant $C > 0$, and $d_n = o\left(\frac{n}{\log^2(n)}\right)$. There exists a sequence of graphs $\{G_n\}_{n \geq 0}$ satisfying:*

- G_n has $\Theta(n)$ vertices, the average degree of G_n is $\Theta(d_n)$, and $\alpha(G_n) = \Theta\left(\frac{n}{\log(d_n)}\right)$.

■ If $\{I_t\}_{t \geq 0}$ is the process of independent sets maintained by MP with any fixed¹ temperature,

$$\mathbb{P} \left(\max_{t \leq \exp\left(\frac{n}{C'd_n}\right)} |I_t| \geq C' \frac{\log(d_n)}{d_n} \alpha(G_n) \right) \leq \exp \left(-\frac{n}{d_n \log(d_n)} \right),$$

where $C' > 0$ is a universal constant.

Let us unpack Theorem 1 and consider the extreme cases for the average degree. The largest degree we can take and still obtain super-polynomial bounds is $d_n = \frac{n}{\text{polylog}(n)}$. In this case, Theorem 1 guarantees that G_n has an independent set of nearly linear size $\frac{n}{\text{polylog}(n)}$. However, with any fixed temperature, if MP runs for only a polynomial number of steps, it will fail to find an independent set of even polynomial size and will only result in a set of size $\text{polylog}(n)$. To get exponential lower bounds we can slightly lower the average degree and take $d_n = n^{1-\varepsilon}$ for any fixed $\varepsilon > 0$. For these slightly sparser graphs, if MP runs for $\exp(n^\varepsilon)$ iterations, it will only find a set of size at most $\tilde{O}(n^\varepsilon)$. To put this result in context, as was mentioned above, it is known that it is NP-hard to approximate the maximum independent set to within an $O\left(\frac{1}{n^{1-\varepsilon}}\right)$ factor [30, 63, 41] for any $\varepsilon \in (0, 1)$. Thus, the exponential lower bound is predicted by this hardness result and should be seen as an unconditional proof of this prediction for MP.

Theorem 1 also applies when the average degree of the graph is a constant, that does not depend on the number of vertices. For these sparse graphs, there is extensive literature surrounding the question of approximating the maximum independent set [28, 5, 29, 3]. Hence, it is interesting to study the approximation achieved by MP (with polynomial running time) for sparse graphs. Theorem 1 allows to take $d_n \equiv d$, for some large enough constant average degree $d > 0$ and obtain a sparse graph. For our sparse graphs, MP will only find an $O\left(\frac{\log(d)}{d}\right)$ approximation of $\alpha(G)$. As an algorithmic counterpart to our lower bound, the randomized greedy algorithm will find an independent set of expected size at least $\frac{n}{d+1}$. Below, in Section 2 we explain how the randomized greedy algorithm can be instantiated as an MP algorithm, which shows that our lower bound is tight up to the $\log(d)$ factor.

Simulated Annealing in dense graphs

To go beyond MP, and allow the temperature to change over time, we specialize Theorem 1 to denser graphs. A key appealing feature of our result in this case is that the theorem applies to any sequence of temperatures. In particular, the sequence can be adaptive (see Section 2 for the exact meaning of an adaptive sequence) and may be changed adversarially during the algorithm's execution.

- **Theorem 2.** *For every $\varepsilon \in (0, \frac{1}{3})$, there exists a sequence of graphs $\{G_n\}_{n \geq 0}$ satisfying:*
- G_n has $\Theta(n)$ vertices and $\alpha(G_n) = \Theta(n^{1-2\varepsilon})$.
 - *For any temperature schedule, which can be adaptive, if $\{I_t\}_{t \geq 0}$ is the process of independent sets maintained by MP, then*

$$\mathbb{P} \left(\max_{t \leq e^{n^\eta}} |I_t| \geq 2n^\varepsilon \right) \leq e^{-n^\eta},$$

for some constant $\eta > 0$.

¹ By fixed we mean that the temperature does not change during the algorithm. The temperature parameter may depend on n .

As mentioned above, when the temperature is some predetermined sequence that decreases over time, the MP algorithm is also known as Simulated Annealing. Therefore, by considering this temperature scheduling, Theorem 2 bounds the best approximation ratio SA can achieve. As discussed, this bound precisely matches the best-known results that follow from NP-hardness and again serves as proof of their prediction. The theorem goes beyond SA and, unsurprisingly, shows that there is no way to change the temperature schedule (even if one is allowed to make changes during execution) to go beyond the hurdle suggested by NP-hardness results. Adaptive changes to the temperature in the SA algorithm have been suggested before [33]. We are not aware of previous rigorous results about the benefits or limitations of adaptivity when using these methods to efficiently solve NP-hard optimization problems.

The proof of Theorem 2 appears in the full version of this paper [11].

Results for bipartite graphs

A key feature of our construction of hard instances for bipartite graphs is that they are built from bipartite graphs. These graphs are then augmented by blowing up some vertices into cliques, losing the bipartite structure. Given our construction, it is also interesting to study the performance of MP on general bipartite graphs. The point is that, in this case, there exists a simple linear time algorithm to obtain a $\frac{1}{2}$ approximation of $\alpha(G)$ by finding a bipartition. Furthermore, the standard linear programming relaxation for the independent set problem can recover the exact size of $\alpha(G)$ in polynomial time. Keeping in mind the tractability of the problem for bipartite graphs, it seems natural to expect that there exists some variant of MP that will fare similarly in these instances. On the contrary, our next result shows that in general, MP with any temperature schedule fails to come close to the performance of the mentioned algorithms.

► **Theorem 3.** *Let $d_n \leq \frac{\log(n)}{100}$. There exists a sequence of bipartite graphs $\{G_n\}_{n \geq 0}$ satisfying.*

- G_n has $\Theta(n)$ vertices, average degree $\Theta(d_n)$.
- For any temperature schedule, which can be adaptive, if $\{I_t\}_{t \geq 0}$ is the process of independent sets maintained by MP, then

$$\mathbb{P} \left(\max_{t \leq e^{n^\eta}} |I_t| \geq (4 + o(1)) \frac{\log(d_n)}{d_n} \alpha(G_n) \right) \leq e^{-n^\eta},$$

for some constant $\eta > 0$.

Theorem 3 implies that there exist n -vertex bipartite graphs for which SA cannot efficiently approximate the size of the largest independent set within a ratio better than $O\left(\frac{1}{\log(n)}\right)$. We did not attempt to optimize the hardness ratio as our main point is that MP, with any temperature (thus also covering SA), fails to find an approximate solution even in instances where the independent set problem is tractable. It is possible that stronger inapproximability results hold for SA: It might be that it fails to efficiently approximate the independent set problem in n -vertex bipartite graphs within a ratio larger than $1/n^c$ for some $c \in (0, 1)$. Studying this question is left for future work.

A direct consequence of our time lower bound for SA is that SA (with empty state initialization) cannot (even approximately) sample efficiently uniformly random independent sets in bipartite graphs. It is a central open question in approximate counting whether it is possible to approximately sample in polynomial time a uniformly distributed independent set in a bipartite graph [7]. As noted, [32] asked whether SA can be used for this purpose and we answer this question negatively.

The proof of Theorem 3 and its implications for approximate counting can be found in section Section 5.

Performance of Simulated Annealing on trees

Theorem 3 shows that, even on tractable instances, MP and its variants achieve significantly worse approximation when compared to polynomial-time algorithms. In our final hardness result, we further emphasize this point by considering the, arguably, easiest class of graphs for the independent set problem: trees. Trees are a strict and simpler sub-class of bipartite graphs. A simple greedy algorithm will return the maximum independent set in polynomial time [43]. In our next theorem, we give a complete characterization of the performance of MP, with a non-adaptive temperature schedule (like in SA), on trees. In particular, we show that MP is not competitive with polynomial-time algorithms: with less than an exponential number of iterations, there are trees where it will fail to find *the* maximum independent set. We complement this hardness result by establishing that MP can return an arbitrarily good *approximation* to $\alpha(G)$ in polynomial time.

► **Theorem 4.** *The following hold:*

- *There exists a sequence of n -vertex trees $\{F_n\}_{n \geq 0}$ such that for any constant $\eta \in (0, \frac{1}{4})$, if $\{I_t\}_{t \geq 0}$ is the process of independent sets maintained by MP with any non-adaptive sequence of fugacities,*

$$\mathbb{P} \left(\max_{t \leq \exp(n^\eta)} |I_t| = \alpha(F_n) \right) \leq e^{-\Omega(\sqrt{n})}.$$

- *For any constant $\delta \in (0, 1)$ and any n -vertex forest F_n , there exists $\lambda = \lambda(\delta, n)$ such that, if $\{I_t\}_{t \geq 0}$ is the process of independent sets maintained by MP with fixed fugacity λ ,*

$$\mathbb{P} \left(\max_{t \leq \text{poly}(n)} |I_t| \geq (1 - \delta)\alpha(F_n) \right) \geq 1 - o(1).$$

The proof of Theorem 4 can be found in Section 6. Our time lower bound for finding optimal solutions in trees holds only for non-adaptive schedules. Obtaining an analogous lower bound for adaptive schedules is an interesting question for future research.

Greedy algorithms vs. MP

As a final remark, one may wonder if there are instances where MP (when restricted to polynomially many iterations) is superior to the well-studied greedy [27] and randomized greedy [24] algorithms for the maximum independent set problem. If this was not the case, one could prove lower bounds for MP by constructing hard instances for greedy algorithms. While greedy algorithms were shown to achieve comparable results to MP for certain problems [38, 8] we provide simple examples in [11] where greedy algorithms achieve significantly worse approximation compared to MP in approximating the independent set problem.

3.1 Proof approach

We first explain our proof approach for Theorems 1 and 2 which establish the failure of the Metropolis process for finding large independent sets in graphs of given edge density characterized by the average degree. We prove this by carefully constructing a family of random graphs. Naively, we would hope to use Erdős-Rényi random bipartite graphs which

are significantly unbalanced as bad instances. More specifically, suppose that the vertex set is partitioned into $V = L \cup R$ and every edge connects one vertex from L and one from R . Ideally, we want to have $|L| \ll |R|$, and be able to show that the Metropolis process is more likely to pick up vertices in L , and thus reaches independent sets mostly contained in L . However, a moment of thought immediately shows this cannot be the case. Especially, if in each step the Metropolis process picks a vertex *uniformly at random*, then vertices in R are more likely to be chosen and MP will get independent sets of large overlap with R , which is nearly optimal.

Instead of simply using an Erdős-Rényi random bipartite graph, we further augment it with a blowup construction. More specifically, we replace each vertex $u \in L$ from the smaller side with a clique of size ℓ and connect this clique to all neighbors of u . We pick ℓ sufficiently large so that $\ell|L| \gg |R|$. This immediately provides two advantages. First, in every step, MP is more likely to choose vertices from the new L , now a disjoint union of $|L|$ cliques of size ℓ , because $\ell|L| \gg |R|$. Second, independent sets of the blowup graph are in one-to-one correspondence to independent sets of the original graph since each clique can have at most one occupied vertex. Furthermore, it is much more difficult to remove an occupied vertex in order to make the corresponding clique unoccupied, since the MP has to pick the correct occupied vertex among all vertices in the clique.

We can then argue that the MP will not be able to find a large independent set for these blowup graphs with polynomially many steps. Suppose that $|L| = n$ and $|R| = kn$ and that $\ell \gg k \gg 1$. Then, after a suitable burn-in phase, we will show that the MP will pick at most a tiny fraction of vertices in R , and at least a constant fraction of vertices in L . Thus, within the burn-in phase, MP only reaches independent sets mostly contained in L . In particular, there is a set $L_1 \subseteq L$ of at least $n/10$ occupied vertices in L , and a set $R_0 \subseteq R$ of at least $(k-1)n$ unoccupied vertices in R , at the end of the burn-in phase. These vertices induce a smaller Erdős-Rényi random bipartite graph and the MP on it with the initialization L_1 will contain vertices mostly from L_1 and barely from R_0 , via simple conductance (i.e., bottleneck) arguments. Thus, within polynomially many steps the MP cannot reach independent sets with too many vertices from R_0 . In fact, the obtained independent set contains at most n vertices from R_0 with high probability and consequently has size at most $3n$ since $|L| = n$ and $|R \setminus R_0| \leq n$. Meanwhile, R is an independent set of size $kn \gg 3n$, exhibiting the failure of MP. We remark that the whole argument works even for constant k, ℓ and edge density $O(1/n)$ so that the average degree is constant, though all “ \gg ” will be replaced with explicit inequalities.

Our construction of the *bipartite* graphs, appearing in Theorem 3, is based on the t -blowup operation. In this blowup, every vertex is replaced by an independent set (“cloud”) of size t and two clouds that correspond to neighboring vertices (before the blowup) are connected by a complete bipartite graph. The main observation is that once a vertex from a cloud is chosen, MP is much more likely to keep adding vertices to the cloud as opposed to deleting vertices from it. By taking many (identical) duplicates of the t -blowup of an initial bipartite graph, we get that for a large fraction of the duplicates no cloud is deleted (assuming a vertex from it is chosen by the algorithm) in polynomial time, hence resulting in essentially the randomized greedy algorithm where deletions do not occur. To conclude the proof we need to provide a *bipartite* graph for which randomized greedy does badly: This can serve as the “base graph” on which we perform the blow-up and duplication. We prove that the random balanced bipartite graph of size $2n$ with edge probability d/n , for a large enough constant² $d > 0$, is with high probability a hard instance for randomized greedy. Our proof

² Our lower bounds can be extended to $d = O(\log(n))$.

follows a martingale argument and may be of independent interest. The limitation of greedy algorithms for coloring (and implicitly independent set) has been observed before [44] for random multipartite graphs m -vertex graphs where the partition includes $m^{\Omega(1)}$ parts. We are not aware of a previous hardness result for the randomized greedy for approximating the size of the maximum independent set in bipartite graphs.

For trees, the core ingredient of the hard instance is a “star-shaped” tree composed of a root r connected to k nodes a_1, \dots, a_k , and each node a_i has a single leaf neighbor b_i . The unique maximum independent set consists of r together with all the v_i ’s. In the first $m = k^{1/2-\varepsilon}$ iterations, MP will add roughly $m/2$ neighbors of r . Let I denote the set of indices i for these chosen neighbors. The crux of the argument is to track the configuration of the branches I and show that they behave roughly like an i.i.d. collection of random variables supported on three states (free, a_i chosen, b_i chosen) where the probability a_i is chosen is at least $1/4$. It follows that with high probability, *some* a_i will be occupied for exponential time, blocking the root r from ever being added. The argument is completed by duplicating many copies of the hard tree, ensuring the probability that the process runs for less than exponentially many iterations is exponentially small. Finally, the forest can be made into a tree by connecting all the roots of the trees to a single additional vertex. The upper bound, showing that MP can efficiently approximate the optimum solution in a tree within a factor of $1 - \delta$ for arbitrary $\delta \in (0, 1)$, is a simple consequence of the rapid mixing of MP for trees [10, 17], taking λ to be a sufficiently large constant to ensure the partition function is concentrated on large independent sets. The complete proofs of upper and lower bounds of running times for trees can be found in Section 6.

4 Further Related work

One of the earliest works studying lower bounds for the Metropolis process is due to Jerrum [35]. In his work, he considered the planted clique problem where one seeks to find a clique of size n^β for $\beta \in (0, 1)$ planted in the Erdős-Rényi random graph $G(n, \frac{1}{2})$. Jerrum proved using a conductance argument the existence of an initial state for which the Metropolis process for cliques fails to find a clique of size $(1 + \varepsilon) \log n$ assuming $\beta < \frac{1}{2}$ so long as it is executed for less than $n^{\Omega(\log(n))}$ iterations.

Several open questions were raised in [35] regarding whether one could prove the same lower bound for MP when initialized from the empty set, whether the same lower bound holds for arbitrary $\beta < 1$ (as opposed to $\beta < \frac{1}{2}$) and whether similar lower bounds could be extended to SA as opposed to MP. The recent paper [12] made the first substantial progress towards answering Jerrum’s question; when the inverse temperature satisfies $\lambda \leq O(\log(n))$, the super-polynomial lower bounds hold with respect to MP initialized from the empty set even when $\beta < 1$. Under the same assumptions, the result of [12] also applies to SA (initialized from the empty set) for a certain temperature scheduling termed simulated tempering [49]. The lower bounds in [12] do not rule out that MP could solve the planted clique problem (even for $\beta < 1/2$) when the inverse-temperature is set to $C \log n$ for a suitable constant C . Establishing a lower bound on the running time of MP for every possible temperature is mentioned as an open question in [12].

It is natural to compare the results in [12] to our lower bounds, such as Theorems 1 and 2 which apply without any restriction on the temperature and establish exponential (as opposed to quasi-polynomial) time lower bounds. Moreover in the dense case, which is the analog of $G(n, \frac{1}{2})$, our bounds go beyond the restriction of simulated tempering and cover any sequence of temperatures. It should be noted however that [12] focused on resolving

Jerrum’s questions, while our work is geared towards proving general lower bounds. Thus, for example, in the planted clique problem a quasi-polynomial lower bound is the best one can hope for, as MP can be shown to solve the problem with high probability in $n^{O(\log(n))}$ iterations. In contrast, to prove our lower bounds we choose carefully crafted instances of random distributions on graphs, which allows for more flexibility in establishing lower bounds.

In [13] exponential lower bounds were proven with respect to the *mixing time* of MP for independent sets in sparse random graphs $G(n, \frac{d}{n})$ assuming d is a large enough constant. Observe however that lower bounds on the mixing time do not imply lower bounds for the time it takes MP to encounter a good approximation of $\alpha(G)$. It is still open whether MP (with empty set initialization) fails to find an independent set of size $(1 + \varepsilon) \frac{\log(d)}{d} n$ in polynomial time in $G(n, \frac{d}{n})$. While the setting and proofs in [13] are different from those in our paper, a common theme in both papers is that a barrier for MP is the need to delete many vertices from a locally optimal solution in order to reach a superior approximation to the optimum.

Few additional lower bounds are known for the time complexity of SA when used to approximately solve combinatorial optimization problems. Sasaki and Hajek [57] proved that, for certain instances, when searching for a maximum matching in a graph, the running time of SA can be exponential in the number of vertices (even when initialized from the empty set). Let us stress the fact that the lower bound in [57] concerns finding an *exact* solution, rather than approximating the solution. Moreover, their family of hard instances cannot be used to prove SA requires super-polynomial time to approximate maximum matching (and hence the independent set problem) within a factor of α for a fixed $\alpha \in (0, 1)$, as they prove that for every fixed $\varepsilon \in (0, 1)$ MP yields a $1 - \varepsilon$ multiplicative approximation for the maximum matching problem in polynomial time. A different proof showing that MP can find a $1 - \varepsilon$ approximation for maximum matching in polynomial time, was discovered later in [36].

In [61] it is shown, by providing a family of hard instances, that SA cannot find in polynomial time a $1 + o(1)$ multiplicative approximation for the minimum spanning tree problem. However, this result cannot be extended in a substantial way to show that SA fails to find a $1 + \varepsilon$ approximation for *fixed* $\varepsilon > 1$: For the MST problem (with non-negative weights), it was proven in [15] that SA can find a $1 + \varepsilon$ approximation for the optimal solution in polynomial time for any fixed $\varepsilon > 0$, extending an earlier result of [61]. In another direction, [6] proves exponential lower bounds on the mixing time of SA-based algorithms, which can be seen as a crucial hindrance for finding approximate solutions. Manthey and van Rhijn [48] have studied recently the performance of SA for the TSP problem in certain random instances. They conjecture that for these random instances SA will require exponential time to find the optimal solution.

In terms of approximation ratios that can be achieved in polynomial time by SA, [39, 40] provide extensive empirical simulations for SA when applied to NP-hard optimization problems such as min bisection and graph coloring. In terms of rigorous results regarding the approximation ratio achieved efficiently by SA in the worst case, [25] provides an algorithm inspired³ by SA that achieves in polynomial time a 0.41-approximation for unconstrained submodular maximization and a 0.325-approximation for submodular maximization subject to a matroid independence constraint. In [59] it is proven that certain properties associated with the energy landscape of a solution space may lead SA to efficiently find the optimal solution.

³ The authors in [25] mention that “We should remark that our algorithm is somewhat different from a direct interpretation of simulated annealing.” For more details, see [25].

Compared to SA, somewhat more is known regarding lower bounds for MP. Sasaki [56] introduced families of n -vertex instances of min-bisection and the traveling salesman problem (TSP) where MP requires exponential time to find the optimal minimum solution. Sasaki's proof is based on a "density of states" argument. The main step is to show, via a conductance argument, that when the number of optimal solutions is smaller by an exponential multiplicative factor than the number of near-optimal solutions, there exists an initialization where the expected hitting time of the optimal solution is exponential in n . It is explicitly mentioned in [56] that the proof methods do not imply lower bounds for SA. While our hard instance for trees has the property that the number of optimal solutions of value OPT is smaller by an exponential factor than the number of solutions of value $\text{OPT}-1$, our proof that SA requires exponential time to find the optimal solution differs from the proofs in [56] (indeed our proof applies for SA whereas the proof in [56] applies only for the fixed temperature case). The paper [50] contains constructions of instances of the traveling salesman problem (TSP) where MP takes exponential time to find the optimal solution but SA with an appropriate cooling schedule finds a tour of minimum cost in polynomial time. Both [56, 50] prove lower bounds for *exact* computation of the optimum and do not prove lower bounds for algorithms approximating the optimum. An informative survey of these and additional results related to SA and MP can be found in [34].

Recently, MCMC methods were shown useful in algorithmic applications despite slow mixing [47]. For example, despite exponential mixing time of the Glauber dynamics (a lazy version of MP) for the hardcore model in bipartite graphs [16], it can find an independent set of size $\Omega(\frac{\log d}{d}n)$ in an n -vertex graph of max degree d in polynomial time.

There is an extensive literature on efficient approximation algorithms for the independent set problem. As mentioned, for general n -vertex graphs, it is NP-hard to approximate $\alpha(G)$ within a factor of $\frac{1}{n^{1-\varepsilon}}$ for any $\varepsilon \in (0, 1)$ [30, 63]. Under a certain complexity-theoretic assumption it was shown in [41] that approximating the size of an independent set in n -vertex graphs within a factor larger than $2^{(\log n)^{3/4+\gamma}}/n$ (for arbitrary $\gamma > 0$) is impossible. The current best efficient approximation algorithm for the independent set problem achieves a ratio of $\Omega(\frac{\log(n)^3}{n \log \log(n)^2})$ [18]. For graphs with average degree d it has long been known that a simple greedy algorithm achieves an approximation of $\Omega(\frac{1}{d})$. This bound has been gradually improved [28, 29], and the state of the art [3] is an algorithm based on the Sherali-Adams hierarchy achieving an approximation ratio of $\tilde{\Omega}\left(\frac{\log(d)^2}{d}\right)$ for graphs of maximum degree d (the $\tilde{\Omega}$ sign hides $\text{poly}(\log \log d)$ multiplicative factors). The running time of this algorithm is polynomial in n and exponential in d . This matches, up to $\text{poly}(\log \log d)$ factors, the lower bound in [5] showing that obtaining an approximation ratio larger than $O\left(\frac{\log^2(d)}{d}\right)$ is NP-hard in graphs of maximum degree d (assuming d is a constant independent of n). In contrast to these lower bounds our lower bounds for the time complexity needed to find approximate solutions apply to specific algorithms (MP and SA). On the other hand, our results are unconditional and also apply to instances (such as bipartite graphs) where polynomial-time algorithms are known to find the optimal solution.

5 Hardness Results for Bipartite Graphs

In this section, we focus on bipartite graphs and prove Theorem 3. The main technical novelty of the proof is a reduction between the Metropolis algorithm and the randomized greedy algorithm on bipartite graphs. The idea is to blow up a hard instance for randomized greedy in a way that makes the added randomness of Metropolis inefficient.

Recall that the randomized greedy algorithm is equivalent to the Metropolis process at 0 temperature, or alternatively $\lambda_t \equiv \infty$. In other words, the algorithm chooses random vertices uniformly at random, adds them to a growing independent set whenever possible, and never deletes them. Thus, the algorithm always terminates after each vertex is chosen at least once, which happens in finite time almost surely. For the remainder of this section, for a graph G , we shall denote by I^G , the (random) independent set obtained at the termination of the randomized greedy algorithm on G .

Having introduced the randomized greedy algorithm Theorem 3 is now an immediate consequence of the following two results.

► **Proposition 5.** *Suppose that there exists a family of bipartite graphs $\{G_n\}_{n \geq 0}$ on $\Theta(n)$ vertices, and a function $r : \mathbb{N} \rightarrow (0, 1)$, such that,*

$$\mathbb{P}(|I^{G_n}| > r(n)\alpha(G_n)) = e^{-n^\eta},$$

for some $\eta > 0$. Then, for $\varepsilon > 0$ small enough, there exists a family of bipartite graphs $\{\tilde{G}_n\}_{n \geq 0}$ on $\text{poly}(n)$ vertices, such if I_t stands for the metropolis process on \tilde{G}_n with any temperature,

$$\mathbb{P}\left(\max_{t < e^{n^{\eta'}}} |I_t| \geq \left(r(n) + \frac{2}{n^{1-\varepsilon}}\right) \alpha(G_n)\right) = e^{-n^{\eta'}},$$

for some $\eta' \leq \eta$.

► **Proposition 6.** *Let $d_n < \frac{\log(n)}{100}$ be a sequence of numbers. There exists a sequence of bipartite graphs G_n on $\Theta(n)$ vertices and average degree $\Theta(d_n)$, such that*

$$\mathbb{P}\left(|I^{G_n}| > (4 + o(1)) \frac{\log(d_n)}{d_n} \alpha(G_n)\right) = e^{-n^\eta},$$

for some $\eta > 0$.

Proposition 5 essentially says that hard instances for randomized greedy can be used to construct hard instances for Metropolis, with any cooling schedule, and Proposition 6 asserts that hard instances for randomized greedy do exist. Theorem 3 immediately follows by coupling these two facts and appropriately choosing d_n . Thus, the rest of this section is devoted to the proofs of the two propositions. In Section 5.1 below we prove Proposition 5. The proof of Proposition 6 can be found in the full version [11].

Implications to sampling independent sets in bipartite graphs

Proposition 5 implies the existence of an infinite family of $2n$ -vertex bipartite graphs G_n with both sides of cardinality n such that the following holds. When SA is ran on G_n for polynomially many iterations, it will return with probability $1 - o(1)$ an independent set of size at most $n/100$. Namely, the maximum cardinality of an independent set the SA algorithm will encounter is at most $n/100$ (The constant $1/100$ can be improved to a function $g(n)$ tending to zero with n but we choose it for simplicity and as it suffices for our purpose). It follows that SA will fail to provide a (nearly) uniform sample of independent sets in polynomial time in G_n , as it will only encounter independent sets of size at most $n/100$ whose total number is at most $\binom{2n}{n/100} \leq (600)^{n/100}$: a negligible fraction of the total number of independent sets in G_n which is at least 2^{n+1} .

5.1 From randomized greedy to Metropolis

We begin by explaining how to blow-up hard instances for randomized greedy into hard instances for Metropolis. Let G be a graph on n vertices $\{v_i\}_{i=1}^n$, and denote by I^G the (random) independent set obtained by running randomized greedy on G . Assume that, for some $r(n) > 0$, and $\eta > 0$,

$$\mathbb{P}(|I^G| > r(n)\alpha(G)) \leq e^{-n^\eta}. \quad (1)$$

Our goal is to show that there exists a graph for which similar estimates hold when we replace randomized greedy with the metropolis process with polynomially many steps.

Let us describe the hard instance. For $K, M > 10$, define the the graph $G^{K,M}$ in the following way:

1. First, let G^M be the disjoint union of M copies of G .
2. $G^{K,M}$ is obtained as the K blow-up of G^M . That is, every node is replaced by an independent set of size K , and a complete bipartite graph replaces every edge.

As usual, we shall write I_t for the set maintained by MP on G and enumerate the vertices of $G^{K,M}$ as $v_{i,m,k}$, the k^{th} element in the blow-up of v_i in the m^{th} copy of G . We shall refer to the set of vertices $v_{i,m} = \{v_{i,m,k}\}_{k=1}^K$ corresponding to a vertex v_i and in the m^{th} copy, as a *cloud*. Clearly, every cloud has exactly K vertices. The following quantity shall play a central role,

$$v_{i,m}^t := |I_t \cap \{v_{i,m,1}, \dots, v_{i,m,K}\}|,$$

the load in the cloud $v_{i,m}$, at time t . The idea is that once a cloud becomes occupied, MP is more likely to add more vertices from the cloud than to remove existing ones. Thus it is very unlikely that a cloud will become empty once occupied. This should be seen as an analogy to randomized greed; as long as no cloud has emptied one can simulate randomized greedy on a given component of the graph. Having many different copies ensures that on most copies no cloud will empty.

In light of this, we first show that once the load becomes positive, and so a cloud becomes occupied, it is very unlikely to drop to 0 again.

► **Lemma 7.** *For any $t_0 \geq 0$, and $t_0 < t < K^{\frac{K^{1/3}}{2}}$ it holds that,*

$$P(v_{i,m}^{t+t_0} = 0 | v_{i,m}^{t_0} = 1) \leq \frac{C}{K^{1/3}},$$

for some absolute constant $C > 0$.

Proof. Let $t'_0 > t_0 + K^{\frac{1}{3}}$ be the first time that $K^{\frac{1}{3}}$ where chosen from the cloud $v_{i,m}$, and observe

$$\mathbb{P}(\exists t \in [t_0, t'_0] : v_{i,m}^t = 0 | v_{i,m}^{t_0} = 1) \geq \mathbb{P}(v_{i,m}^{t'_0} = K^{\frac{1}{3}} | v_{i,m}^{t_0} = 1) \geq \frac{1}{K^{\frac{1}{3}}}. \quad (2)$$

Indeed, this estimate follows from a standard Balls and Bins argument. The probability that after the first $K^{\frac{1}{3}}$ choices of vertices in $v_{i,m}$ the algorithm deletes a vertex from I_t is at most

$$\sum_{i=1}^{K^{\frac{1}{3}}} \frac{i}{K} \geq \frac{K^{\frac{2}{3}}}{K} \geq \frac{1}{K^{\frac{1}{3}}},$$

since by the union bound this upper bounds the probability $K^{\frac{1}{3}}$ balls will have at least one collision when distributed randomly across K bins. This estimate gives the right inequality in (2). The left inequality in (2) then follows from the fact that to reach a load $K^{\frac{1}{3}}$ every chosen vertex, out of the $K^{\frac{1}{3}}$ was added to the cloud, and so no vertex was removed.

Conditional on the event $v_{i,m}^{t'_0} = K^{\frac{1}{3}}$ denote now $Y_t = v_{i,m}^{t+t'_0}$ and observe that as long as $Y_t < 2K^{\frac{1}{3}}$ we have that, at any t in which the value Y_t changes,

$$P(Y_t - Y_{t-1} = 1) \geq \frac{K - 2K^{\frac{1}{3}}}{K},$$

$$P(Y_t - Y_{t-1} = -1) \leq \frac{2K^{\frac{1}{3}}}{K}.$$

Define the stopping time $\tau = \min\{t | Y_t = 2K^{\frac{1}{3}} \text{ or } Y_t = 0\}$. Similar to the proof of Theorem 2, since Y_t is stochastically dominated by a biased random walk with the above increments and since $Y_0 = K^{\frac{1}{3}}$, the results in [19, (2.8), Chapter XIV.2] imply

$$\mathbb{P}(Y_\tau = 0) \leq \left(\frac{2K^{\frac{1}{3}}}{K - 2K^{\frac{1}{3}}} \right)^{K^{\frac{1}{3}}} \leq \left(4 \frac{K^{\frac{1}{3}}}{K} \right)^{K^{\frac{1}{3}}} \leq 4K^{-\frac{2K^{\frac{1}{3}}}{3}}.$$

The proof concludes by iterating this argument for a polynomial number of steps and applying a union bound. \blacktriangleleft

For fixed time $t > 0$ we now define the number of deloaded clouds as

$$\text{deload}(t) := \#\{v_{i,m} | \exists t' < t'' \leq t \text{ such that } v_{i,m}^{t'} = 1 \text{ and } v_{i,m}^{t''} = 0\}.$$

In words, $\text{deload}(t)$ measures the number of clouds that were at some point occupied (had at least one vertex chosen by the algorithm) and later become unoccupied (all vertices in the cloud were deleted at a later point). The main upshot of the previous result is that the number of deloaded vertices remains very small after polynomially many iterations.

► **Lemma 8.** *Suppose that $C \frac{nM}{K^{\frac{1}{3}}} \geq 1$. Then, for any $t < K^{\frac{K^{1/3}}{2}}$, and $\varepsilon > 0$*

$$\mathbb{P}\left(\text{deload}(t) \geq n^\varepsilon \frac{nM}{K^{\frac{1}{3}}}\right) = e^{-\Omega(n^\varepsilon)}.$$

Proof. Observe that $G^{K,M}$ contains nM clouds. Thus, for $t < K^{\frac{K^{1/3}}{2}}$, by Lemma 7, $\text{deload}(t)$ is stochastically dominated by $B := \text{binomial}\left(nM, \frac{C}{K^{\frac{1}{3}}}\right)$. Since $\mathbb{E}[B] = C \frac{nM}{K^{\frac{1}{3}}} \geq 1$, by Chernoff's inequality,

$$\mathbb{P}\left(B \geq n^\varepsilon \frac{nM}{K^{\frac{1}{3}}}\right) \leq e^{-\Omega(n^\varepsilon)}.$$

The proof is complete. \blacktriangleleft

We can now prove Proposition 5.

Proof of Proposition 5. Set $M = n$ and $K = n^6$, so that $\frac{nM}{K^{\frac{1}{3}}} = 1$. By Lemma 8, with probability $e^{-\Omega(n^\varepsilon)}$ at most n^ε clouds were de-loaded by time $t \leq K^{\frac{K^{1/3}}{2}}$. With no loss of generality, they belong to the first n^ε copies in G^M , $\{G_m\}_{m=1}^{n^\varepsilon}$. On the other $M - n^\varepsilon$ copies, $\{G_m\}_{m=n^\varepsilon+1}^M$, since no deloading happened we can couple Metropolis on G_m with (a lazy version of) randomized greedy on the base graph G in the following way:

If Metropolis chooses vertex $v_{i,m,k}$, from cloud $v_{i,m}$ we let randomized greedy choose v_i from G . Since all clouds have the same size, the probability of choosing a vertex from cloud $v_{i,m}$ is equal to the probability of choosing the vertex v_i . Moreover, since edges exist only between clouds, whenever $v_{i,m,k}$ can be added to I_t then either v_i can also be added, if $v_{i,m}^t = 0$, or v_i was already added, if $v_{i,m}^t > 0$. If metropolis removes $v_{i,m,k}$ then randomized greedy does nothing and maintains its chosen v_i in the independent set. This coupling remains valid until the first deloading happens in G_m .

Thus, by the assumption (1) on the base graph G , with probability $1 - e^{-\Omega(n^\varepsilon)}$ for every $m \geq n^\varepsilon$, $\mathbb{P}(|I_t \cap G_m| > r(n)K\alpha(G)) \leq e^{-n^\alpha}$, where we allow metropolis to fill in all K vertices from every occupied cloud. So, suppose that $\varepsilon < \alpha$, then

$$\mathbb{P}(\forall m \geq n^\varepsilon, |I_t \cap G_m| < r(n)K\alpha(G)) \geq 1 - 2Me^{-n^\alpha}.$$

On the other hand, clearly for $m \leq n^\varepsilon$ we have $|I_t \cap G_i| \leq Kn$. It follows that

$$\mathbb{P}(|I_t| \leq Kn \cdot n^\varepsilon + r(n)KM\alpha(G)) \geq 1 - 2Me^{-n^\alpha}.$$

Finally, by construction $G^{K,M}$ has a maximal independent set of size $\alpha(G)Mk$, so the approximation ratio is $\frac{Kn \cdot n^\varepsilon + r(n)KM\alpha(G)}{\alpha(G)MK}$. Let us choose now $M = n$ and $K = n^2$ to get an approximation ratio of $\frac{n^\varepsilon}{\alpha(G)} + r(n) \leq \frac{2}{n^{1-\varepsilon}} + r(n)$. Here we have used the fact that if G is a bipartite graph on n vertices then $\alpha(G) \geq \frac{n}{2}$. ◀

6 Lower and Upper Bounds for the Time Complexity of Simulated Annealing in Trees

Here we analyze the performance of MP on trees and forests with an aim to prove Theorem 4. The proof of Theorem 4 is separated into two parts. First, in Section 6.1 we construct a *deterministic* hard instance for MP. The hard instance is a union of identical trees, each having weak hardness guarantees. It turns out that taking polynomially many copies of the same tree is enough to imply exponential lower bounds, and so proves the first point of Theorem 4. In Section 6.2 we apply recent results about mixing times of MP on graphs with bounded treewidth to prove the second point of Theorem 4.

6.1 Exponential lower bound

We consider MP with an arbitrary non-adaptive fugacity schedule λ_t , and fix a small constant $\varepsilon \in (0, 1/2)$. To construct the hard instance, first consider a “star-shaped” tree T_k that consists of a root r connected to k nodes a_1, \dots, a_k , and each node a_i has a single leaf neighbor b_i . That is, T_k consists of a root connected to k edge-disjoint length-2 paths. Let $A = \{a_1, \dots, a_k\}$ and $B = \{b_1, \dots, b_k\}$. The unique optimal independent set in T_k is $r \cup B$, which has size $k + 1$.

We first describe the first phase of the algorithm and show that it is hard for MP to include the root.

► **Lemma 9** (Burn-in phase). *The following holds with high $(1 - o_k(1))$ probability. After $m = k^{1/2-\varepsilon}$ iterations, MP has at least $(1/2 - \varepsilon)m$ vertices from A (and, as a result, does not include the root r).*

Proof. With high probability, the root r is not selected during the first m iterations. By a standard balls-and-bins argument [53], with high probability the vertices selected during the first m iterations are all distinct, and furthermore at most one vertex from each branch is

selected (a_i or b_i , but not both). Thus, MP adds all the vertices that are selected during the first m iterations and does not delete any. With high probability, at least $(1/2 - \varepsilon)m$ of the selected vertices belong to A . \blacktriangleleft

Now condition on the first m steps of MP and suppose the high-probability event from Lemma 9 holds. Letting $I \subseteq [k]$ denote the set of branches i for which MP includes a_i at the end of m steps, we have $|I| \geq (1/2 - \varepsilon)m$. We will now focus on only the branches I and show that MP continues to include at least one of the corresponding a_i 's for exponential time, blocking the root r from being added.

For the analysis, it will be convenient to consider an auxiliary process MP', defined as follows. MP' has the same underlying graph T_k and starts at the same state as MP at timestep m . MP' has the same update rule as MP except it *never* adds the root r (even if r is selected and none of its neighbors are present). The random choices of the two processes are coupled so that MP and MP' share the same state until the first time that MP adds the root.

Fix a time horizon $T > m$. Our goal is to show that with high probability, at each timestep $t \leq T$, MP includes at least one of the vertices $\{a_i : i \in I\}$. It suffices to show that the same holds for MP', as the presence of any a_i prevents MP from adding the root. Therefore we turn our attention to the analysis of MP'.

Now reveal, and condition on, the choice of which *branch* is selected by MP' at each timestep t for $m < t \leq T$. That is, we reveal the variable σ_t which is equal to $i \in I$ if MP' chooses either a_i or b_i at step t , and equal to \emptyset otherwise (i.e., if MP' selects r or a branch outside I). The random choice between a_i and b_i is not revealed yet.

For $i \in I$, let s_i denote the number of timesteps t (where $m < t \leq T$) for which $\sigma_t = i$. For each $i \in I$, we define an associated Markov chain $X_0^{(i)}, X_1^{(i)}, \dots, X_{s_i}^{(i)}$ on the state space $\{\mathcal{A}, \mathcal{B}, \emptyset\}$ with initial state $X_0^{(i)} = \mathcal{A}$. The state \mathcal{A} encodes that MP' has a_i (but not b_i), the state \mathcal{B} encodes that MP' has b_i (but not a_i), and the state \emptyset encodes that MP' has neither a_i nor b_i . Every time branch i updates (that is, t such that $\sigma_t = i$), the Markov chain $X^{(i)}$ updates according to the following rules.

- If the previous state is \emptyset , the new state is

$$\begin{cases} \mathcal{A} & \text{with probability } 1/2, \\ \mathcal{B} & \text{with probability } 1/2. \end{cases}$$

- If the previous state is \mathcal{A} , the new state is

$$\begin{cases} \emptyset & \text{w.p. } 1/(2\lambda_t), \\ \mathcal{A} & \text{w.p. } 1 - 1/(2\lambda_t). \end{cases}$$

- If the previous state is \mathcal{B} , the new state is

$$\begin{cases} \emptyset & \text{w.p. } 1/(2\lambda_t), \\ \mathcal{B} & \text{w.p. } 1 - 1/(2\lambda_t). \end{cases}$$

Note that (conditioned on $\{\sigma_t\}$) the Markov chains $X^{(1)}, \dots, X^{(|I|)}$ are independent (since MP' never adds the root, by definition).

► **Lemma 10.** *For any fixed $i \in I$ and any fixed $0 \leq \ell \leq s_i$, we have $\mathbb{P}(X_\ell^{(i)} = \mathcal{A}) \geq 1/4$.*

Proof. Proceed by induction on ℓ , strengthening the induction hypothesis to include both (i) $\mathbb{P}(X_\ell^{(i)} = \mathcal{A}) \geq 1/4$ and (ii) $\mathbb{P}(X_\ell^{(i)} = \mathcal{A}) \geq \mathbb{P}(X_\ell^{(i)} = \mathcal{B})$. The base case $\ell = 0$ is immediate, as $X^{(i)}$ is defined to start at state \mathcal{A} . For the inductive step, we analyze the update rules given above. If $X_\ell^{(i)}$ takes values $\mathcal{A}, \mathcal{B}, \emptyset$ according to the vector of probabilities (a, b, c) , then $X_{\ell+1}^{(i)}$ is distributed according to the vector of probabilities

$$(a', b', c') := \left(\frac{c}{2} + \left(1 - \frac{1}{2\lambda_t}\right)a, \frac{c}{2} + \left(1 - \frac{1}{2\lambda_t}\right)b, \frac{1}{2\lambda_t}(a+b) \right).$$

By induction we have $a \geq 1/4$ and $a \geq b$. Note that from $a \geq b$ we immediately have $a' \geq b'$, which proves (ii). For (i), $a \geq b$ implies $b \leq 1/2$ and so

$$a' = \frac{c}{2} + \left(1 - \frac{1}{2\lambda_t}\right)a \geq \frac{c+a}{2} = \frac{1-b}{2} \geq \frac{1}{4},$$

completing the proof. \blacktriangleleft

Therefore, by independence across branches, we have for any fixed timestep t (with $m < t \leq T$), the probability that MP' has *none* of the vertices $\{a_i : i \in I\}$ is at most $(3/4)^{|I|}$. Taking a union bound over t , the probability that MP' intersects $\{a_i : i \in I\}$ until time T is at least $1 - T(3/4)^{|I|}$. For $T \leq \exp(k^{1/2-2\epsilon})$ this is $1 - o_k(1)$, recalling $|I| \geq (1/2 - \epsilon)m = (1/2 - \epsilon)k^{1/2-\epsilon}$. As discussed above, we conclude the same result for the original process MP.

► **Proposition 11.** *Fix any constant $\eta \in (0, 1/2)$ and consider MP run on T_k with an arbitrary non-adaptive fugacity schedule λ_t . With probability $1 - o_k(1)$, MP does not add the root r at any point during the first $\exp(k^\eta)$ iterations (and thus does not find a maximum independent set).*

The hardness result in Theorem 4 now follows by bootstrapping Proposition 11.

Proof of first item in Theorem 4. Let us first prove the result for a forest rather than a tree. Consider k disjoint copies of T_k , which has a total of $k(2k+1)$ vertices. Additional isolated vertices can be added to create a forest of any desired size. To find the maximal independent set in the union, MP must add the root of every copy of T_k . Conditional on the exact number of updates per copy, MP evolves independently on each copy. Thus, the independence of the different copies and Proposition 11 yield an exponential bound $\exp(-\Omega(k))$ on the probability that it adds all the roots by time $\exp(k^\alpha)$.

We now modify the construction, turning the forest into a tree. Add a new vertex and connect it to the root of each T_k (and also to any isolated vertices that were added to pad the instance size). The maximum independent set still includes the root of every T_k . If it were not for the new vertex, we have from above that with probability $1 - \exp(-\Omega(k))$, there is at least one copy of T_k where the root is never added before time $\exp(k^\alpha)$. With the new vertex present, this copy of T_k still will never add the root before time $\exp(k^\alpha)$, as the new vertex can only affect T_k by *blocking* the root from being added. This completes the proof. \blacktriangleleft

6.2 The Metropolis process can efficiently find approximate solutions on trees

Here we show that if one seeks an *approximate* solution to the size of the largest independent set in a forest then MP finds such a solution efficiently:

Proof of second item in Theorem 4. Let N be the maximum cardinality of an independent set in F_n . Since F_n is bipartite we have that $N \geq n/2$. Set the fugacity as $\lambda \geq 4^{1/\delta + \frac{\log 2}{\delta n}}$. We upper bound the contribution of independent sets whose size is smaller than $(1 - \delta)N$ to the partition function:

$$\sum_{|I| < (1-\delta)N} \lambda^{|I|} \leq 2^n \lambda^{(1-\delta)N} \leq \lambda^N / n.$$

Therefore, for the stationary distribution of MP with this value of λ the probability we get an independent set of size smaller than $(1 - \delta)N$ is at most $1/n$. The desired result now follows from the fact [17, 10] that MP on forests mixes in time $n^{O(1)}$ (assuming λ is a fixed constant independent of n) as well as the fact [46] that after $t_{\text{mix}} \cdot \log n$ iterations of MP the total variation distance between the chain and the stationary distribution is at most $1/n$. Therefore, MP finds an independent set of size at least $(1 - \delta)N$ with probability at least $1 - 2/n$. ◀

► **Remark 12.** For graphs of treewidth t it is known that the mixing time of MP is $n^{O(t)}$ [17, 10]. Since any graph with treewidth t is t -degenerate (has a vertex of degree at most t) it has an independent set of size at least $n/(t+1)$. The argument above shows that assuming t is constant (independent of n), MP will find a $1 - \varepsilon$ approximation to the optimal solution in polynomial time with high probability.

7 Conclusion

We have proved super polynomial lower bounds on the time complexity of the SA and MP when approximating the size of a maximum independent set in several graph families. Several questions remain:

- Prove that SA cannot approximate in polynomial time $\alpha(G)$ in graphs with constant average degree d within a factor larger than $O(\log d/d)$. Currently we only know how to prove this when the temperature is fixed and does not change throughout the algorithm.
- Analyze the approximation ratio MP achieves in polynomial time for $\alpha(G)$ in *planar graphs*. A concrete first step would be to analyze how well MP approximates $\alpha(G)$ on the square grid.
- It would be interesting to compare MP (with polynomially many iterations) to the well-studied *min-degree greedy algorithm* [31, 27] which is known [27] to achieve a $3/(\Delta + 2)$ -approximation on graphs with maximum degree Δ . Whether this approximation ratio can be achieved (or improved upon) by MP in polynomial time is currently open. The new machinery introduced recently in [47] may be relevant for this question.

References

- 1 Emile Aarts and Jan Korst. *Simulated annealing and Boltzmann machines*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Ltd., Chichester, 1989. A stochastic approach to combinatorial optimization and neural computing.
- 2 David Aldous and Umesh Vazirani. “Go with the winners” algorithms. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 492–501. IEEE, 1994.
- 3 Nikhil Bansal, Anupam Gupta, and Guru Guruganesh. On the Lovász theta function for independent sets in sparse graphs. *SIAM J. Comput.*, 47(3):1039–1055, 2018. doi:10.1137/15M1051002.
- 4 Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. In *Probability and algorithms*, pages 17–29. Nat. Acad. Press, Washington, DC, 1992.
- 5 Amey Bhangale and Subhash Khot. UG-hardness to NP-hardness by losing half. *Theory Comput.*, 18:Paper No. 5, 28, 2022. doi:10.4086/toc.2022.v018a005.

- 6 Nayantara Bhatnagar and Dana Randall. Torpid mixing of simulated tempering on the Potts model. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 478–487. ACM, New York, 2004. URL: <http://dl.acm.org/citation.cfm?id=982792.982860>.
- 7 Jin-Yi Cai, Andreas Galanis, Leslie Ann Goldberg, Heng Guo, Mark Jerrum, Daniel Štefankovič, and Eric Vigoda. # bis-hardness for 2-spin systems on bipartite bounded degree graphs in the tree non-uniqueness region. *Journal of Computer and System Sciences*, 82(5):690–711, 2016. doi:10.1016/J.JCSS.2015.11.009.
- 8 Ted Carson and Russell Impagliazzo. Hill-climbing finds random planted bisections. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, pages 903–909. SIAM, Philadelphia, PA, 2001. URL: <http://dl.acm.org/citation.cfm?id=365411.365805>.
- 9 Fang Chen, László Lovász, and Igor Pak. Lifting Markov chains to speed up mixing. In *Annual ACM Symposium on Theory of Computing (Atlanta, GA, 1999)*, pages 275–281. ACM, New York, 1999. doi:10.1145/301250.301315.
- 10 Zongchen Chen. Combinatorial approach for factorization of variance and entropy in spin systems. *arXiv preprint arXiv:2307.08212*, 2023. doi:10.48550/arXiv.2307.08212.
- 11 Zongchen Chen, Dan Mikulincer, Daniel Reichman, and Alexander S Wein. Time lower bounds for the metropolis process and simulated annealing. *arXiv preprint arXiv:2312.13554*, 2023. doi:10.48550/arXiv.2312.13554.
- 12 Zongchen Chen, Elchanan Mossel, and Ilias Zadik. Almost-linear planted cliques elude the Metropolis process. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4504–4539. SIAM, Philadelphia, PA, 2023. doi:10.1137/1.9781611977554.ch171.
- 13 Amin Coja-Oghlan and Charilaos Efthymiou. On independent sets in random graphs. *Random Structures Algorithms*, 47(3):436–486, 2015. doi:10.1002/rsa.20550.
- 14 Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Virkumar Vazirani. *Algorithms*. McGraw-Hill Higher Education New York, 2008.
- 15 Benjamin Doerr, Amirhossein Rajabi, and Carsten Witt. Simulated annealing is a polynomial-time approximation scheme for the minimum spanning tree problem. In *GECCO’22—Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1381–1389. ACM, New York, 2022. doi:10.1145/3512290.3528812.
- 16 Martin Dyer, Alan Frieze, and Mark Jerrum. On counting independent sets in sparse graphs. *SIAM J. Comput.*, 31(5):1527–1541, 2002. doi:10.1137/S0097539701383844.
- 17 David Eppstein and Daniel Frishberg. Rapid mixing of the hardcore glauber dynamics and other markov chains in bounded-treewidth graphs. *arXiv preprint arXiv:2111.03898*, 2021. arXiv:2111.03898.
- 18 Uriel Feige. Approximating maximum clique by removing subgraphs. *SIAM J. Discrete Math.*, 18(2):219–225, 2004. doi:10.1137/S089548010240415X.
- 19 William Feller. *An introduction to probability theory and its applications. Vol. I*. John Wiley & Sons, Inc., New York-London-Sydney, third edition, 1968.
- 20 James Allen Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *Ann. Appl. Probab.*, 1(1):62–87, 1991. URL: [http://links.jstor.org/sici?sici=1050-5164\(199102\)1:1<62:EB0CTS>2.0.CO;2-M&origin=MSN](http://links.jstor.org/sici?sici=1050-5164(199102)1:1<62:EB0CTS>2.0.CO;2-M&origin=MSN).
- 21 Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald De Wolf. Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 95–106, 2012. doi:10.1145/2213977.2213988.
- 22 Andreas Galanis, Qi Ge, Daniel Štefankovič, Eric Vigoda, and Linji Yang. Improved inapproximability results for counting independent sets in the hard-core model. *Random Structures & Algorithms*, 45(1):78–110, 2014. doi:10.1002/RSA.20479.

- 23 Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic ising and hard-core models. *Combinatorics, Probability and Computing*, 25(4):500–559, 2016. doi:10.1017/S0963548315000401.
- 24 David Gamarnik and David A. Goldberg. Randomized greedy algorithms for independent sets and matchings in regular graphs: exact results and finite girth corrections. *Combin. Probab. Comput.*, 19(1):61–85, 2010. doi:10.1017/S0963548309990186.
- 25 Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1098–1116. SIAM, Philadelphia, PA, 2011. doi:10.1137/1.9781611973082.83.
- 26 Bruce Hajek. Cooling schedules for optimal annealing. *Math. Oper. Res.*, 13(2):311–329, 1988. doi:10.1287/moor.13.2.311.
- 27 Magnús Halldórsson and Jaikumar Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. In *Proceedings of the twenty-sixth annual ACM symposium on theory of computing*, pages 439–448, 1994. doi:10.1145/195058.195221.
- 28 Magnús M. Halldórsson and Jaikumar Radhakrishnan. Improved approximations of independent sets in bounded-degree graphs. In *Algorithm theory – SWAT ’94 (Aarhus, 1994)*, volume 824 of *Lecture Notes in Comput. Sci.*, pages 195–206. Springer, Berlin, 1994. doi:10.1007/3-540-58218-5_18.
- 29 Eran Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM J. Comput.*, 31(5):1608–1623, 2002. doi:10.1137/S0097539700381097.
- 30 Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Math.*, 182(1):105–142, 1999. doi:10.1007/BF02392825.
- 31 Dorit S Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6(3):243–254, 1983. doi:10.1016/0166-218X(83)90080-X.
- 32 Brice Huang, Sidhanth Mohanty, Amit Rajaraman, and David X Wu. Weak Poincaré inequalities, simulated annealing, and sampling from spherical spin glasses. *arXiv preprint arXiv:2411.09075*, 2024. doi:10.48550/arXiv.2411.09075.
- 33 Lester Ingber. Very fast simulated re-annealing. *Mathematical and computer modelling*, 12(8):967–973, 1989.
- 34 Thomas Jansen. Simulated annealing. In *Theory of randomized search heuristics*, volume 1 of *Ser. Theor. Comput. Sci.*, pages 171–195. World Sci. Publ., Hackensack, NJ, 2011. doi:10.1142/9789814282673_0006.
- 35 Mark Jerrum. Large cliques elude the Metropolis process. *Random Structures Algorithms*, 3(4):347–359, 1992. doi:10.1002/rsa.3240030402.
- 36 Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989. doi:10.1137/0218077.
- 37 Mark Jerrum and Alistair Sinclair. The markov chain monte carlo method: an approach to approximate counting and integration. *Approximation Algorithms for NP-hard problems*, PWS Publishing, 1996.
- 38 Mark Jerrum and Gregory B. Sorkin. The Metropolis algorithm for graph bisection. *Discrete Appl. Math.*, 82(1-3):155–175, 1998. doi:10.1016/S0166-218X(97)00133-9.
- 39 David S Johnson, Cecilia R Aragon, Lyle A McGeoch, and Catherine Schevon. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations research*, 37(6):865–892, 1989. doi:10.1287/OPRE.37.6.865.
- 40 David S Johnson, Cecilia R Aragon, Lyle A McGeoch, and Catherine Schevon. Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partitioning. *Operations research*, 39(3):378–406, 1991. doi:10.1287/OPRE.39.3.378.
- 41 Subhash Khot and Ashok Kumar Ponnuswami. Better inapproximability results for MaxClique, chromatic number and Min-3Lin-Deletion. In *Automata, languages and programming. Part I*, volume 4051 of *Lecture Notes in Comput. Sci.*, pages 226–237. Springer, Berlin, 2006. doi:10.1007/11786986_21.
- 42 S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi:10.1126/science.220.4598.671.
- 43 Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

- 44 Luděk Kučera. The greedy coloring is a bad probabilistic algorithm. *Journal of Algorithms*, 12(4):674–684, 1991. doi:10.1016/0196-6774(91)90040-6.
- 45 James R Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 567–576, 2015. doi:10.1145/2746539.2746599.
- 46 David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.
- 47 Kuikui Liu, Sidhant Mohanty, Prasad Raghavendra, Amit Rajaraman, and David X Wu. Locally stationary distributions: A framework for analyzing slow-mixing markov chains. *arXiv preprint arXiv:2405.20849*, 2024. doi:10.48550/arXiv.2405.20849.
- 48 Bodo Manthey and Jesse van Rhijn. Towards a lower bound for the average case runtime of simulated annealing on tsp. *arXiv preprint arXiv:2208.11444*, 2022. doi:10.48550/arXiv.2208.11444.
- 49 Enzo Marinari and Giorgio Parisi. Simulated tempering: a new Monte Carlo scheme. *Euro-physics letters*, 19(6):451, 1992.
- 50 Klaus Meer. Simulated annealing versus Metropolis for a TSP instance. *Inform. Process. Lett.*, 104(6):216–219, 2007. doi:10.1016/j.ipl.2007.06.016.
- 51 Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- 52 Milena Mihail. Conductance and convergence of Markov chains – a combinatorial treatment of expanders. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 526–531. IEEE, 1989. doi:10.1109/SFCS.1989.63529.
- 53 Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.
- 54 Elchanan Mossel, Dror Weitz, and Nicholas Wormald. On the hardness of sampling independent sets beyond the tree threshold. *Probab. Theory Related Fields*, 143(3-4):401–439, 2009. doi:10.1007/s00440-007-0131-9.
- 55 Ricardo Restrepo, Jinwoo Shin, Prasad Tetali, Eric Vigoda, and Linji Yang. Improved mixing condition on the grid for counting and sampling independent sets. *Probability Theory and Related Fields*, 156(1):75–99, 2013.
- 56 Galen Sasaki. The effect of the density of states on the Metropolis algorithm. *Inform. Process. Lett.*, 37(3):159–163, 1991. doi:10.1016/0020-0190(91)90037-I.
- 57 Galen H. Sasaki and Bruce Hajek. The time complexity of maximum matching by simulated annealing. *J. Assoc. Comput. Mach.*, 35(2):387–403, 1988. doi:10.1145/42282.46160.
- 58 Allan Sly. Computational transition at the uniqueness threshold. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 287–296. IEEE, 2010. doi:10.1109/FOCS.2010.34.
- 59 Gregory B. Sorkin. Efficient simulated annealing on fractal energy landscapes. *Algorithmica*, 6(3):367–418, 1991. doi:10.1007/BF01759051.
- 60 V. Černý. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *J. Optim. Theory Appl.*, 45(1):41–51, 1985. doi:10.1007/BF00940812.
- 61 Ingo Wegener. Simulated annealing beats Metropolis in combinatorial optimization. In *Automata, languages and programming*, volume 3580 of *Lecture Notes in Comput. Sci.*, pages 589–601. Springer, Berlin, 2005. doi:10.1007/11523468_48.
- 62 Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 223–228, 1988.
- 63 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *STOC’06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 681–690. ACM, New York, 2006. doi:10.1145/1132516.1132612.