Polynomial-Time Approximation of Minimum Spanning Trees: Time Complexity of Simulated Annealing and the Metropolis–Hastings algorithm

Lin

A Dissertation submitted to the Department of Mathematics of the London School of Economics and Political Science

June 12, 2025

Summary

Given an undirected, connected and weighted graph, the task in the minimum spanning tree problem is to find a subset of edges that connects all the vertices without any cycles and with the minimum possible total edge weight.

The Metropolis algorithm is Simulated Annealing with a fixed temperature. Surprisingly, in many instances of optimisation problems, Simulated Annealing is less efficient than the Metropolis algorithm with an optimally chosen fixed temperature.

In this dissertation, we give an overview of two papers: one by Wegener (2005), covering that the Metropolis algorithm with arbitrary fixed temperature is unsuccessful on instances of the minimum spanning tree problem, whereas Simulated Annealing with a geometric cooling schedule is successful; the other by Doerr, Rajabi, and Witt (2022), covering that Simulated Annealing with an appropriate cooling schedule computes arbitrarily tight constant-factor approximations to the minimum spanning tree problem in polynomial time, which is a result conjectured by Wegener (2005).

Contents

Summary ii

1 Introduction 1

2 Local Search and Iterative Improvement algorithm 3

3 Simulated Annealing and the Metropolis algorithm 5

4 The Minimum Spanning Tree Problem 8

5 Simulated Annealing beats Metropolis in Combinatorial Optimisation 12

6 Simulated Annealing as a Polynomial-time Approximation Scheme For the MST Problem 19

Bibliography 24

Chapter 1

Introduction

An instance of a combinatorial optimisation problem is formalised as a pair (S, f). We call S the solution space, which is a finite or countably infinite set of discrete variables that satisfy all the constraints of the problem. An element i of the solution space is called a solution. We call $f : S \to \mathbb{R}$ the cost function, which assigns assigns some "cost" to each solution which we seek to maximise or minimise. In a minimisation problem, the lower the f, the better the corresponding solution. The problem can then be solved by finding a solution for which f takes its minimum value, i.e. an optimal solution $i_0 \in S$ satisfying

$$f_{\text{opt}} = f(i_0) = \min_{i \in S} f(i),$$

where f_{opt} is the optimum (minimum) cost.

Combinatorial optimisation problems can be solved by randomised search heuristics such as the Metropolis algorithm (MA) and Simulated Annealing (SA). SA belongs to the class of iterative improvement algorithms and features the ability to accept inferior solutions with some probability. Such probability is governed by a parameter T which we call temperature. The value of the temperature is determined based on a cooling schedule at each time step of the algorithm. MA is SA with fixed temperature.

It is a natural question how to choose a cooling schedule for SA and whether this option is essential. Little was known before Wegener (2005) which led Jerrum and Sinclair (1996) to conclude that "it remains an outstanding open problem to exhibit a natural example in which simulated annealing with any non-trivial cooling schedule provably outperforms the Metropolis algorithm at a carefully chosen fixed value." The solution to this "outstanding open problem" is provided by Wegener (2005).

Wegener (2005) proved that SA outperforms MA on instances of the minimum spanning tree (MST) problem. More precisely, Wegener (2005) proved that on instances of the so-called "connected triangles", for a constant c and sufficiently large m[1], the probability

[1]When we say "sufficiently large m", we mean there exists $N \in \mathbb{N}$ such that for all $m \geq N$, the theorem/lemma/corollary holds.

that MA solves the MST problem with arbitrary fixed temperature within e^{cm} time steps is abounded above by $e^{-0.001m}$; whereas for all arbitrary constants $k > 0$, the probability that SA with an appropriate geometric cooling schedule solves the MST problem within $2400km \log m$[2] time steps is at least $1 - 4m^{-k}$. Clearly, SA "beats" MA on those instances.

Wegener (2005) also proved that SA with a suitable geometric cooling schedule can efficiently find optimal solutions to the MST problem when the edge weights are $(1 + \varepsilon)$-separated. He further conjectured that SA for general weights computes $(1+\varepsilon)$-approximate minimum spanning trees. Doerr, Rajabi, and Witt (2022) proved this conjecture, in that they proved that there exists a time T^* in a run of SA with cooling factor β such that at all time steps beyond T^*, the current solution is a $(1 + \varepsilon)$-approximation. Doerr, Rajabi, and Witt (2022) provided an explicit numerical value of the approximation ratio $(1 + \varepsilon)$ and improved on Wegener (2005)'s time bound.

Chapter 2 of this dissertation will cover the iterative improvement algorithm and local search concepts which form the basis of MA and SA. Chapter 3 will define MA and SA. Chapter 4 will define the MST problem and introduce Wegener (2005)'s setup of SA and MA to solve the problem. Chapter 4 will also discuss theoretical relevance of the results in Wegener (2005) and Doerr, Rajabi, and Witt (2022). Chapter 5 will give a full proof of Wegener (2005)'s theorem that SA beats MA. Chapter 6 will cover the theorem of Doerr, Rajabi, and Witt (2022) that SA is a polynomial-time approximation scheme for the MST problem.

[2]Throughout this dissertation, "log" refers to natural logarithms, i.e., logarithms to the base of e.

Chapter 2

Local Search and Iterative Improvement algorithm

There are infinitely many instances of a combinatorial optimisation problem, we cannot list an optimal solution for each instance. Instead, we look for algorithms that can compute an optimal solution given an instance. Roughly speaking, an algorithm consists of a set of inputs and a sequence of instructions each of which can include elementary steps, such that for each valid input the computation of the algorithm is a uniquely defined finite series of elementary steps which produces a certain output. Examples of elementary step include any one of the arithmetic operations (addition, subtraction, multiplication, division) or a comparison between two numbers.

The input to an algorithm usually consists of a list of numbers. If the input a is an integer, we can store a in binary representation using $O(\log(|a| + 2))$ bits.

Definition 1 (Input size)**.** Consider an instance (S, f) of a combinatorial optimisation problem. The input size $size(i)$ of $i \in S$ is the total number of bits needed for the binary representation of i.

Definition 2 (Running time)**.** Let A be an algorithm which accepts inputs from a set W, and let $f : \mathbb{N} \to \mathbb{R}_+$. If there exist constants $\alpha, \beta > 0$ such that A terminates its computation after at most $\alpha f(\text{size}(W)) + \beta$ elementary steps for each input $i \in W$, then we say that A runs in $O(f)$ time. We also call $O(f)$ the running time (or the time complexity) of A.

In general, algorithms that run in polynomial time are considered efficient.

Definition 3 (Polynomial running time)**.** An algorithm with rational input is said to run in polynomial-time if there exists an integer k such that it runs in $O(n^k)$ time, where n is the input size.

There are two possible approaches when trying to solve a combinatorial optimisation problem. One can either use an optimisation algorithm, yielding a globally optimal

solution in a possibly prohibitive amount of computation time or an approximation algorithm (also called heuristics), yielding an approximate solution with a polynomial running time. Simulated Annealing (SA) and the Metropolis algorithm (MA) can be viewed as heuristics. Their principles belong to the general local search framework and incorporate a number of aspects from the iterative improvement algorithm. Since these aspects play an important role in understanding MA and SA, we first introduce some local search concepts and elaborate on the iterative improvement algorithm.

The local search framework constitutes a widely used, general approach to hard combinatorial optimisation problems with the same feature of an underlying neighbourhood structure and a generation mechanism which are used to guide the search for an optimal solution.

Definition 4 (Neighbourhood structure)**.** Consider an instance (S, f) of a combinatorial optimisation problem. Let $\mathcal{N}: S \rightarrow 2^S$ be a function which defines for each solution $i \in S$ a subset $S_i \subset S$ of solutions that are considered "close" (based on definition of user) to i. We call S_i the neighbourhood of the solution i and $\mathcal{N}$ the neighbourhood structure. We call elements of S_i neighbours of i.

Local search can be thought of as a framework which we consider neighbours of the current solution as potential replacements. If we accept a new solution from this neighbourhood, then we move to that solution and consider its neighbours. In the next definitions, we consider that $\mathcal{N}$ is a neighbourhood structure associated with (S, f).

Definition 5 (Generation mechanism)**.** A generation mechanism refers to a method used to select a solution j from the neighbourhood S_i of a specific solution i.

The iterative improvement algorithm is a type of local search which starts its search by selecting an initial solution from the solution space. The generation mechanism is then applied to select a solution from the neighborhood of the current solution. If the selected solution is more optimal, it becomes the current solution. The algorithm is continued until no improvement can be found and the current solution is considered as the approximate solution for the optimisation problem.

By definition, iterative improvement algorithm terminates in a local optimum. To avoid this disadvantage while maintaining the local search concepts, one can consider to implement a strategy that permits transitions to new neighborhoods, even if these temporarily result in a worse cost function value. This is the basic idea underlying MA and SA.

Chapter 3

Simulated Annealing and the Metropolis algorithm

In the early 1980s, Kirkpatrick, Gelatt, and Vecchi (1982) and independently Černý (1985) introduced the concepts of annealing in combinatorial optimisation. Originally, these concepts were inspired by an analogy between the physical annealing of solids and the problem of solving large combinatorial optimisation problems.

In condensed matter physics, annealing is known as a thermal process for obtaining low energy states of a solid in a heat bath. The process consists of the following two steps:

- increase the temperature of the heat bath to a value at which the solid melts;
- decrease the temperature of the heat bath until the particles arrange themselves in the ground state of the solid.

Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953) introduced an algorithm which models the evolution of the physical structure of a material undergoing annealing and generates a sequence of states for a material in the following way:

Initialise from a state i of energy E_i of the material. The generation mechanism is defined as displacing the position of a particle in the material which produces a new state j of energy E_j. If the energy difference $E_j - E_i$ is less than or equal to 0, the state j is accepted as the current state. If the energy difference is greater than zero, then j is accepted with probability $e^{(-\frac{E_j - E_i}{k_B T})}$, where T is the temperature of the material and k_B is the Boltzmann constant.

This acceptance criterion is known as the Metropolis criterion and the algorithm is known as the Metropolis algorithm. The Metropolis criterion enables the acceptance of an inferior solution with a probability determined by the temperature T, which allows MA to escape local minima. The Metropolis algorithm can be generalised for solving combinatorial optimisation problems (see Algorithm 1 for the case of a minimisation problem).

Algorithm 1: the Metropolis algorithm

Initialise (i_{start}, T);
$i \leftarrow i_{\text{start}}$;
repeat
  Do generation mechanism;
  if $f(j) \leq f(i)$ **then**
    $i \leftarrow j$;
  else
    $i \leftarrow j$ with probability $e^{(-\frac{f(j)-f(i)}{T})}$;
  end
until *stopcriterion is met*;

Simulated Annealing extends the Metropolis algorithm by making the following analogies:

- States of the material correspond to solutions in the optimisation problem.
- Energy levels are analogous to the cost function.
- Temperature T is a control parameter that governs the acceptance of worse solutions.

The main additional feature of SA is a flexible temperature parameter which governs SA's acceptance criterion.

Definition 6 (Acceptance criterion of SA)**.** Let (S, f) be a combinatorial minimisation problem. For two points i and j in S, the acceptance criterion of SA for selecting the current solution i to solution j is given by the following probability:

$$\mathbb{P}(\text{accept } j) = \begin{cases} 1 & \text{if } f(j) < f(i) \\ e^{\left(-\frac{f(j)-f(i)}{T_k}\right)} & \text{otherwise} \end{cases},$$

where T_k is the temperature at time step k.

It is assumed that the user provides for each solution of the solution space a neighbourhood structure and a generation mechanism. After initialisaiton, SA first applies the generation mechanism to select a solution j from the neighbourhood S_i of the current solution i, then determine whether to accept j as the current solution based on the acceptance criterion — this is one transition to a solution. After L_k transitions, one iteration is considered complete for the temperature T_k at time step k; k is then increased by 1 subsequently for the next time step. At the next time step $k+1$, L_k and T_k are then updated based on some predefined cooling schedule. For instance, T_k may decrease geometrically: $T_k = \alpha T_{k-1}$, where $\alpha < 1$ (α is called cooling factor); L_k might remain constant or change based on the iteration. SA terminates based on user's stop criterion (see Algorithm 2 for the case of a minimisation problem).

Algorithm 2: Simulated Annealing

Initialise $(i_{\text{start}}, T_0, L_0)$;
$k \leftarrow 0$;
$i \leftarrow i_{\text{start}}$;
repeat
  for $l \leftarrow 1$ ***to*** L_k **do**
    Generation mechanism;
    if $f(j) \leq f(i)$ **then**
      $i \leftarrow j$;
    else
      $i \leftarrow j$ with probability $\exp(-\frac{f(j)-f(i)}{T_k})$;
    end
  end
  $k \leftarrow k + 1$;
  Calculate (L_k);
  Calculate (T_k);
until *stopcriterion is met*;

In early stages of the optimisation process, the temperature parameter is high, enabling SA to accept solutions that have high objective degradation hence ensuring sufficient exploration of the solution space. As the process progresses, the temperature decreases and only solutions that improve the cost function or with low cost degradation are accepted. Eventually, when the temperature approaches 0, SA will not accept solutions with any cost degradation. This feature means that SA can escape from local minima while still exhibiting the simplicity and general applicability that iterative improvement algoirthms feature.

In the next chapter, we will see an application of SA and MA to the minimum spanning tree problem.

Chapter 4

The Minimum Spanning Tree Problem

The minimum spanning tree (MST) problem is one of the most typical and well-known problems of combinatorial optimisation. Methods for its solution have generated important ideas for modern combinatorics and have played a central role in the design of computer algorithms. We will define the MST problem and cover Wegener (2005)'s setup of SA and MA for solving the MST problem.

Let $G = (V, E)$ be an undirected, connected and weighted graph which has n vertices and m edges. For the set of edges $E = \{e_1, \ldots, e_m\}$, the weight of edge $e_i \in E$ is a positive number w_i for $i \in \{1, \ldots, m\}$. We write $w_{\min} := \min\{w_i \mid i \in \{1, \ldots, m\}\}$ and $w_{\max} := \max\{w_i \mid i \in \{1, \ldots, m\}\}$ for the minimum and maximum edge weights. Let $w(E) = \sum_{e_i \in E} w_i$ be the total weight of G.

Definition 7 (Spanning tree)**.** A tree is an undirected graph in which any two vertices are connected by exactly one path. A spanning tree $\mathcal{T}$ of an undirected graph G is a subgraph that is a tree which includes all of the vertices of G.

The objective of the MST problem is to find a subset $E' \subseteq E$ such that $\mathcal{T} = (V, E')$ is a spanning tree of G with minimum total weight $w(E') = \sum_{e_i \in E'} w_i$. To implement SA to solve the MST problem, we use the bit-string representation for E': a bit string $x = (x_1, \ldots, x_m) \in \{0,1\}^m$ represents the set $E(x) = \{e_i \mid x_i = 1\}$. For a solution x, we define the cost function f as following:

$$f(x) = \begin{cases} w_1 x_1 + \cdots + w_m x_m & \text{if } (V, E(x)) \text{ is connected} \\ \infty & \text{otherwise} \end{cases}.$$

This ensures that unconnected graphs are never accepted as current solution. To ensure that we start with a finite cost value, we initialise SA with the all-ones string $x^{(0)} = (1, \ldots, 1)$. Let $x^{(t)}$ be the current solution. We define the generation mechanism as

choosing $i \in (1, \ldots, m)$ uniformly at random and flip x_i, i.e., let $y = (y_1, \ldots, y_m)$ where $y_j = x_j^{(t)}$ if $j \neq i$ and $y_i = 1 - x_i^{(t)}$. Starting from this initial string, SA can transition to solutions with fewer edges by flipping one-bits; however, it will reject any solutions that are not connected due to their infinitely high f-value. We use a geometric cooling schedule $T_{t+1} = T_t \cdot \beta$ and apply only one transition at each time step (see Algorithm 3).

Algorithm 3: SA with starting temperature T_0 and cooling factor $\beta \leq 1$ for the minimisation of $f : \{0,1\}^n \to \mathbb{R}$

Initialise $(x^{(0)}, T_0, \beta)$;
for $t \leftarrow 0, 1, 2, \ldots$ **do**
  Generation mechanism;
  if $f(y) \leq f(x^{(t)})$ **then**
    $x^{(t+1)} \leftarrow y$;
  else
    $x^{(t+1)} \leftarrow y$ with probability $\exp\left(-\frac{f(y)-f(x^{(t)})}{T_t}\right)$;
    $x^{(t+1)} \leftarrow x^{(t)}$ otherwise;
  $T_{t+1} \leftarrow T_t \cdot \beta$;

We now consider when the current solution of SA is already a spanning tree. If the current spanning tree is suboptimal, it means that an edge in the tree needs to be replaced with another edge to reduce the total weight. Excluding an existing edge from the current solution would disconnect the graph. To ensure the current solution remains connected, another edge must be included first.

A spanning tree of $G = (V, E)$ has n vertices and $m - 1$ edges. If another edge is included, the total number of edges becomes m which connects two vertices that are already connected via the tree structure, creating a cycle. Therefore, to achieve the desired improvement, SA effectively closes a cycle and then excludes another edge from the cycle in the next time step. The temperature must still be sufficiently high to allow for this inclusion of an edge to be accepted in the solution.

The setup of MA shares the exact same principles except with a fixed temperature parameter T for the entire search process.

One conventional way to analyse the time complexity of a heuristic is to analyse its expected running time. The heuristic is then considered efficient on the problem if its expected running time is a polynomial. However, if there exists a small probability that an important "good event" is missed within a temperature range $[T_1, T_2]$, then for temperatures smaller than T_1 this event might become highly improbable. This could lead to cases where the algorithm has an expected running time that is superpolynomial or even exponential, even if the actual running time is polynomially bounded with overwhelming probability. Therefore, we wish to define our efficiency measure in the following way:

Definition 8 (Efficiency Measure)**.** [Wegener (2005)] Let A be a heuristic running for a

polynomial number of $p(m)$ rounds and let $s(m)$ be the success probability which is the probability that A finds an optimal solution within this phase. A is called

- *successful*, if $s(m) \geq 1/q(m)$ for some polynomial $q(m)$,
- *highly successful*, if $s(m) \geq 1 - 1/q(m)$ for some polynomial $q(m)$, and
- *successful with overwhelming probability*, if $s(m) = 1 - e^{-\Omega(m^\varepsilon)}$ for some $\varepsilon > 0$.

In fact, one cannot hope that randomised search heuristics outperform sophisticated problem-specific algorithms for well-studied problems such as the MST problem. In particular, we do not hope that SA or MA outperforms the famous algorithms due to Kruskal (1956) and Prim (1957). Instead, the main contribution of Wegener (2005) is not to provide a particularly effective algorithm for the MST problem, but to analyse the performance of a popular heuristic applied to a realistic and well-known problem in combinatorial optimisation. This is particularly relevant given that SA remains a widely used heuristic, with no signs of declining interest in practical applications. In fact, at the time of writing this dissertation, the Scopus bibliographic database indexes over 6,500 articles with "Simulated Annealing" in the title, a number that rises to 35,000 when including abstracts and keywords.

Despite SA's continued popularity in practice, its theoretical analysis has received less attention over time, resulting in a growing gap between theory and application. In the early years, efforts were devoted to the theory of SA (see Sasaki and Hajek (1988), Jerrum and Sorkin (1998), Wegener (2005), Jansen and Wegener (2007)); more recently, analysis of SA and MA mostly appear in side results of works focused on other heuristics (see Yang, Sun, Cheng, Bian, Liu, Sun et al. (2022) as an example). One reason is that compared to problem-specific algorithms, heuristics such as SA are not designed specifically to support their analysis.

Moreover, before Wegener (2005), it was a question how to use the freedom to choose a cooling schedule for SA and whether this option is essential. In fact, those problems that could be solved efficiently by simulated annealing could be solved just as effectively by "annealing" at a single carefully selected temperature (MA). This led Jerrum and Sinclair (1996) to conclude that "a rigorous demonstration that annealing is provably beneficial for some natural optimization problems would rate as a significant theoretical advance." In Chapter 5, we will show that SA outperforms MA on instances of the MST problem with the so-called connected triangles as underlying graphs (see Figure 4.1).

Definition 9. The "connected triangles" is a graph $\mathcal{H}_m = (V, E)$ with m edges, where $V = \{v_1, v_2, \ldots, v_{\frac{2m}{3}+1}\}$ and $E = \{(v_{2i-1}, v_{2i}), (v_{2i}, v_{2i+1}), (v_{2i+1}, v_{2i-1}) \mid i = 1, \ldots, \frac{m}{3}\}$. Each set $\{v_{2i-1}, v_{2i}, v_{2i+1}\}$ forms a triangle and edges $\{(v_{2i-1}, v_{2i}), (v_{2i}, v_{2i+1}), (v_{2i+1}, v_{2i-1})\}$ connect consecutive vertices in a triangle. The weight profile (w^1, w^2, w^3) of a triangle is simply the ordered vector of the three edge weights. There are $\frac{m}{3}$ triangles in total; $n = \frac{m}{6}$ triangles with weight profile $(1, 1, m)$ and $n = \frac{m}{6}$ triangles with weight profile (m^2, m^2, m^3). The unique minimum spanning tree consists of all edges of weight 1 or m^2.

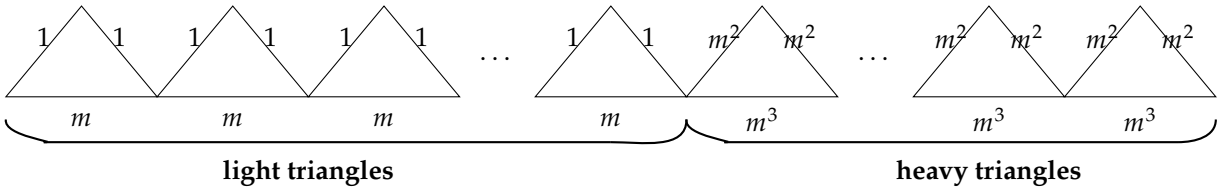

Figure 4.1: Connected triangles $\mathcal{H}_m$

We then identify three states a triangle can occupy during the search phase.

Definition 10. We distinguish each triangle within the search phase of an MST on $\mathcal{H}_m$ with three states. We call a triangle "complete" if the current solution includes all three of its edges; "optimal" if only the two light weight edges are included; "bad" if the heavy weight edge is included and one of the light weight edges is also included.

The idea behind the choice of $\mathcal{H}_m$ is the following. An MST on $\mathcal{H}_m$ consists of an MST on all the $(1, 1, m)$ triangles and an MST on all the (m^2, m^2, m^3) triangles. For the (m^2, m^2, m^3) triangles, MA needs to accept the inclusion of a weight m^2 edge to optimise a bad triangle. This requires a high temperature T^* such that $e^{-\frac{m^2}{T^*}}$ is very large. However, m^2 edges are still much heavier than m edges of $(1, 1, m)$ triangles. At temperature T^*, MA's acceptance probability on the $(1, 1, m)$ triangles is at least $e^{-\frac{m}{T^*}}$ which is very high, so MA will accept any move on the $(1, 1, m)$ triangles. This means that the solutions will not be stable — they are destroyed and reconstructed from time to time. On the other hand, SA is successful due to its feature of a cooling schedule.

Chapter 5

Simulated Annealing beats Metropolis in Combinatorial Optimisation

We first prove two lemmas which we will use to prove that SA beats MA.

Lemma 11 (Chernoff Bound for Binomial Distribution)**.** *Let $X \sim$ Binomial(n, p) and $\mu = \mathbb{E}(X)$. For any $0 < \delta < 1$,*

$$\mathbb{P}\left(X \leq (1-\delta)\mu\right) \leq e^{\left(-\frac{\delta^2\mu}{2}\right)}.$$

Proof. Let $X = \sum_{i=1}^{n} X_i$, where $X_1, X_2, \ldots, X_n$ are i.i.d. random variables with $X_i \sim$ Bernoulli(p). Then $X \sim$ Binomial(n, p) with $\mu = \mathbb{E}(X) = np$. For any $t < 0$, by the Markov inequality,

$$\begin{aligned}
\mathbb{P}(X \leq k) &= \mathbb{P}\left(e^{tX} \geq e^{tk}\right) \\
&\leq \frac{\mathbb{E}(e^{tX})}{e^{tk}} = \frac{M_X(t)}{e^{tk}} \\
&= \frac{\prod_{i=1}^{n} M_{X_i}(t)}{e^{tk}}. \qquad (5.1)
\end{aligned}$$

Since $X_i \sim$ Bernoulli(p), using the inequality $x + 1 \leq e^x$ for all $x \in \mathbb{R}$ with $x = (e^t - 1)p$,

$$\begin{aligned}
M_{X_i}(t) &= \mathbb{E}\left(e^{tX_i}\right) \\
&= e^t \cdot p + 1 \cdot (1-p) \\
&= (e^t - 1) \cdot p + 1 \\
&\leq e^{(e^t-1)p}. \qquad (5.2)
\end{aligned}$$

Substituting (5.2) into (5.1), we have that

$$\begin{aligned}\mathbb{P}(X \leq k) &\leq \frac{\prod_{i=1}^{n} M_{X_i}(t)}{e^{tk}} \leq \frac{\left(e^{(e^t-1)p}\right)^n}{e^{tk}} \\ &= \frac{e^{(e^t-1)np}}{e^{tk}} = \frac{e^{(e^t-1)\mu}}{e^{tk}}. \end{aligned} \tag{5.3}$$

Now let $k = (1-\delta)\mu$, $0 < \delta < 1$. We wish to minimise $f(t) = \frac{e^{(e^t-1)\mu}}{e^{tk}}$ over $t < 0$ to obtain a tight bound for $\mathbb{P}(X \leq k)$. We first take the natural logarithm to simplify the function: $\log f(t) = (e^t - 1)\mu - tk$. We can verify that $t = \log(1-\delta)$ minimises $f(t)$ by checking the first order condition:

$$\frac{d}{dt}\left((e^t - 1)\mu - tk\right) = e^t\mu - k = (1-\delta)\mu - k = 0.$$

Finally, substituting our k and t into (5.3) and using $(1-\delta)^{1-\delta} \geq e^{-\delta+\frac{\delta^2}{2}}$, we have the desired inequality:

$$\begin{aligned}\mathbb{P}(X \leq (1-\delta)\mu) &\leq \frac{e^{\left(e^{\log(1-\delta)}-1\right)\mu}}{e^{(1-\delta)\mu\log(1-\delta)}} = \frac{e^{((1-\delta)-1)\mu}}{(e^{\log(1-\delta)})^{(1-\delta)\mu}} \\ &= \frac{e^{-\delta\mu}}{(1-\delta)^{(1-\delta)\mu}} = \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu} \\ &\leq \left(\frac{e^{-\delta}}{e^{-\delta+\frac{\delta^2}{2}}}\right)^{\mu} = (e^{-\delta+\delta-\frac{\delta^2}{2}})^{\mu} = e^{\left(\frac{-\delta^2\mu}{2}\right)}.\end{aligned}$$

□

Our next lemma provides an upper bound for the probability that a discrete-time stochastic process with bounded upward and downward drift reaches the upper absorbing barrier before the lower one. This will play a central role in proving that MA is unsuccessful on $\mathcal{H}_m$.

Lemma 12. *Let $\{X_t\}_{t\in\mathcal{T}}$ be a discrete-time stochastic process defined on the filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$. The process evolves as $X_{t+1} = X_t + \xi_t$, where the increments ξ_t are $\mathcal{F}_t$-measurable random variables and always result in $X_t \in \{i, i+1, \ldots, n\}$ for all t. When X reaches the absorbing barriers at i (ruin) or n (success), the increments remain 0.*

Conditional on not yet being absorbed, the increments evolve according to the following transition probabilities at each time step t:

- *Upward step:* $\mathbb{P}(\xi_{t+1} = 1 \mid \mathcal{F}_t) = p_t$,

- *Downward step:* $\mathbb{P}(\xi_{t+1} = -1 \mid \mathcal{F}_t) = q_t$,
- *No movement (stay in place):* $\mathbb{P}(\xi_{t+1} = 0 \mid \mathcal{F}_t) = 1 - p_t - q_t$.

The transition probabilities are assumed to satisfy uniform bounds for all t:

$$p_t \le p_{\max}, \quad q_t \ge q_{\min}, \quad p_{\max} < q_{\min},$$

where $0 < p_{\max} \le 1$ *and* $0 \le q_{\min} < 1$.

Suppose the process starts at an initial state $X_0 = k$, $i < k < n$. Let $\tau_k = \min\{t \ge 0 : X_t \in \{i, n\} \mid X_0 = k\}$ be the first time when X gets absorbed conditioning on $X_0 = k$. Let $\mathbb{P}_k = \mathbb{P}(X_{\tau_k} = n)$ denote the probability that the process X hits n before i starting at k. Then, $\mathbb{P}_k$ is bounded above by:

$$\mathbb{P}_k \le \frac{\left(\frac{q_{\min}}{p_{\max}}\right)^{k-i} - 1}{\left(\frac{q_{\min}}{p_{\max}}\right)^{n-i} - 1}$$

Proof. The idea is that we construct a supermartingale to apply the Optional Stopping Theorem.

Let $M_t = \left(\frac{q_{\min}}{p_{\max}}\right)^{X_t}$. Indeed M_t is a supermartingale,

$$\begin{aligned}
\mathbb{E}[M_{t+1} - M_t \mid \mathcal{F}_t] &= p_t[(\frac{q_{min}}{p_{max}})^{X_t+1} - (\frac{q_{min}}{p_{max}})^{X_t}] + q_t[(\frac{q_{min}}{p_{max}})^{X_t-1} - (\frac{q_{min}}{p_{max}})^{X_t}] \\
&= (\frac{q_{min}}{p_{max}})^{X_t}[p_t(\frac{q_{min}}{p_{max}} - 1) + q_t(\frac{p_{max}}{q_{min}} - 1)] \\
&\le (\frac{q_{min}}{p_{max}})^{X_t}[p_{max}(\frac{q_{min}}{p_{max}} - 1) + q_{min}(\frac{p_{max}}{q_{min}} - 1)] = 0,
\end{aligned}$$

since $\frac{p_{max}}{q_{min}} < 1$.

Moreover, M_t is bounded and τ_k is finite almost surely. Applying the Optional Stopping Theorem at τ_k, we obtain:

$$\begin{aligned}
\mathbb{E}[M_{\tau_k}] &\le M_0 \\
\mathbb{P}_k \left(\frac{q_{\min}}{p_{\max}}\right)^n + (1 - \mathbb{P}_k) \left(\frac{q_{\min}}{p_{\max}}\right)^i &\le \left(\frac{q_{\min}}{p_{\max}}\right)^k \\
\mathbb{P}_k &\le \frac{\left(\frac{q_{\min}}{p_{\max}}\right)^{k-i} - 1}{\left(\frac{q_{\min}}{p_{\max}}\right)^{n-i} - 1}.
\end{aligned}$$

□

We now proceed to prove that the Metropolis algorithm is unsuccessful at computing an MST on $\mathcal{H}_m$ for arbitrary fixed temperature T.

Theorem 13 (Wegener (2005))**.** *Let $c = 0.001$. For sufficiently large m, the probability that the Metropolis algorithm applied to $\mathcal{H}_m$ with arbitrary fixed temperature T computes the MST within e^{cm} time steps is bounded above by $e^{-0.001m}$, i.e., MA is unsuccessful on these instances.*

Proof. We distinguish two cases of $T \geq m$ and $T < m$. For the case $T < m$, we will show that optimising the (m^2, m^2, m^3) triangles will result in the failure of MA.

Let A_i be the event that the first edge of a weight (m^2, m^2, m^3) triangle i to be flipped is the m^3 edge. Then, $\bigcap_{i=1}^{n} A_i$ is the event that all (m^2, m^2, m^3) triangles become optimal after flipping only one edge in each triangle. For each triangle, MA always accepts the exclusion of the first flipped edge since the vertices of $\mathcal{H}_m$ remain connected. As MA chooses an edge uniformly at random, the family of events A_i is independent and $\mathbb{P}(A_i) = 1/3$. Therefore, $\mathbb{P}(\bigcap_{i=1}^{n} A_i) = (1/3)^n$.

If the first edge of a weight (m^2, m^2, m^3) triangle to be flipped is not the m^3 edge, then one m^2 edge and one m^3 edge remain. To optimise such triangle, MA needs to first include the missing m^2 edge and remove the heavy m^3 edge in later time steps. If a missing m^2 edge is flipped, the probability that MA accepts this inclusion is $e^{-m^2/T} \leq e^{-m}$. Therefore, let B denote the event that MA accepts the inclusion of an m^2 edge within e^{cm} steps, $c = 0.001$; such event is necessary for MA to optimise a bad triangle. We have that $\mathbb{P}(B) \leq e^{-m} \cdot e^{cm} = e^{-0.999m}$.

For MA to successfully compute an MST on $\mathcal{H}_m$, either $\bigcap_{i=1}^{n} A_i$ happens or MA optimises all non-optimal triangles within e^{cm} steps. By the union bound, the success probability of MA within e^{cm} steps is then at most $\mathbb{P}(\bigcap_{i=1}^{n} A_i) + \mathbb{P}(B) \leq (1/3)^n + e^{-(1-c)m} = (3)^{-m/6} + e^{-0.999m} < e^{\log 2 - \frac{\log 3}{6} m} < e^{-0.001m}$ for $m > \frac{6 \log 2}{\log 3 - 0.006}$.

For the case when $T \geq m$, we will show that optimising only the $(1, 1, m)$ triangles will result in the failure of MA.

For the triangles with weight profile $(1, 1, m)$, let X_t be the number of optimal triangles after time step t. We note that $|X_{t+1} - X_t| \leq 1$ since we can only possibly change the state of at most one triangle at each time step. Let $G_t = (V, E_t)$ be the connected subgraph induced by the current solution of MA applied to $\mathcal{H}_m$ at time step t. Effectively, X_t is a discrete-time stochastic processes defined on the filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$, where $\Omega = \{G_0, G_1, G_2, \dots\}$, $\mathcal{F} = \sigma(G_0, G_1, G_2, \dots)$ and $\mathcal{F}_t = \sigma(X_0, G_1, G_2, \dots, G_t)$.

We are interested in the probability that MA optimises all triangles within e^{cm} time steps. Let $\frac{9}{11}n$ and n be two absorbing barriers. We first claim that there exists a phase ϕ where the process X_t increases from $\frac{10}{11}n$ to n without hitting $\frac{9}{11}n$ first. Suppose for contradiction that this is not the case. Then, every path where X_t increases from $\frac{10}{11}n$ to n must hit $\frac{9}{11}n$ at some point before hitting n. But since $\frac{9}{11}n < \frac{10}{11}n < n$, this would imply that X_t cannot reach n without first being absorbed at $\frac{9}{11}n$, a contradiction.

Thus, within ϕ, $X_t \in \{\frac{9}{11}n, \frac{9}{11}n+1, \ldots, n\}$ for all t and evolves as

$$X_{t+1} = X_t + \Delta_t \quad X_0 = \frac{10}{11}n,$$

where $\{\Delta_t\}$ forms a sequence of random variables distributed as

$$\mathbb{P}(\Delta_t = 1) = p_t, \quad \mathbb{P}(\Delta_t = -1) = q_t, \quad \mathbb{P}(\Delta_t = 0) = 1 - (p_t + q_t),$$

$$p_t \leq p_{\max}, \quad q_t \geq q_{\min}, \quad 0 < p_{\max} \leq 1, \quad 0 \leq q_{\min} < 1.$$

Within ϕ,

$$p_t \leq p_{max} \leq \frac{n - (9/11)n}{m} = \frac{2}{11}\frac{n}{m},$$

since it is necessary for MA to flip the heavy m edge in one of the at most $n - \frac{9}{11}n = \frac{2}{11}n$ complete triangles for an upward step, and

$$q_t \geq q_{min} \geq (\frac{9}{11}n) \cdot \frac{1}{m} \cdot e^{\frac{-m}{T}} \geq (\frac{9}{11}n) \cdot \frac{1}{m} \cdot e^{-1} \geq \frac{3}{11}\frac{n}{m},$$

since it is necessary for MA to flip one heavy m edge and accept its inclusion in one of the at least $\frac{9}{11}n$ optimal triangles for a downward step.

Let $\tau_{\frac{10}{11}n} = \min\{t \geq 0 : X_t \in \{\frac{9}{11}n, n\} \mid X_0 = \frac{10}{11}n\}$ and $\mathbb{P}_{\frac{10}{11}n} = \mathbb{P}(X_{\tau_{\frac{10}{11}n}} = n)$. We wish to apply Lemma 12 to obtain an upper bound for $\mathbb{P}_{\frac{10}{11}n}$. The increments $\{\Delta_t\}$ are determined by the Metropolis criterion and G_t, so indeed $\{\Delta_t\}$ are random variables measurable with respect to $\mathcal{F}_t$.

Moreover, $p_{\max} < q_{\min}$ and $\frac{q_{min}}{p_{max}} \leq \frac{(3/11)(n/m)}{(2/11)(n/m)} = \frac{3}{2}$. By Lemma 12,

$$\begin{aligned}
\mathbb{P}_{\frac{10}{11}n} &\leq \frac{(3/2)^{n/11} - 1}{(3/2)^{2n/11} - 1} \\
&= \frac{(3/2)^{n/11} - 1}{((3/2)^{n/11} - 1) \cdot ((3/2)^{n/11} + 1)} \\
&= \frac{1}{(3/2)^{n/11} + 1} = \exp(-\log((3/2)^{m/66} + 1)) \\
&< \exp(-\frac{m}{66}\log(3/2)) < e^{-0.002m}.
\end{aligned}$$

Finally, since $|X_{t+1} - X_t| \leq 1$, there are at least $n - (\frac{9}{11})n = (\frac{2}{11})n$ time steps for X to reach n from $(\frac{9}{11})n$. For $c = 0.001$, this happens within e^{cm} time steps at most e^{cm} times. Therefore, the probability that MA computes an MST on $\mathcal{H}_m$ is bounded above by $e^{cm} \cdot e^{-0.002m} = e^{-0.001m}$. This completes the proof.

□

Next, we prove that SA with a geometric cooling schedule is highly successful at computing an MST on $\mathcal{H}_m$.

Theorem 14 (Wegener (2005)). *Let k be an arbitrary positive constant, $c = 800k$. Let the cooling schedule be described as $T_t = m^3 \cdot \alpha^t$, where $\alpha = 1 - 1/(cm)$. For sufficiently large m, the probability that simulated annealing applied to $\mathcal{H}_m$ computes the MST within $3cm \log m$ time steps is bounded below by $1 - 4m^{-k}$, i.e., SA is highly successful on these instances.*

Proof. For all arbitrary constants $k > 0$, let $c = 800k$ and $c'' = 170k$. We investigate the search phase of SA until the temperature drops below 1, that is, by time step t such that $T_t = \alpha^t m^3 < 1$. By choosing $t < 3cm \log m$, we can verify that $(1 - \frac{1}{cm})^{3cm \log m} \cdot m^3 \leq \exp((-3cm \log m)/cm) \cdot m^3 = m^{-3} \cdot m^3 = 1$, where the first inequality follows from $(1 - x)^y \leq e^{xy}$, for all $x \geq 0$. So this search phase has at most $3cm \log m$ time steps. Moreover, there are two subphases where the temperature is in $[m^2, m^{5/2}]$ and $[1, m^{1/2}]$; each subphase has at least $(c/4)m \log m = 200km \log m$ time steps.

To compute the MST on $\mathcal{H}_m$ by time step t, we will show that SA will compute an MST on all (m^2, m^2, m^3) triangles during the first subphase when the temperature is high and compute an MST on all $(1, 1, m)$ triangles during the second subphase, while ensuring that optimal (m^2, m^2, m^3) triangles stay optimal after the first subphase.

For the first subphase $T_i \in [m^2, m^{5/2}]$, if an (m^2, m^2, m^3) triangle is optimal, this triangle remains optimal unless SA flips an edge of weight m^3 and accepts its inclusion. This event happens with probability at most $e^{-m^3/T} \leq e^{-m^3/m^{5/2}} = e^{-m^{1/2}}$. So an optimal (m^2, m^2, m^3) triangle will turn to non-optimal within the first subphase with probability bounded above by $200km \log m \cdot e^{-m^{1/2}}$; this is one bad behaviour of SA.

Moreover, SA can be unsuccessful if each edge does not get flipped frequently enough. This is a second bad behaviour of SA. There are at least $(c/4)m \log m$ time steps for $T_i \in [m^2, m^{5/2}]$; at each step an edge is chosen uniformly at random from the total m edges. Hence, let X be the number of times an edge is flipped in $(c/4)m \log m$ steps, $X \sim Binomial((c/4)m \log m, 1/m)$. The expected times an edge is flipped is $\mu = \mathbb{E}(X) = \frac{1}{m} \cdot (c/4)m \log m = (c/4) \log m$.

Let $\delta = 1 - 4\frac{c''}{c}$. We have that $(1 - \delta)\frac{c}{4} \log m = c'' \log m$. By Lemma 11,

$$\mathbb{P}(X \leq (1 - \delta)\mu) \leq \exp(-\frac{\delta^2 \mu}{2})$$
$$\mathbb{P}(X \leq c'' \log m) \leq \exp(-\frac{(1 - 4\frac{c''}{c})^2(\frac{c}{4}) \log m}{2}).$$

For all $k > 0$, $\exp(-\frac{(1-4\frac{c''}{c})^2(\frac{c}{4}) \log m}{2}) = m^{-(1-4\frac{c''}{c})^2(\frac{c}{8})} = m^{-(1-4\cdot\frac{170}{800})^2(\frac{800k}{8})} = m^{-2.25k}$. That is, the probability of not flipping an edge $c'' \log m$ times is bounded above by $m^{-2.25k}$.

If SA does not optimise a non-optimal (m^2, m^2, m^3) triangle for the number of times each edge is flipped within the first subphase, then SA does not compute an MST on all (m^2, m^2, m^3) triangles during the first subphase. This is a third bad behaviour of

SA. Each complete or bad (m^2, m^2, m^3) triangle can turn into an optimal triangle in 2 time steps. In the worst case, this event happens when SA flips the missing m^2 edge first and then flips the m^3 edge consecutively with probability $\frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}$; and SA accepts the inclusion of m^2 edge with probability $e^{-m^2/T_i} \geq e^{-1}$. Thus, the probability of this event not happening among the at least $(c''/2) \log m$ pairs of steps is bounded above by $(1 - \frac{1}{9} \cdot e^{-1})^{\frac{c''}{2} \log m} = m^{\frac{c''}{2} \log(1 - \frac{1}{9} e^{-1})} = m^{\frac{170}{2} k \log(1 - \frac{1}{9} e^{-1})} < m^{-k}$, as these events are independent within one individual triangle. Hence, the probability of not optimising an non-optimal (m^2, m^2, m^3) triangle within the first subphase is bounded above by m^{-k}.

SA fails to compute an MST on $\mathcal{H}_m$ if any of the above mentioned bad behaviour happens. Therefore, by the union bound, the probability that the first subphase does not finish with an MST on all (m^2, m^2, m^3) triangles is at most $200km \log m \cdot e^{-m^{1/2}} + m^{-2.25k} + m^{-k}$. Similarly, for the subphase of $T_j \in [1, m^{1/2}]$, the probability that SA does not compute an MST on all $(1, 1, m)$ triangles is bounded above by $200km \log m \cdot e^{-m^{1/2}} + m^{-2.25k} + m^{-k}$, since we repeat the same calculation with both the temperature and edge weights being scaled down by a factor of m^{-2}.

Finally, an optimal (m^2, m^2, m^3) triangle can turn into non-optimal for all temperature $T_h < m^2$ if SA flips the missing m^3 edge and accepts with probability $\frac{1}{3} e^{-\frac{m^3}{T_h}} < \frac{1}{3} e^{-m}$. After the first subphase, SA will run for at most $(3c/4)m \log m$ steps until T drops below 1. Hence, the probability that an (m^2, m^2, m^3) triangle has turned from optimal into non-optimal after the first subphase is bounded above by $\frac{3}{4} cm \log m \cdot \frac{1}{3} e^{-m} = 100km \log m \cdot e^{-m}$.

Therefore, the overall probability that SA computes the MST on $\mathcal{H}_m$ within $cm \log m$ time steps is bounded below by $1 - (2 \cdot (200km \log m \cdot e^{-m^{1/2}} + m^{-2.25k} + m^{-k}) + 100km \log m \cdot e^{-m}) > 1 - 2 \cdot (m^{-2.25k} + m^{-k}) > 1 - 4m^{-k}$ for sufficiently large m. This completes the proof. □

Chapter 6

Simulated Annealing as a Polynomial-time Approximation Scheme For the MST Problem

Wegener (2005) is well-known for the construction of $\mathcal{H}_m$ where the Metropolis algorithm is unsuccessful in computing an MST, whereas SA with a geometric cooling schedule is successful. Another result in his work is that SA with a suitable geometric cooling schedule can find optimal solutions to the MST problem in polynomial-time when the edge weights of the instance are $(1+\varepsilon)$-separated.

Definition 15 ($(1+\varepsilon)$-separated edge weights)**.** Let $G = (V, E)$ with edge weights $w : E \to \mathbb{Z}_{>0}$. Let $\varepsilon > 0$ be such that for all edges $e_i, e_j \in E$, if $w(e_i) > w(e_j)$, then $w(e_i) \geq (1+\varepsilon)w(e_j)$. We say that the edge weights of G are $(1+\varepsilon)$-separated.

For the remainder of this chapter, we denote by n, m, w_{max} and w_{min} the number of vertices, the number of edges, the maximum and minimum edge weight of the instance of the MST problem respectively.

Theorem 16. *[Wegener (2005)] Let $G = (V, E)$ with $w : E \to \mathbb{Z}_{>0}$ be an instance of the MST problem with $(1+\varepsilon)$-separated edge weights. Assume further that $w(e) \leq 2^m$ for all $e \in E$. Then SA with initial temperature $T_0 = 2^m$ and cooling factor $\beta = (1+\varepsilon/2)^{-m-7-8/\varepsilon}$ with probability $1 - O(1/m)$ finds an optimal solution in at most $2\log_2(1+\varepsilon/2)^{-1}m^{8+8/\varepsilon}$ iterations.*

Wegener (2005) conjectured that his SA algorithm for general edge weights instead of $(1+\varepsilon)$-separated edge weights computes trees with edge weight at most $(1+\varepsilon)$ times the edge weight of a true MST, which we call $(1+\varepsilon)$-approximate MSTs. The main result of Doerr, Rajabi, and Witt (2022) is that Wegener's conjecture is indeed true.

Theorem 17. *[Doerr, Rajabi, and Witt (2022)] Let $\delta < 1$. Consider a run of SA with multiplicative cooling schedule with $\beta = 1 - 1/\ell$ for some $\ell = \omega(mn\log(m/\delta))$ and $T_0 \geq w_{\max}$ on an instance*

of the MST problem. With probability at least $1-\delta$*, at all times*

$$t \geq (\ell/2)\log\left(\frac{\log(4(\ell-1)/\delta)T_0}{w_{\min}}\right)$$

the current solution is a $(1+\kappa)$*-approximation, where*

$$1+\kappa \leq (1+o(1))\cdot\frac{\log(\ell/\delta)}{\log(\ell)-\log(mn\log(m/\delta))}.$$

SA cannot compute $(1+\varepsilon)$-approximate MSTs if $\varepsilon(w)=o(1)$ (see Wegener (2005)), so the result of constant ε in Theorem 17 hits the theoretical limit. Doerr, Rajabi, and Witt (2022) also improved on Wegener (2005)'s time bound. Doerr, Rajabi, and Witt (2022)'s runtime guarantee is roughly

$$O\left((mn\log n)^{1+1/\varepsilon}\log\frac{w_{\max}}{w_{\min}}\right)$$

as opposed to $O(m^{8+8/\varepsilon})$ in Theorem 16.

We will provide an outline of the proof below, consisting of 4 lemmas. Let $\ell > 2$. Consider a run of SA with multiplicative cooling schedule with $\beta = 1-1/\ell$ and $T_0 \geq w_{\max}$ on an instance of the MST problem. The idea is that we compare the edge weights of the invariant current solution to the MST. By "invariant", we mean a time step after when SA does not include any new edges to the current solution with high probability. We would like to first show that edges above a certain weight are no longer included in the current solution after the temperature has dropped sufficiently low. Let $a > 1$ and t_w be the earliest time step when $T_{t_w} \leq w/a$. The following lemma states that the probability of SA accepting weight w edges after time t_w is exponentially small.

Lemma 18. *[Doerr, Rajabi, and Witt (2022)] Let* $\ell > 2, 1 < a \leq \ell-1$ *and for any* $w > 0$*,* t_w *be the earliest point of time when* $T_{t_w} \leq w/a$*. It holds that no new edge of weight at least* w *is included in the solutions after time* t_w *with probability at least*

$$1-\frac{2(\ell-1)}{ae^a}.$$

We note that for $\delta < 1$, $1-\frac{2(\ell-1)}{ae^a}$ is at least $1-\delta/2$ for $a \geq log(4(\ell-1)/\delta)$. We want this probability to be at least $1-\delta$, so we will choose $a = log(4(\ell-1)/\delta)$ from now on.

Lemma 18 describes a time step t_w after which with high probability edges of weight at least w are no longer included. Essentially, such edges can be ignored in the rest of the analysis due to the exponential decay in SA's acceptance probability. Therefore, when the temperature drops below w_{min}/a, SA with high probability does not include any edges into the current solution - this is the time step from when the current solution is invariant. We define t_{end} to be the earliest time step when $T_{t_{end}} \leq w_{min}/a$.

We would like to understand the weight profile of the current solution after t_{end}. Our next lemma deals with this by stating that at the end of $4.21\gamma mn\log(2m^2/\delta)+1$ time steps

starting from t_w, there are no edges of weight at least w left that could be replaced by an edge of weight at most $w/(1+\kappa)$, where κ depends on parameters ℓ, γ and α. We will state the optimal values for parameters ℓ and γ are optimised in Lemma 22.

Lemma 19. *[Doerr, Rajabi, and Witt (2022)]*

Let $\gamma > 1$, $\delta < 1$, $\ell > 2$. Let t_w be the earliest point of time when $T_{t_w} \leq w/a$, and assume that no further edges of weight at least w are added to the solution from time t_w. Let

$$1 + \kappa = \frac{a \exp\left(\gamma \frac{4.21mn \log(2m^2/\delta)}{\ell - 1}\right)}{\log \gamma}.$$

Let n_w be the number of connected components in the subgraph using only edges with weight at most $w/(1+\kappa)$ in G. After time step $t_w + 4.21\gamma mn \log(2m^2/\delta)$, the number of edges in the current solution with weight at least w is at most $n_w - 1$ with probability at least $1 - \delta/(2m)$.

With Lemma 19 in hand now, we can now conclude how the weight profile of the current solution at time step t_{end} compares with an MST. Our next lemma states that there is a bijective relation between the edges of the current solution at time step t_{end} and an MST such that the ratio between the weights of corresponding edges is less than $(1+\kappa)$.

Lemma 20. *[Doerr, Rajabi, and Witt (2022)] Let $\delta < 1$, $\gamma > 1$ and $\ell = \omega(1)$. Let*

$$1 + \kappa = \frac{a \exp\left(\gamma \frac{4.21mn \log(2m^2/\delta)}{\ell - 1}\right)}{\log \gamma}.$$

Assume that $\mathcal{T}^*$ is a minimum spanning tree and $\mathcal{T}'$ is the solution of SA at time t_{end} where $T_{t_{end}} \leq w_{\min}/a$.

For an arbitrary spanning tree $\mathcal{T}$, let $w_{\mathcal{T}} = (w_{\mathcal{T}}(1), \ldots, w_{\mathcal{T}}(n-1))$ be a decreasingly sorted list of the weights on its edges, i.e., $w_{\mathcal{T}}(j) \geq w_{\mathcal{T}}(i)$ for all $1 \leq j \leq i \leq n-1$. With probability at least $1 - \delta$, we have

$$w_{\mathcal{T}^*}(k) \leq w_{\mathcal{T}'}(k) < (1+\kappa) w_{\mathcal{T}^*}(k) \quad \text{for each } k \in [1..n-1].$$

With the above lemmas, our next theorem states a time step t from when the temperature is less than w_{min}/a and the ratio of approximation our SA algorithm can obtain.

Theorem 21. *[Doerr, Rajabi, and Witt (2022)] Let $\delta < 1$, $\gamma > 1$ and $\ell = \omega(1)$. For $a \geq \log(4(\ell-1)/\delta)$, with probability at least $1 - \delta$, at all times*

$$t \geq (\ell/2) \log(aT_0/w_{\min})$$

the current solution is a $(1+\kappa)$-approximation where

$$1 + \kappa = \frac{a \exp\left(\gamma \frac{4.21mn \log(2m^2/\delta)}{\ell - 1}\right)}{\log \gamma}.$$

Proof. Assume $\mathcal{T}^*$ is an MST and $\mathcal{T}'$ is the current solution of SA at time step t_{end}, where $w(\mathcal{T}^*)$ is the total weight of $\mathcal{T}^*$. By Lemma 20, with probability $1-\delta$, we have $w_{\mathcal{T}^*}(k) \leq w_{\mathcal{T}'}(k) < (1+\kappa)w_{\mathcal{T}^*}(k)$ for each $k \in [1..n-1]$. That is,

$$w(\mathcal{T}') = \sum_{i=1}^{n-1} w_{\mathcal{T}'}(i) < (1+\kappa) \sum_{i=1}^{n-1} w_{\mathcal{T}^*}(i) = (1+\kappa)w(\mathcal{T}^*).$$

We now find the time step t_{end} so that the current solution is invariant. We have that t_{end} satisfies

$$\begin{aligned} T_0(1-1/\ell)^{t_{\text{end}}} &= \frac{w_{\min}}{a} \\ (1-1/\ell)^{t_{\text{end}}} &= \frac{w_{\min}}{aT_0} \\ t_{\text{end}} \log(1-1/\ell) &= \log(\frac{w_{\min}}{aT_0}) \\ t_{\text{end}} &= \frac{\log(w_{\min}/(aT_0))}{\log(1-1/\ell)}. \end{aligned}$$

Using the inequality $1 - x/2 \geq e^{-x}$ for $0 \leq x \leq 1$ with $x = 2/\ell$, we have that $\log(1-1/\ell) \geq -2/l$. We can bound t_{end} from above by

$$t_{\text{end}} \leq \frac{\log(w_{\min}/(aT_0))}{-2/\ell} = \frac{-\log((aT_0)/w_{\min})}{-2/\ell} = (\ell/2)\log\left(\frac{aT_0}{w_{\min}}\right).$$

□

The formula for κ from Theorem 21 holds for all $\gamma > 1$. In the following lemma, we state a value for γ which leads to the smallest value for $1+\kappa$. With the help of that, we also give some bounds on $1+\kappa$ considering different cases for ℓ.

Lemma 22. *[Doerr, Rajabi, and Witt (2022)] Let κ be defined as in Theorem 23 and $T_{base} := 4.21mn\log(2m^2/\delta)$. Then the minimum value of κ is achieved by setting $\gamma = \exp\left(W\left(\frac{\ell-1}{T_{base}}\right)\right)$, where W is the Lambert W function. Moreover, if $\ell < eT_{base} + 1$,*

$$1 + \kappa \geq e^{(1/e)-1}a.$$

Otherwise, if $\ell \geq eT_{base} + 1$,

$$1 + \kappa \leq a\frac{\exp\left(\left(\log\frac{\ell-1}{T_{base}}\right)\frac{e}{e-1}\log^{-1}\left(\frac{\ell-1}{T_{base}}\right) - 1\right)}{\log\frac{\ell-1}{T_{base}} - \log(\log\frac{\ell-1}{T_{base}})}.$$

For $\ell = \omega(T_{base})$, the last fraction is $(1+o(1))\frac{a}{\log(\ell-1)-\log(T_{base})}$.

Now we are ready to prove Theorem 17.

Proof of Theorem 17. By Theorem 21, we have

$$1 + \kappa = \frac{a \exp\left(\gamma \frac{T_{\text{base}}}{\ell - 1}\right)}{\log \gamma}.$$

Let $\ell = \omega(T_{\text{base}}) = \omega(mn \log(m/\delta))$. By Lemma 22, we have

$$\begin{aligned}
1 + \kappa &\le (1 + o(1)) \frac{a}{\log(\ell - 1) - \log(T_{\text{base}})} \\
&\le (1 + o(1)) \frac{\log(4(\ell - 1)/\delta)}{\log(\ell - 1) - \log(4.21mn \log(2m^2/\delta))} \\
&= (1 + o(1)) \cdot (1 + o(1)) \frac{\log((\ell - 1)/\delta)}{\log(\ell) - \log(mn \log(m/\delta))} \\
&\le (1 + o(1)) \frac{\log(\ell/\delta)}{\log(\ell) - \log(mn \log(m/\delta))}.
\end{aligned}$$

□

Bibliography

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45(1), 41–51. URL https://doi.org/10.1007/BF00940812.

Doerr, B., A. Rajabi, and C. Witt (2022). 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*, 1381–1389. ACM, New York.

Jansen, T. and I. Wegener (2007). A comparison of simulated annealing with a simple evolutionary algorithm on pseudo-boolean functions of unitation. *Theoretical Computer Science* 386(1–2), 73–93.

Jerrum, M. and A. Sinclair (1996). The markov chain monte carlo method: An approach to approximate counting and integration. In: *Approximation Algorithms for NP-hard Problems*, edited by D. Hochbaum, 482–520. PWS Publishing. Available at http://kobus.ca/teaching/cs645/spring09/ua_only/jerrum_sinclair_96.pdf.

Jerrum, M. and G. B. Sorkin (1998). The metropolis algorithm for graph bisection. *Discrete Applied Mathematics* 82(1–3), 155–175.

Kirkpatrick, S., D. Gelatt, and M. Vecchi (1982). Optimization by simulated annealing. Technical Report RC 9355, IBM Research. Acts of PTRC Summer Annual Meeting.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7(1), 48–50.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21(6), 1087–1092.

Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal* 36(6), 1389–1401. URL https://ieeexplore.ieee.org/document/6773228.

Sasaki, G. H. and B. Hajek (1988). The time complexity of maximum matching by simulated annealing. *Journal of the ACM* 35(2), 387–403.

Wegener, I. (2005). Simulated annealing beats metropolis in combinatorial optimization. In: *Automata, Languages and Programming, ICALP 2005*, 589–601. Springer.

Yang, J., Y. Sun, S. Cheng, K. Bian, Z. Liu, X. Sun, and Y. Cao (2022). A memetic algorithm based on adaptive simulated annealing for community detection. In: *Proceedings of the 5th International Conference on Intelligence Science (ICIS)*, volume 659 of *IFIP Advances in Information and Communication Technology*, 20–28. URL https://link.springer.com/chapter/10.1007/978-3-031-14903-0_3.