



# Simulated Annealing is a Polynomial-Time Approximation Scheme for the Minimum Spanning Tree Problem

Benjamin Doerr<sup>1</sup> · Amirhossein Rajabi<sup>2</sup> · Carsten Witt<sup>2</sup>

Received: 26 November 2022 / Accepted: 8 May 2023 / Published online: 22 July 2023

© The Author(s) 2023

## Abstract

We prove that Simulated Annealing with an appropriate cooling schedule computes arbitrarily tight constant-factor approximations to the minimum spanning tree problem in polynomial time. This result was conjectured by Wegener (Automata, Languages and Programming, ICALP, Berlin, 2005). More precisely, denoting by  $n$ ,  $m$ ,  $w_{\max}$ , and  $w_{\min}$  the number of vertices and edges as well as the maximum and minimum edge weight of the MST instance, we prove that simulated annealing with initial temperature  $T_0 \geq w_{\max}$  and multiplicative cooling schedule with factor  $1 - 1/\ell$ , where  $\ell = \omega(mn \ln(m))$ , with probability at least  $1 - 1/m$  computes in time  $O(\ell(\ln \ln(\ell) + \ln(T_0/w_{\min})))$  a spanning tree with weight at most  $1 + \kappa$  times the optimum weight, where  $1 + \kappa = \frac{(1+o(1))\ln(\ell m)}{\ln(\ell) - \ln(mn \ln(m))}$ . Consequently, for any  $\epsilon > 0$ , we can choose  $\ell$  in such a way that a  $(1 + \epsilon)$ -approximation is found in time  $O((mn \ln(n))^{1+1/\epsilon+o(1)}(\ln \ln n + \ln(T_0/w_{\min})))$  with probability at least  $1 - 1/m$ . In the special case of so-called  $(1 + \epsilon)$ -separated weights, this algorithm computes an optimal solution (again in time  $O((mn \ln(n))^{1+1/\epsilon+o(1)}(\ln \ln n + \ln(T_0/w_{\min})))$ ), which is a significant speed-up over Wegener's runtime guarantee of  $O(m^{8+8/\epsilon})$ . Our tighter upper bound also admits the result that in some situations a hybridization of simulated annealing and the  $(1 + 1)$  EA can lead to stronger runtime guarantees than either algorithm alone.

**Keywords** Simulated annealing · Approximation scheme · Minimum spanning trees · Runtime analysis · Theory

---

✉ Carsten Witt  
cawi@dtu.dk

Benjamin Doerr  
doerr@lix.polytechnique.fr

Amirhossein Rajabi  
amraj@dtu.dk

<sup>1</sup> Laboratoire d'Informatique (LIX), CNRS, École Polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France

<sup>2</sup> DTU Compute, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

## 1 Introduction

The theory of randomized search heuristics, mostly in the last 25 years, has considerably increased our understanding of this class of algorithms. A closer look at this field shows that in the early years, significant efforts were devoted also to simulated annealing (SA) [1–4], whereas more recently these algorithms at most appear in side results of works focused on other heuristics. Due to this decline in attention, the gap between theory and practice, at least as wide in heuristics as in classic algorithms, is even wider for SA.

Since we do not see a reducing interest in SA in practice [5], with this first theoretical work solely devoted to SA after a longer time, we aim at reviving the theoretical analysis of this famous heuristic. To this aim, we revisit a classic problem, namely how SA computes minimum spanning trees (MSTs) [3]. We are, of course, not finally interested in using SA for this purpose – for this several very efficient near-linear time algorithms are known –, but we use this problem to try to understand the working principles of SA.

Wegener’s seminal work [3] is well-known for the construction of an instance of the MST problem where the Metropolis algorithm with any fixed temperature fails badly, but SA with a simple multiplicative cooling schedule computes an optimal solution efficiently. Much less known, but equally interesting is another result in this work, namely that SA with a suitable multiplicative cooling schedule can efficiently find optimal solutions to the MST problem when the edge weights are  $(1 + \epsilon)$ -separated (see the theorem for a definition of this term).

**Theorem 1** ([3]) *Let  $G = (V, E)$  with  $w : E \rightarrow \mathbb{Z}_{>0}$  be an instance of the MST problem. Let  $\epsilon > 0$  be such that for all edges  $e_1, e_2 \in E$ , we have that  $w(e_1) > w(e_2)$  implies  $w(e_1) \geq (1 + \epsilon)w(e_2)$ . 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  $\beta = (1 + \epsilon/2)^{-m^{-7-8/\epsilon}}$  with probability  $1 - O(1/m)$  finds an optimal solution in at most  $2 \log_2(1 + \epsilon/2)^{-1} m^{8+8/\epsilon}$  iterations.*

Wegener [3] conjectured that his SA algorithm for general weights instead of  $(1 + \epsilon)$ -separated ones computes  $(1 + \epsilon)$ -approximate minimum spanning trees, that is, trees with weight at most  $(1 + \epsilon)$  times the weight of a true minimum spanning tree. While this conjecture is very natural, it was never proven.

Our main result is that Wegener’s conjecture is indeed true, even though our proof does not confirm his statement that “it is easy to generalize our result to prove that SA is always highly successful if one is interested in  $(1 + \epsilon)$ -optimal spanning trees.” More precisely, we show the following result (see Theorem 4 for a slightly stronger, but more complicated version of this result). We note that SA cannot compute  $(1 + \epsilon)$ -approximations for sub-constant  $\epsilon$ , see again [3], so in this sense our result is as good as possible.

Let  $\epsilon > 0$  be a constant. Consider a run of SA with cooling factor  $\beta = 1 - 1/\ell$ , where  $\ell = (2mn \ln(m))^{1+1/\epsilon+o(1)}$ , and  $T_0 \geq w_{\max}$  on an instance of the MST problem. Then there is a time  $T^* = O((mn \ln(n))^{1+1/\epsilon+o(1)} (\ln \ln n + \ln(T_0/w_{\min})))$  such that with probability at least  $1 - 1/m$ , at all times  $t \geq T^*$  the current solution is a  $(1 + \epsilon)$ -approximation.

Due to the use of proof methods not available at that time, our time bound is significantly better than Wegener's. To compute a  $(1 + \epsilon)$ -approximation, or to compute an optimal solution when the edge weights are  $(1 + \epsilon)$ -separated (see Theorem 11), our runtime guarantee is roughly  $O((mn \log n)^{1+1/\epsilon} \log \frac{w_{\max}}{w_{\min}})$  as opposed to  $O(m^{8+8/\epsilon})$  in Theorem 1.

Mostly because of a different organization of the proof, our result gives more insights into the influence of the algorithm parameters. Our result only applies to initial temperatures  $T_0$  that are at least the maximum edge weight. This is very natural since with substantially smaller temperatures, the heaviest edge cannot be included in the solution with reasonable probability (this follows right from the definition of the algorithm). It is also not difficult to prove that once the temperature is somewhat below the smallest edge weight, then no new edges will ever enter the solution (see Lemma 6 for the precise statement of this result). This implies that there is no reason to run the algorithm longer than roughly for time  $\log_{1/\beta}(T_0/w_{\min}) = O(\ell \log(T_0/w_{\min}))$ , see Theorem 9 for the details. From the perspective of the algorithm user, this is an interesting insight since it gives an easy termination criterion. Also without understanding the precise influence of the cooling factor  $\beta$  on the approximation quality, this insight motivates to use the algorithm for decreasing values of  $\beta$ , say  $\beta_i = 2^{-i}$ , always until the above-determined time is reached, and follow this procedure until a sufficiently good MST approximation is found.

The remainder of this paper is organized as follows. In Sect. 2, we describe the most relevant previous works. We define SA and the minimum spanning tree problem in Sect. 3. The core of this work is our mathematical runtime analysis in Sect. 4. Afterwards, in Sect. 5, we give the result carried out for the MST problem with  $(1 + \epsilon)$ -separated weights. In Sect. 6, we compare our runtime results for SA with the known performance guarantees for the  $(1 + 1)$  EA and discuss a possible hybridization of the two algorithms. The paper ends with a conclusion and a discussion of possible future works.

*Extensions to conference version.* An extended abstract of this work appeared in [6]. This paper provides full mathematical proofs and includes the new Sect. 6.

## 2 Previous Work

As mentioned in the introduction, there are relatively few runtime analyses for SA as a discrete optimization algorithm, see also the survey [7].

The first such result [1] proves that SA can compute good approximations to the maximum matching problem. A closer look at the result reveals that a constant temperature is used, that is, the SA algorithm is in fact the special case of the Metropolis algorithm. It has to be noted that to obtain a particular approximation quality, the temperature has to be set suitably. In this light, the following result from [8] shows a light advantage for evolutionary algorithms: When running the  $(1 + 1)$  EA with standard mutation rate on this problem, then the expected first time to find a  $(1 + \epsilon)$ -approximation is  $O(m^{2\lceil 1/\epsilon \rceil})$ . Note that in this result, the parameters of the algorithm do not need to be adjusted to the desired approximation rate.

For a different problem, namely the bisection problem, it was shown in [2] that SA, again with constant temperature, can solve certain random instances in quadratic time.

Wegener's above mentioned work [3] on the MST problem was the first to show that for some non-artificial problem, a non-trivial cooling schedule is necessary.

A runtime analysis of the Metropolis algorithm on the classic benchmark ONEMAX was conducted in [4]. Not surprisingly, the ability to accept inferior solutions is not helpful when optimizing this unimodal function. The interesting side of this result, though, is that the Metropolis algorithm is efficient on ONEMAX only for very small temperatures.

A recent study [9] on the deceiving-leading-blocks (DLB) problem shows that here the Metropolis algorithm with a constant temperature has a good performance, beating the known runtime results for evolutionary algorithms by a factor of  $\Theta(n)$ . We note that the DLB problem, just as the MST problem, has many local optima which all can be left by flipping two bits.

As side results of a fundamental analysis of hyper-heuristics, two easy lower bounds on the runtime of the Metropolis algorithm (that is, SA with constant temperature) are proven in [10]: (i) The Metropolis algorithm needs time  $\tilde{\Omega}(n^{d-1/2})$  on cliff functions with constant cliff width  $d$  and super-polynomial time when the cliff width is super-polynomial. (ii) The Metropolis algorithm with a temperature small enough to allow efficient hill-climbing needs exponential time to optimize jump functions.

As part of a broader analysis of single-trajectory search heuristics, it was found that the Metropolis algorithm can optimize all weakly monotonic pseudo-Boolean functions in at most exponential time [11].

Some more results exist on problems designed for demonstrating a particular phenomenon. In [12], a problem called VALLEY is designed that has the property the Metropolis algorithm with any temperature needs at least exponential expected time, whereas SA with a suitable cooling schedule only needs time  $O(n^5 \log n)$ . In [4], examples are constructed where one of  $(1+1)$  EA and SA has a small polynomial runtime and the other has an exponential runtime. Also, a class of functions is constructed where both algorithms have a similar performance despite dealing with the local optimum in a very different manner. In [13], a class of problems with tunable width and depths of a valley of low fitness is proposed. It is proven that the performance of the elitist  $(1+1)$  EA is mostly influenced by the width of the valley, whereas the performance of the Metropolis algorithm and a similar non-elitist algorithm inspired from population genetics is mostly influenced by the depths of the valley.

For evolutionary algorithms, for which the theory is more developed than for SA, there are a larger number of results showing that they can serve as approximation algorithms for optimization problems, including NP-hard problems [14]. However, results describing an approximation scheme where the user can provide a parameter  $\epsilon$  to the evolutionary algorithm to compute a  $(1+\epsilon)$ -approximation are rare; apart from the maximum matching problem mentioned above, we are only aware of related results for parallel  $(1+1)$  EAs,  $(1+1)$  EAs with ageing and simple artificial immune systems on the number partitioning problem [15, 16] and for an evolutionary algorithm on the multi-objective shortest path problem [17]. Evolutionary algorithms that approximate the optimum are also known in the subfield of fixed-parameter tractability. While most of these results prove an approximation within a constant factor or growing slowly with

---

**Algorithm 1** Simulated Annealing (SA) with starting temperature  $T_0$  and cooling factor  $\beta \leq 1$  for the minimization of  $f: \{0, 1\}^n \rightarrow \mathbb{R}$

---

```

Select  $x^{(0)}$  from  $\{0, 1\}^n$ .
for  $t \leftarrow 0, 1, \dots$  do
  Create  $y$  by flipping a bit of  $x^{(t)}$  chosen uniformly at random.
  if  $f(y) \leq f(x^{(t)})$  then
     $x^{(t+1)} \leftarrow y$ .
  else
     $x^{(t+1)} \leftarrow y$  with probability  $e^{(f(x^{(t)})-f(y))/T_t}$  and
     $x^{(t+1)} \leftarrow x^{(t)}$  otherwise.
  end if
   $T_{t+1} := T_t \cdot \beta$ .
end for

```

---

the problem dimension, there are also statements similar to approximation schemes for the vertex cover problem [18]. However, in general it is safe to say that there are only few results in the literature that characterize very simple randomized search heuristics like the  $(1 + 1)$  EA and SA as polynomial-time approximation schemes for classical (non-noisy) combinatorial optimization problems.

Finally, we remark that the classical  $(1 + 1)$  EA and a variant of randomized local search can solve the MST problem in expected pseudo-polynomial time  $O(m^2 \log(nw_{\max}))$  [19]. While SA in general does not solve the problem in expected polynomial time, its time bound to achieve a  $(1 + \epsilon)$ -approximation (see Theorem 4 below) can be smaller than the time bound for the  $(1 + 1)$  EA in certain cases where  $m = \omega(n)$  and  $\epsilon$  is a constant. We will compare SA and  $(1 + 1)$  EA more closely in Sect. 6.

### 3 Preliminaries

We now define the SA algorithm and the MST problem. Also, we state a technical tool our main proof builds on.

*Simulated annealing (SA)* is a simple stochastic hill-climber first proposed as optimization algorithm in [20]. Different from a true hill-climber it may, with small probability, also accept inferior solutions. Working with bit-string representations, we use the classic *bit-flip neighborhoods*, that is, the neighbors of a solution are all other solutions that differ from it in a single bit value. For the acceptance of inferior solutions, we use the widely accepted *Metropolis condition*, that is, a solution with fitness loss  $\delta$  over the current solution is accepted with probability  $e^{-\delta/T}$ , where  $T$  is the current *temperature*. The temperature is usually not taken as constant, but is reduced during the run of the algorithm. This allows the algorithm to accept worsening moves easy in the early stages of the run, whereas later worsening moves are accepted with smaller probability, bringing the algorithm closer to a true hill-climber. The choice of the *cooling schedule* is a critical decision in the design of a SA algorithm. A popular choice, already proposed in [20], is a multiplicative cooling schedule (also called geometric cooling scheme). Here we start with a given temperature  $T_0$  and reduce the temperature by some factor  $\beta$  in each iteration. This common variant of

SA, see Algorithm 1 for the pseudocode, was regarded also in the predecessor work of Wegener [3].

The *minimum spanning tree (MST) problem* is defined as follows. We are given an undirected, connected, weighted graph  $G = (V, E)$ . We denote by  $n$  its number of vertices and by  $m$  its number of edges. Let the set of edges be  $E = \{e_1, \dots, e_m\}$ . The weight of edge  $e_i$ , where  $i \in \{1, \dots, m\}$ , is a positive number  $w_i$ . We write  $w_{\min} := \min\{w_i \mid i \in \{1, \dots, m\}\}$  and  $w_{\max} := \max\{w_i \mid i \in \{1, \dots, m\}\}$  for the minimum and maximum edge weight.

The task in the MST problem is to find a subset  $E' \subseteq E$  such that  $(V, E')$  is a spanning tree of  $G$  having minimal total weight  $w(E') = \sum_{e_i \in E'} w_i$ . We use the natural bit-string representation for sets  $E'$  of edges, that is, a bit string  $x = (x_1, \dots, x_m) \in \{0, 1\}^m$  represents the set  $E(x) = \{e_i \mid x_i = 1\}$ . As objective function, we use the sum of the weights of the selected edges when these form a connected graph on  $V$  and  $\infty$  otherwise:

$$f(x) = \begin{cases} w_1x_1 + \dots + w_mx_m & \text{if } (V, E(x)) \text{ is connected,} \\ \infty & \text{otherwise.} \end{cases}$$

Here  $\infty$  can be replaced by an extremely large value without essentially changing the result. To ensure that we start with a feasible solution (one that has finite objective value), we assume that SA is initialized with the all-ones string  $x^{(0)} = (1, \dots, 1)$ . From this initial string, SA can move to solutions having fewer edges by flipping one-bits; however, it will never accept solutions that are not connected due to their infinitely high  $f$ -value. We note that, similarly to the analysis of the  $(1 + 1)$  EA on the MST problem [19], one could use a more involved fitness function to penalize connected components and thus lead the algorithm towards connected subgraphs when the current solution is not connected. However, since we assume SA to start from a connected solution and connected solutions will not be replaced with disconnected solutions with the present definition of  $f$ , this would not provide new insights. Overall, our setup is the same as the one used by Wegener [3].

When the temperature has become sufficiently low, it is likely that SA has reached a solution describing a spanning tree. If this spanning tree is suboptimal, improvements require a change of at least 2 bits. Since SA only flips one bit per iteration, this is only possible by temporarily including one more edge, i. e., closing a cycle, and then removing another edge from the cycle in the next iteration. This requires a temperature still being sufficiently high for the temporary inclusion to be accepted.

Our measure of complexity is the first hitting time  $T^*$  for a certain set of solutions  $S^*$ , e. g., globally optimal solutions or solutions satisfying a certain approximation guarantee with respect to the set of global optima. That is, we give bounds on the smallest  $t$  such that SA has found a solution in  $S^*$ . Due to the probabilistic nature of the algorithm, we will usually give bounds that hold with high probability, e. g., with probability  $1 - 1/n$ . The expected value of  $T^*$  may be undefined since the cooling schedule may make it less and less likely to hit the set  $S^*$  when the algorithm has been unsuccessful during the steps where a promising temperature held. This is different from the analysis of, e. g., simple evolutionary algorithms, where one often considers the

so-called *runtime* as the first hitting time of the set of optimal solutions and bounds the expected runtime. However, as described in detail by Wegener [3], there are simple restart schemes for SA that guarantee expected polynomial optimization times if there is a sufficiently high probability of a single run being successful in polynomial time.

The proof of our main result uses multiplicative drift analysis as a state-of-the-art technical tool, which was not available to Wegener [3]. The multiplicative drift theorem in Theorem 2 below goes back to [21] and was enhanced with tail bounds in [22]. We give a slightly generalized presentation that can be found in [23].

**Theorem 2** (*Multiplicative Drift, cf. [21–23]*) *Let  $(X_t)_{t \geq 0}$ , be a stochastic process, adapted to a filtration  $\mathcal{F}_t$ , over a state space  $S \subseteq \{0\} \cup [s_{\min}, s_{\max}]$ , where  $s_{\min} > 0$  and  $\{0\} \in S$ . Suppose that there exists a  $\delta > 0$  such that for all  $t \geq 0$ , we have*

$$E[X_t - X_{t+1} \mid \mathcal{F}_t] \geq \delta X_t.$$

*Then the first hitting time  $T := \min\{t \mid X_t = 0\}$  satisfies*

$$E[T \mid \mathcal{F}_0] \leq \frac{\ln(X_0/s_{\min}) + 1}{\delta}.$$

*Moreover,  $\Pr(T > (\ln(X_0/s_{\min}) + r)/\delta) \leq e^{-r}$  for any  $r > 0$ .*

## 4 SA as Approximation Scheme for the Minimum Spanning Tree Problem

In this section, we prove our main results on how well SA computes approximate solutions for the MST problem. These results easily imply improved bounds for the previously regarded special case of  $(1 + \epsilon)$ -separated instances, see Sect. 5.

### 4.1 Main Results and Proof Outline

As outlined above in the introduction, this paper revisits Wegener's [3] analysis of SA on the MST problem. Our main result is Theorem 3 below, proving that SA is a polynomial-time approximation scheme for the MST problem as originally conjectured by Wegener. The statement of our main theorem describes the approximation quality and the required time to reach it as a function of the cooling factor, the desired success probability and of course the instance parameters. Theorem 4 takes the dual perspective of computing cooling schedules and running times that allow SA to find a  $(1 + \epsilon)$ -approximation for a given  $\epsilon$  with high probability.

We now present the main theorem and a variant of it, corresponding to the two perspectives mentioned above for analyzing the approximation quality.

**Theorem 3** *Let  $\delta < 1$ . Consider a run of SA with multiplicative cooling schedule with  $\beta = 1 - 1/\ell$  for some  $\ell = \omega(mn \ln(m/\delta))$  and  $T_0 \geq w_{\max}$  on an instance of the MST problem. With probability at least  $1 - \delta$ , at all times  $t \geq (\ell/2) \ln\left(\frac{\ln(4(\ell-1)/\delta)T_0}{w_{\min}}\right)$  the current solution is a  $(1 + \kappa)$ -approximation, where*

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

**Theorem 4** *Let  $\delta = \omega(1/(mn \ln n))$  and  $\delta < 1$ ,  $\epsilon > 0$ . Consider a run of SA with  $\beta = 1 - 1/\ell$  for  $\ell = (mn \ln(m/\delta))^{1+1/\epsilon}$  and  $T_0 \geq w_{\max}$  on an instance of the MST problem. With probability at least  $1 - \delta$ , at all times  $t \geq (\ell/2) \ln \left( \frac{\ln(4(\ell-1)/\delta)T_0}{w_{\min}} \right)$  the current solution is a  $(1 + o(1))(1 + \epsilon)$ -approximation.*

The last theorem is stated in somewhat weaker, but simpler form in the following corollary. In particular, it gives a concrete time bound until SA has computed a  $(1 + \epsilon)$ -approximation with probability at least  $1 - \delta$ , where  $\delta$  and  $\epsilon$  are chosen by the user.

**Corollary 5** *Let  $\epsilon > 0$  be a constant,  $\delta = \omega(1/(mn \ln n))$  and  $\delta < 1$ . Consider a run of SA with  $\beta = 1 - 1/\ell$ , where*

$$\ell = (mn \ln(m/\delta))^{1+1/\epsilon+o(1)},$$

*and  $T_0 \geq w_{\max}$  on an instance of the MST problem. With probability at least  $1 - \delta$ , at all times  $t \geq T^* := (\ell/2) \ln \left( \frac{\ln(4(\ell-1)/\delta)T_0}{w_{\min}} \right)$  the current solution is a  $(1 + \epsilon)$ -approximation. Moreover,*

$$T^* = O((mn \ln(n))^{1+1/\epsilon+o(1)} (\ln \ln n + \ln(T_0/w_{\min}))).$$

The idea of the proof of all results formulated above is to consider phases in the optimization process, concentrating on different intervals for the edge weights, with the size and center of the intervals decreasing over time. In each phase, the number of edges chosen from such an interval will achieve some close-to-optimal value with high probability. After the end of the phase, the temperature of SA is so low that basically no more changes occur to the edges with weights in the interval.

In more detail, the proofs of Theorem 3 and its variant are composed of several lemmas. We are now going to outline the main ideas of these lemmas and how they relate to each other in the roadmap of the final proof.

It is useful to formulate the main results in terms of a cooling factor  $\beta = 1 - 1/\ell$  for some  $\ell > 1$  since  $\ell$  carries the intuition of a “half-life” for the temperature; more precisely, after  $\ell$  iterations of SA the temperature has decreased by the constant factor of  $(1 - 1/\ell)^\ell \approx e^{-1}$ . Lemma 6 is (on top of the usual graph parameters and the starting temperature) based on  $\ell$ , a weight  $w$  and some parameter  $a$ . Intuitively, it describes a point of time  $t_w$  after which edges of weight at least  $w$  are no longer flipped in with high probability and can be ignored for the rest of the analysis due to an exponential decay in the probability of accepting search points of higher  $f$ -value. This probability depends on the parameter  $a$  which will be optimized later in the composition of the main proof.

While Lemma 6 will be used to show that edges above a certain weight are no longer included in the current solution after the temperature has dropped sufficiently, Lemma 7, which is the main lemma in our analysis, deals with the structure of the current solution after edges of a certain weight  $w$  are no longer included. It considers



connected components that can be spanned by cheaper edges and states that these connected components are essentially connected in an optimal way in the whole solution up to multiplicative deviations of a factor  $(1 + \kappa)$  in the weights of the connecting edges. Lemma 7 uses careful edge exchange arguments in its proof and bounds the time to do these exchanges in a multiplicative drift analysis. Moreover, it features another parameter called  $\gamma$  that will be optimized later along with the above-mentioned  $a$ .

Lemma 8 puts together the previous two lemmas to consider the run of SA over up to  $n$  phases depending on the weight spectrum of the graph until the temperature has dropped to a value being so small that no more changes are accepted. This will be the final solution considered in the main proof. Essentially, having listed the weights of an MST decreasingly, the lemma will match the weights of the final solution to the weights of the MST and show for each element in the list that the final solution matches the weight of the element up to a factor  $1 + \kappa$ . Its proof uses a bijection argument proved by induction to apply Lemma 7 and is crucially different from Wegener's analysis.

The final lemma, Lemma 10, finds choices for the parameter  $\gamma$  to minimize the bound  $1 + \kappa$  on the approximation ratio. Its proof uses several results from calculus. Afterwards, Theorem 3 also chooses the parameter  $a$  carefully and arrives at the first statement on the approximation ratio depending on  $\ell$ , the desired success probability  $1 - \delta$ , and the graph parameters, only. The second main theorem, Theorem 4 then essentially translates parameters into each other to compute  $\ell$  and to express time bounds based on the desired  $\epsilon$ . A weaker but simpler formulation of that theorem is finally stated in Corollary 5.

## 4.2 Detailed Technical Analysis

In this subsection, we collect the technical lemmas and theorems outlined above.

Let  $a > 1$  and, for  $w > 0$ , let  $t_w$  be the earliest point of time when  $T(t_w) \leq w/a$ . In the following lemma, we state that the probability that SA accepts edges of weight  $w$  after  $t_w$  is exponentially small with respect to  $a$ . It shows that after the temperature becomes less than  $w$ , the probability of accepting such an edge is sharply decreasing.

**Lemma 6** *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. 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*

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

which is at least  $1 - \delta/2$  for  $\delta < 1$ , if we set  $a \geq \ln(4(\ell - 1)/\delta)$ .

**Proof** Let  $s$  be an edge of weight at least  $w$ , which is not in the solution at the beginning of the step  $t_w$ . Let  $t \in \mathbb{N}_{\geq 0}$  and  $E_s^{(t_w+t)}$  be the event of accepting the edge  $s$  at step  $t_w+t$ . This event happens if the edge  $s$  is flipped with probability  $1/m$  and the algorithm accepts this worse solution. Thus

$$\Pr[E_s^{(t_w)}] = m^{-1} \cdot \exp\left(\frac{-w}{T(t_w)}\right) \leq e^{-a}/m.$$

For all integers  $t \geq 0$ , we have  $T(t_w + t) = T(t_w)(1 - 1/\ell)^t$ . Then

$$\begin{aligned} \Pr[E_s^{(t_w+t)}] &= m^{-1} \exp\left(\frac{-w}{T(t_w)(1 - 1/\ell)^t}\right) \\ &\leq m^{-1} e^{-a(1 + \frac{1}{\ell-1})^t} \\ &\leq m^{-1} e^{-a(1 + \frac{t}{\ell-1})}, \end{aligned}$$

where we used the inequality  $(1 + x)^t \geq 1 + rx$  for  $x > -1$  and  $r \in \mathbb{N}_{\geq 0}$ .

Let  $E_s^{\geq t_w}$  be the event of accepting the edge  $e$  of weight at least  $w$  after step  $t_w$  at least once. Then, using a union bound and the geometric series sum formula, we get

$$\begin{aligned} \Pr[E_s^{\geq t_w}] &\leq \sum_{t=0}^{\infty} \Pr[E_e^{(t_w+t)}] \leq \sum_{t=0}^{\infty} m^{-1} e^{-a(1 + \frac{t}{\ell-1})} \\ &= m^{-1} \frac{e^{-a}}{1 - e^{-a/(\ell-1)}} \leq m^{-1} \frac{e^{-a}}{1 - (1 - \frac{a}{2(\ell-1)})} \\ &= m^{-1} \frac{2(\ell - 1)}{ae^a}, \end{aligned}$$

where we have  $a \leq \ell - 1$  and use the inequality  $e^{-x} \leq 1 - x/2$  for  $0 \leq x \leq 1$ .

Since there are  $m$  edges, with probability  $1 - \frac{2(\ell-1)}{ae^a}$ , there is no inclusion of edges after their corresponding steps  $t_w$ .

Moreover, if we set  $a \geq \ln(4(\ell - 1)/\delta)$ , the probability is at least

$$1 - \frac{2(\ell - 1)}{\ln(4(\ell - 1)/\delta) \cdot 4(\ell - 1)/\delta} = 1 - \frac{\delta/2}{\ln(4(\ell - 1)/\delta)} \geq 1 - \delta/2,$$

where we have  $\ell > 2$  and  $\delta < 1$ . □

In the following lemma, we consider a time interval of length  $4.21\gamma mn \ln(2m^2/\delta) + 1$  starting from  $t_w$  (for fixed  $a$ ) and prove that at the end of this period, 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  $\kappa$  depends on the algorithm parameter  $\ell$  and parameters  $\gamma$  and  $a$ . We optimize these parameters later in this paper.

**Lemma 7** *Let  $\gamma > 1$ ,  $\delta < 1$ ,  $\ell > 2$ ,  $a > 1$ . 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. 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*

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

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  $t_w + 4.21\gamma mn \ln(2m^2/\delta)$ , the number of edges in the current solution with weight at least  $w$  is at most  $n_w - 1$  with probability at least  $1 - \delta/(2m)$ .

**Proof** Let  $T_{base} = 4.21mn \ln(2m^2/\delta)$ . We analyze the steps  $t_w, \dots, t_w + \gamma T_{base}$ . The temperature during this phase is at least

$$T(t_w)(1 - 1/\ell)^{\gamma T_{base}} \geq T(t_w)e^{-(\gamma T_{base})/(\ell-1)},$$

using  $1 - x \geq e^{-x/(1-x)}$  for  $x \leq 1$ , so the probability to accept a chosen edge with weight at most  $w/(1 + \kappa)$  in one step is bounded from below by

$$\exp\left(\frac{-w/(1 + \kappa)}{T(t_w)e^{-\gamma T_{base}/(\ell-1)}}\right) = \exp\left(-\frac{ae^{\gamma T_{base}/(\ell-1)}}{(1 + \kappa)}\right) = 1/\gamma$$

during this phase. By our assumption in the statement, we do not include edges of weight at least  $w$ .

Let us partition the set of edges with weight at least  $w$  in the current solution  $x$ , that is, the graph  $G_x = (V, E(x))$ , into three disjoint subsets. An edge  $e = \{u, v\}$  with weight at least  $w$  has one of the following three properties,

- the edge  $e$  lies on a cycle in  $G_x$ ;
- the edge  $e$  does not lie on a cycle, but there is at least one edge  $e' \in E \setminus E(x)$  with weight at most  $w/(1 + \kappa)$  such that  $e$  lies on a cycle in the graph  $(V, E(x) \cup \{e'\})$ ;
- the edge  $e$  has neither of the two properties. In this case, we call this edge essential for the current and forthcoming solutions.

As long as an edge with weight at least  $w$  is not essential, it can either be removed from the current solution or become an essential edge. When the edge disappears, since its weight is at least  $w$ , it will not appear again.

Also, when the edge becomes essential, it remains essential in the solution to the end, because in order to create a cycle containing this edge, an edge with weight at least  $w$  has to appear, which does not happen, and also removing this edge makes the graph unconnected.

We claim that the number of essential edges does not exceed  $n_w - 1$ . In order to prove this, we define the graph  $H = (V_H, E_H)$  as follows. There is a vertex in  $V_H$  for each connected component of the induced subgraph on the edges of weight at most  $w/(1 + \kappa)$  in  $G$ , and there is an edge between two vertices  $v_i, v_j \in V_H$  if there is an essential edge  $e = \{u, v\}$  in the solution that  $u$  and  $v$  belong to the corresponding connected components  $C_i$  and  $C_j$  respectively. Formally, let  $C = \{C_1, \dots, C_{n_w}\}$  be the connected components of the induced subgraph on the edges of weight at most  $w/(1 + \kappa)$ . Then,  $V_H = \{v_1, \dots, v_{n_w}\}$  and

$$E_H = \{\{i, j\} \mid \exists \text{ essential } e = \{u, v\}, u \in C_i, v \in C_j\}.$$

We claim that there is no essential edge with both endpoints in the same  $C_i$ . To prove this, we assume for contradiction that there is such an edge  $e = \{u, v\}$ . Then, since  $e$

is essential, it cannot be on a cycle in the current solution. Let  $S_u$  and  $S_v$  denote the sets of vertices connected to  $u$  and  $v$  respectively using edges in the solution except  $e$ .  $S_u \cup S_v = V(G)$  because the solution is always connected. Since  $e$  is essential, there is no edge with weight at most  $w/(1 + \kappa)$  in  $G$  from  $S_u$  to  $S_v$  (see the property (2)), so there is no such cheap edge in  $G$  from  $S_u \cap C_i$  to  $S_v \cap C_i$ , which results in that there is a partition of vertices of  $C_i$  that are disconnected in the subgraph using only edges with weight at most  $w/(1 + \kappa)$  in  $G$ , which contradicts the definition of  $C_i$ . Also,  $H$  has to be a forest since we also know that essential edges are not on a cycle. Therefore, since there are  $n_w$  connected components, there are at most  $n_w - 1$  essential edges.

Now, in the next paragraphs, we state the number of steps needed to remove edges with weight at least  $w$  or to make them essential. We consider some epochs consisting of  $2m$  iterations each and let  $X_t$  be the random variable denoting the number of non-essential edges with weight at least  $w$  whose exclusion is possible at epoch  $t$ . We claim that

$$\Delta_t(s) := E[X_t - X_{t+1} \mid X_t = s] \geq s \cdot (1 - e^{-3})n^{-1}/(2\gamma).$$

If no cycle with a non-essential edge  $e = \{u, v\}$  with weight at least  $w$  exists, the probability of creating such a cycle by adding the cheap edge considered in Case 2 between  $S_u$  and  $S_v$  in each step is at least  $1/(\gamma m)$  and in  $m$  steps, is at least

$$1 - \left(1 - \frac{1}{\gamma m}\right)^m \geq 1 - e^{-1/\gamma} \geq 1/(2\gamma),$$

where we have  $1 + x \leq e^x$  for all  $x \in \mathbb{R}$  and the inequality  $e^{-x} \leq 1 - x/2$  for  $0 \leq x \leq 1$ .

Then, after the cycle is created in the first  $m$  iterations, or the cycle already existed, the probability of the exclusion of such an edge in  $m$  steps of the second half of the epoch is only  $(1 - e^{-3})n^{-1}$  because the probability of observing at least one edge from the cycle of length  $k$  in  $m$  steps is  $1 - (1 - k/m)^m \geq 1 - (1 - 3/m)^m \geq 1 - e^{-3}$ , and the probability that the edge selected is  $e$  equals  $1/n$ . Altogether, the probability of excluding a non-essential edge with weight at least  $w$  is at least  $(1 - e^{-3})n^{-1}/(2\gamma)$ , which results in decreasing  $X_t$  by at least one because removing  $e$  might also make some other edges essential. Since there are  $s$  non-essential edges, we have  $\Delta_t(s) \geq s \cdot (1 - e^{-3})n^{-1}/(2\gamma)$ . Since there can be at most  $m$  essential edges at the beginning, we have  $X_0 \leq m$ . Assume  $Y$  denotes the number of epochs needed to have only essential edges with weight at least  $w$ . Using the upper tail bound of multiplicative drift in Theorem 2, we have

$$\Pr\left[Y > \frac{\ln(2m/\delta) + \ln X_0}{(1 - e^{-3})n^{-1}/(2\gamma)}\right] \leq e^{-\ln(2m/\delta)} = \delta/(2m).$$

Since each epoch consists of  $2m$  iterations,

$$2m \cdot 2(1 - e^{-3})^{-1}n\gamma \ln(2m^2/\delta) \leq 4.21mn \ln(2m^2/\delta)$$

is sufficient to arrive at a solution where all edges of weight at least  $w$  are essential.  $\square$

SA does with high probability not accept an inclusion of any edge using Lemma 6 when the temperature is colder than  $w_{\min}/a$  for some  $a$  that is still a parameter chosen later. This is the time from when the solution is invariant. Let  $t_{w_{\min}}$  be the earliest time when  $T(t_{w_{\min}}) \leq w_{\min}/a$  and  $t_{\text{end}} := t_{w_{\min}}$ .

In the following lemma, we show that there is a bijective relation between the edges of the solution at time  $t_{\text{end}}$  and a MST such that the ratio between the weights of corresponding edges is less than  $(1 + \kappa)$ .

**Lemma 8** *Let  $\delta < 1$ ,  $\gamma > 1$ ,  $\ell = \omega(1)$  and  $a \geq \ln(4(\ell - 1)/\delta)$ . Let*

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

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

*For an arbitrary spanning tree  $\mathcal{T}$ , let  $w_{\mathcal{T}} = (w_{\mathcal{T}}(1), \dots, w_{\mathcal{T}}(n - 1))$  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*

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

**Proof** We recall that  $t_w$  is the earliest point of time when  $T(t_w) \leq w/a$ . With probability  $1 - \delta/2$ , edges of weight  $w$  are not included after their corresponding times  $t_w$  via Lemma 6. Thus conditional on this event, we can use Lemma 7 stating that with probability at least  $1 - \delta/(2m)$ , the number of edges with weight at least  $w$  is at most  $n_w - 1$ . This condition must hold for at most  $m$  distinct values, happening with probability at least  $1 - \delta/2$  according to a union bound. Altogether, since the event in Lemma 6 must happen with probability  $1 - \delta/2$  and the condition in Lemma 7 must hold for all weights, with probability at least  $1 - \delta$ , the statement in Lemma 7 is valid for all possible weights.

We use induction on the index  $k$ . The case  $k = 0$  is trivial as the basic step. Regarding the inductive step, assume that for all  $0 \leq k \leq i - 1$ , the inequality is valid. If  $i = n$ , the claim is proved. Otherwise, let  $w_{\mathcal{T}^*}(i)$  be the next unique largest weight and  $j \geq i$  be the largest index that  $w_{\mathcal{T}^*}(j) = w_{\mathcal{T}^*}(i)$ . In fact, we have

$$w_{\mathcal{T}^*}(i - 1) > w_{\mathcal{T}^*}(i) = \dots = w_{\mathcal{T}^*}(j) > w_{\mathcal{T}^*}(j + 1).$$

There are exactly  $j - i + 1$  edges with weight  $w_{\mathcal{T}^*}(i)$  in the minimum spanning tree  $\mathcal{T}^*$ . The number of connected components in  $G$  using only edges with weight at most  $w_{\mathcal{T}^*}(i)$  is  $i$  since they are connected using  $i - 1$  edges in  $\mathcal{T}^*$ . Using Lemma 7 with  $w = (1 + \kappa)w_{\mathcal{T}^*}(i)$  and considering  $n_w = i$ , there are at most  $i - 1$  edges with weight at least  $(1 + \kappa)w_{\mathcal{T}^*}(i)$  in  $\mathcal{T}'$ , which means that the rest of the weight values in  $\mathcal{T}'$  are less than  $(1 + \kappa)w_{\mathcal{T}^*}(i)$ . Since we know that the graph cannot be connected using less than  $j$  edges with weight at least  $w_{\mathcal{T}^*}(i)$ , we can conclude that there are at least

$j$  edges with weight between  $w_{\mathcal{T}^*}(i)$  and  $(1 + \kappa)w_{\mathcal{T}^*}(i)$ . Therefore, for  $i \leq k \leq j$ , the inequality suggested above holds.  $\square$

With the above lemmas at hand, we can prove the first theorem. Given  $\ell$ , Theorem 9 states the approximation ratio that the algorithm with cooling schedule  $\beta = 1 - 1/\ell$  can obtain.

**Theorem 9** *Let  $\delta < 1$ ,  $\gamma > 1$  and  $\ell = \omega(1)$ . 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. For  $a \geq \ln(4(\ell - 1)/\delta)$ , with probability at least  $1 - \delta$ , at all times  $t \geq (\ell/2) \ln(aT_0/w_{\min})$  the current solution is a  $(1 + \kappa)$ -approximation where*

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

**Proof** We consider the time  $t_{\text{end}}$  when  $T(t_{\text{end}}) \leq w_{\min}/a$  and show the approximation result for the current solution of SA at that time. Concretely, assume that  $\mathcal{T}^*$  is a minimum spanning tree and  $\mathcal{T}'$  is the solution of the algorithm at time  $t_{\text{end}}$ . Assume  $w(\mathcal{T})$  is the total weight of edges in the tree  $\mathcal{T}$ . Using Lemma 8, with probability  $1 - \delta$ , we have  $w_{\mathcal{T}'}(k) < (1 + \kappa)w_{\mathcal{T}^*}(k)$  for each  $k \in [1..n - 1]$ . Thus, we have

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

To complete the proof, we only have to find the time  $t_{\text{end}}$  from when the temperature is less than  $w_{\min}/a$ , so after that, no edges are included anymore via Lemma 6. Then  $t_{\text{end}}$  satisfies

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

Then

$$t_{\text{end}} = \log_{1-1/\ell} ((w_{\min}/a)/T_0) = \frac{\ln(w_{\min}/(aT_0))}{\ln(1 - 1/\ell)}.$$

Using the inequality  $1 - x/2 \geq e^{-x}$  for  $0 \leq x \leq 1$  with  $x = 2/\ell$ , we can bound  $t_{\text{end}}$  from above by

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

$\square$

The formula for  $\kappa$ , which we obtained in Theorem 9, holds for all  $\gamma > 1$ . In the following lemma, we suggest a value for  $\gamma$ , leading to the smallest value for  $1 + \kappa$ . With the help of that, we give also some bounds on  $1 + \kappa$  considering different cases for  $\ell$ .

**Lemma 10** *Let  $a$  and  $\kappa$  be defined as in Theorem 9 and  $T_{base} := 4.21mn \ln(2m^2/\delta)$ . Then the minimum value of  $\kappa$  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  $\ell \geq eT_{base} + 1$ ,*

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

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

**Proof** From the definition of  $\kappa$  in Theorem 9, for  $\gamma > 1$ , we have

$$1 + \kappa = a \frac{e^{\gamma/b}}{\ln \gamma}, \tag{1}$$

where  $b := \frac{\ell-1}{T_{base}}$ .

Let  $f(x) = e^{x/b} / \ln x$  for  $x > 1$ . Then its derivative is  $f'(x) = \frac{e^{x/b}}{b \ln(x)} - \frac{e^{x/b}}{x \ln^2(x)}$ . For  $x > 1$ , we have the only root  $x = e^{W(b)}$ , where  $W$  is the Lambert  $W$  function. Therefore, Eq. (1) with  $\gamma = e^{W(b)}$  gives us the minimum value for  $(1 + \kappa)$  and equals

$$a \frac{e^{e^{W(b)}/b}}{W(b)}. \tag{2}$$

Now, we aim at finding some bounds on  $1 + \kappa$ . We analyze Eq. (2) for two cases of  $b$ .

For  $b \geq e$ , using the inequality

$$\ln b - \ln \ln b + \frac{\ln \ln b}{2 \ln b} \leq W(b) \leq \ln b - \ln \ln b + \frac{e}{e-1} \frac{\ln \ln b}{\ln b},$$

from [24], we get

$$\begin{aligned} a \frac{e^{e^{W(b)}/b}}{W(b)} &\leq a \frac{\exp\left(b^{-1} e^{\ln(b)} e^{-\ln \ln b} e^{\frac{e}{e-1} \frac{\ln \ln b}{\ln b}}\right)}{\ln(b) - \ln \ln(b)} \\ &= a \frac{\exp\left(e^{-\ln \ln b} e^{\frac{e}{e-1} \frac{\ln \ln b}{\ln b}}\right)}{\ln(b) - \ln \ln(b)} \\ &= a \frac{\exp\left((\ln b)^{-1 + \frac{e}{(e-1)\ln b}}\right)}{\ln(b) - \ln \ln(b)}. \end{aligned}$$

For  $b = \omega(1)$ , the last expression equals  $\frac{a(1+o(1))}{\ln b - \ln \ln b} = (1 + o(1))\frac{a}{\ln b}$  since

$$(\ln b)^{-1 + \frac{e}{(e-1)\ln b}} = \frac{e^{\frac{e \ln \ln b}{(e-1)\ln b}}}{\ln b} = \frac{o(1)}{\ln b} = o(1).$$

Regarding the case  $b < e$ , using the definition  $W(x)e^{W(x)} = x$ , we have  $e^{W(x)} = \frac{x}{W(x)}$ . By applying these inequalities on Eq. (2), we obtain

$$a \frac{e^{e^{W(b)}/b}}{W(b)} = a \frac{e^{\left(\frac{b}{bW(b)}\right)}}{W(b)} = a \frac{e^{1/W(b)}}{W(b)}.$$

From the definition again, we have  $W(b)e^{W(b)} = b$ . Since for  $x \geq 0$ , we have  $e^x \geq 1$ , we can conclude  $W(b) \leq b$ , resulting in  $W(b) < e$ . Thus the last expression can be bounded from below by

$$a \frac{e^{1/e}}{e} = e^{(1/e)-1} a.$$

□

Finally, we give the proofs of the two main theorems in this paper.

**Proof of Theorem 3** Using Theorem 9, we have

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

By setting  $a = \ln(4(\ell - 1)/\delta)$  and using the upper bound on  $(1 + \kappa)$  obtained in Lemma 10 for  $\ell = \omega(T_{base}) = \omega(mn \ln(m/\delta))$ , we get

$$\begin{aligned} 1 + \kappa &\leq (1 + o(1)) \frac{\ln(4(\ell - 1)/\delta)}{\ln(\ell - 1) - \ln(4.21mn \ln(2m^2/\delta))} \\ &= (1 + o(1)) \cdot (1 + o(1)) \frac{\ln((\ell - 1)/\delta)}{\ln(\ell) - \ln(mn \ln(m/\delta))} \\ &\leq (1 + o(1)) \frac{\ln(\ell/\delta)}{\ln(\ell) - \ln(mn \ln(m/\delta))}. \end{aligned}$$

□

In Theorem 3, we only consider the case  $\ell = \omega(T_{base})$  since the other cases for  $\ell$  cannot lead to constant approximation ratios and therefore are not interesting to study. More precisely, let us assume  $\ell = \omega(1)$ . In the case that  $\ell < eT_{base} + 1$ , we have the lower bound  $\Omega(\ln(4(\ell - 1)/\delta)) = \omega(1)$  on  $1 + \kappa$  from Lemma 10. Regarding the case that  $\ell \geq eT_{base} + 1$  and  $\ell = O(T_{base})$ , it can be proved that  $1 + \kappa = \Omega(a) = \omega(1)$ ,



since  $\ell/T_{base} = O(1)$  makes all terms constant except  $a$  in Eq. (2). Then again for  $a \geq \ln(4(\ell - 1)/\delta)$  and  $\ell = \omega(1)$ , the approximation ratio is  $\omega(1)$ .

Now, we give the proof of Theorem 4.

**Proof of Theorem 4** Let  $\ell = (mn \ln(m/\delta))^{1+1/\epsilon}$ . Via Theorem 3, we have

$$\begin{aligned} 1 + \kappa &\leq (1 + o(1)) \frac{\ln(\ell/\delta)}{\ln\left(\frac{\ell}{mn \ln(m/\delta)}\right)} \\ &= (1 + o(1)) \frac{(1 + 1/\epsilon) \ln(mn \ln(m/\delta)) + \ln(1/\delta)}{(1/\epsilon) \ln(mn \ln(m/\delta))} \\ &= (1 + o(1)) \left( \frac{1 + 1/\epsilon}{1/\epsilon} + \frac{\ln(1/\delta)}{(1/\epsilon) \ln(mn \ln(m/\delta))} \right) \\ &\leq (1 + o(1)) \left( 1 + \frac{\ln(1/\delta)}{\ln(mn \ln(m/\delta))} \right) (1 + \epsilon). \end{aligned}$$

For  $\delta^{-1} = o(mn \ln n)$ , the last expression can be bounded from above by  $(1 + o(1))(1 + \epsilon)$ .  $\square$

A more straightforward result of Theorem 4 is stated in Corollary 5. In this corollary, we are aiming at expressing an asymptotic time for the algorithm to find the approximation, and we assume that  $\epsilon$  is constant.

**Proof of Corollary 5** Using Theorem 4, we will first prove the result for an approximation ratio of  $(1 + o(1))(1 + \epsilon')$  for some constant  $\epsilon' > 0$  and then bound this by a ratio of at most  $1 + \epsilon$  such that  $(1 + o(1))(1 + \epsilon') \leq 1 + \epsilon$  for  $n$  large enough.

Note that  $\ell = (mn \ln(m/\delta))^{1+1/\epsilon'}$  and  $\delta = \omega(1/(mn \ln n))$  and invoke Theorem 4. The asymptotic bound on  $T^*$  is obtained in the following way: we note that  $\ln(n/\delta) = O(\ln n)$  since  $1/\delta = n^{O(1)}$  by assumption and  $m \leq n^2$ . Since  $\epsilon' > 0$  is constant, we have  $\ln(\ell) = O((1 + 1/\epsilon') \ln(mn \ln(m/\delta))) = O((1 + 1/\epsilon') \ln n) = O(\ln n)$ . Moreover,  $\ell = O((mn \ln(n/\delta))^{1+\epsilon'}) = O((mn \ln(n))^{1+\epsilon'})$ . Putting this together, we have

$$T^* = O((mn \ln(n))^{1+1/\epsilon'} (\ln \ln n + \ln(T_0/w_{\min}))).$$

We have that  $1/\epsilon' = 1/\epsilon + o(1)$  since  $\epsilon$  and  $\epsilon'$  are constants. Hence, we obtain the statement of the corollary.  $\square$

## 5 $(1 + \epsilon)$ -Separated Weights

In this section, we revisit the case that the weights  $w_1, \dots, w_m$  are  $(1 + \epsilon)$ -separated, i. e., there is a constant  $\epsilon > 0$  such that  $w_j \geq (1 + \epsilon)w_i$  if  $w_j > w_i$  for all  $i, j \in \{1, \dots, n\}$ . As mentioned in the introduction in Theorem 1, Wegener proves that SA with high probability finds an MST for any instance with  $(1 + \epsilon)$ -separated weights if  $w_{\max} \leq 2^m$ . More precisely, the proof of his theorem considers a time span of

$O(m^{8+8/\epsilon})$  steps and shows that SA constructs an MST within this time span with probability  $1 - O(1/m)$ .

In the following, we improve this result in two ways. As acknowledged by Wegener himself, he did not optimize the parameters in the final bound on the runtime. Therefore, we can give an improved time bound of  $O((mn \ln(n))^{1+1/\epsilon+o(1)}(\ln \ln n + \ln(T_0/w_{\min})))$ , see Theorem 11 for the precise, more general result. Moreover, we replace the assumption on the largest edge weight by the parameter  $w_{\max}$ . Essentially, we have done all work necessary to show the following theorem already in the previous section, where we proved an approximation result. Now, the  $(1 + \epsilon)$ -separation implies that indeed an optimal solution is found with high probability.

**Theorem 11** *Let  $\delta = \omega(1/(mn \ln n))$  and  $\delta < 1$ ,  $\epsilon > 0$  be a constant. Consider a run of SA with multiplicative cooling schedule with  $\beta = 1 - 1/\ell$  for  $\ell = (mn \ln(m/\delta))^{1+1/\epsilon+o(1)}$  and  $T_0 \geq w_{\max}$  on an instance of the MST problem with  $(1 + \epsilon)$ -separated weights. With probability at least  $1 - \delta$ , at all times  $t \geq T^* := (\ell/2) \ln\left(\frac{\ln(4(\ell-1)/\delta)T_0}{w_{\min}}\right)$  the current solution is optimal. Moreover,*

$$T^* = O((mn \ln(n))^{1+1/\epsilon+o(1)}(\ln \ln n + \ln(T_0/w_{\min}))).$$

**Proof** We first prove the result for  $(1 + o(1))(1 + \epsilon')$ -separated weights for some constant  $\epsilon' > 0$ . Then we prove the result for  $(1 + \epsilon)$ -separated weights such that  $(1 + o(1))(1 + \epsilon') \leq (1 + \epsilon)$  for  $n$  large enough.

Using Lemma 8, with probability  $1 - \delta$ , we have  $w_{\mathcal{T}^*}(k) \leq w_{\mathcal{T}'}(k) < (1 + \kappa)w_{\mathcal{T}^*}(k)$  for each  $k \in [1..n - 1]$ . The  $(1 + \kappa)$ -separated graphs do not have edge weights between  $w_{\mathcal{T}^*}(k)$  and  $(1 + \kappa)w_{\mathcal{T}^*}(k)$  except  $w_{\mathcal{T}^*}(k)$ . Therefore, the algorithm finds an optimal solution.

We need to bound  $1 + \kappa$  using the assumptions in the statement. By setting  $a = \ln(4(\ell - 1)/\delta)$  and using Lemma 10 for  $\ell = (mn \ln(m/\delta))^{1+1/\epsilon'}$ , we bound  $1 + \kappa$  from above by  $(1 + o(1))(1 + \epsilon')$  similarly to the proof of Theorem 4. Since  $\epsilon$  and  $\epsilon'$  are constants and we have  $1/\epsilon' = 1/\epsilon + o(1)$ , we obtain the claim for  $(1 + \epsilon)$ -separated weights.

Regarding  $T^*$ , since  $\epsilon > 0$  is constant, we have  $\ln(\ell) = O((1 + 1/\epsilon + o(1)) \ln(mn \ln(m/\delta))) = O((1 + 1/\epsilon + o(1)) \ln n) = O(\ln n)$ . Moreover,  $\ell = O((mn \ln(n/\delta))^{1+\epsilon+o(1)}) = O((mn \ln(n))^{1+\epsilon+o(1)})$ . Putting this together, we have

$$T^* = O((mn \ln(n))^{1+1/\epsilon+o(1)}(\ln \ln n + \ln(T_0/w_{\min}))). \quad \square$$

## 6 Comparison and Hybridization of SA and (1 + 1) EA

In this section, we compare the performance proven for SA in this work with the known performance of the (1 + 1) EA, a simple single-trajectory evolutionary algorithm. This will both show that SA can have a superior performance on graphs that are not too dense and that a hybridization of the two algorithms, computing an approximate solution via SA and refining it via the (1 + 1) EA, can be a superior approach.

## 6.1 Runtime Analysis for the (1 + 1) EA

To achieve the goals of this section, we need the following result on the performance of the (1 + 1) EA when starting with a solution of a given quality. Since it takes almost no additional effort, we also formulate this result for approximations, that is, for the problem of computing a connected graph on the whole vertex set with total edge weight at most  $(1 + \epsilon)$  times the weight of a minimum spanning tree. In the following,  $w_{\text{opt}}$  denotes the weight of a minimum spanning tree.

**Theorem 12** *Consider using the standard (1 + 1) EA to compute minimum spanning trees. Assume that the objective function  $f : \{0, 1\}^m \rightarrow \mathbb{Z}$  is such that  $f(x)$  equals the sum  $w(x) := \sum_{i=1}^m w_i x_i$  of the edge weights in the solution when  $x$  represents a connected graph and that  $f(x) > \sum_{i=1}^m w_i$  otherwise. Assume that the initial solution  $x^{(0)}$  is connected.*

(a) *The number  $T$  of iterations until an optimal solution is computed satisfies*

$$E[T] \leq em^2(1 + \ln(f(x^{(0)}) - w_{\text{opt}})).$$

For  $\lambda > 0$ , we have

$$\Pr[T > \lceil em^2(\lambda + \ln(f(x^{(0)}) - w_{\text{opt}})) \rceil] \leq \exp(-\lambda).$$

(b) *Let  $\epsilon > 0$ . Then the number  $T_\epsilon$  of iterations until the current solution of the (1 + 1) EA is a connected graph with total weight at most  $(1 + \epsilon)w_{\text{opt}}$  satisfies*

$$E[T_\epsilon] \leq em^2(1 + \ln(1/\epsilon) + \ln(f(x^{(0)})/w_{\text{opt}})).$$

and

$$\Pr[T_\epsilon > \lceil em^2(\lambda + \ln(1/\epsilon) + \ln(f(x^{(0)})/w_{\text{opt}})) \rceil] \leq \exp(-\lambda)$$

for all  $\lambda > 0$ .

This result is similar to Theorem 12 in [19]. It is more general in that it takes into account the quality  $w(x^{(0)})$  of the initial solution. There is no doubt that such a statement could easily have been obtained also with the approach of [19]. Since the proof of Theorem 12 in [19] refers to many arguments of the proof of Theorem 11 in that paper, and since that work could not yet use the more elegant multiplicative-drift approach [21], we now give a short complete proof of our result via multiplicative drift. Compared to [19], it also uses a slightly more elegant way to deal with the two different ways of making progress, omitting edges that are not needed for connectivity and exchanging expensive edges for a cheaper ones (without disconnecting the graph). While we believe that all these changes lead to a more comprehensive proof, it is clear that the central arguments have already appeared in [19].

**Proof of Theorem 12** To apply the multiplicative drift (Theorem 2), we show that the expected fitness of an accepted offspring of a connected solution is smaller than the fitness of the parent. More precisely, let

$$g(x) := w(x) - w_{\text{opt}}$$

for any connected solution  $x$ . Note that apart from the additive offset  $w_{\text{opt}}$ , this is just the fitness of  $x$ . Let  $y$  be obtained from  $x$  via standard bit-wise mutation (flipping each bit independently with probability  $\frac{1}{n}$ ). Let  $z := y$ , if  $f(y) \leq f(x)$ , and  $z := x$  otherwise. That is,  $z$  is the parent individual after one iteration started with  $x$  as parent. We show that

$$E[g(z)] \leq \left(1 - \frac{1}{em^2}\right) g(x). \tag{3}$$

We start by analyzing the structure of the graph  $G_x$  represented by  $x$ . Let  $m' = \|x\|_1$  be the number of edges of  $G_x$ . If  $r = m' - (n - 1)$  is positive, then there are  $r$  edges  $e_{i_1}, \dots, e_{i_r}$  such that  $G' := G_x - \{e_{i_1}, \dots, e_{i_r}\}$  is a spanning tree (this is an elementary result from graph theory). Let us assume that these edges were chosen with maximal weight (“maximality assumption”), that is,  $e_{i_1}, \dots, e_{i_r}$  are  $r$  edges such that  $G' := G_x - \{e_{i_1}, \dots, e_{i_r}\}$  is a spanning tree and for all other sets  $F$  of  $r$  edges such that  $G - F$  is a spanning tree, we have  $\sum_{f \in F} w(f) \leq \sum_{\ell=1}^r w(e_{i_\ell})$ .

By Lemma 1 of [19], which is originally from [25], there are a number  $s \in [0..n-1]$  and pairs  $(e_{j_1}, e_{k_1}), \dots, (e_{j_s}, e_{k_s})$  of edges such that (i) for all  $\ell \in [1..s]$ ,  $e_{j_\ell}$  is an edge of  $G'$ ,  $e_{k_\ell}$  is not an edge of  $G'$ , and the graph  $G' - e_{j_\ell} + e_{k_\ell}$  is connected and has a smaller total weight than  $G'$  (hence  $w(e_{j_\ell}) > w(e_{k_\ell})$ ), and (ii)  $w(G') - \sum_{\ell=1}^s (w(e_{j_\ell}) - w(e_{k_\ell})) = w_{\text{opt}}$ .

Let  $A_\ell, \ell \in [1..r]$ , denote the event that  $y$  is obtained from flipping exactly the  $i_\ell$ -th bit in  $x$  (that is,  $G_y = G_x - e_{i_\ell}$ ). Note that by definition,  $G_y$  is a connected graph in this case, and  $f(y) \leq f(x)$ . Let  $B_\ell, \ell \in [1..s]$ , denote the event that  $y$  is obtained from flipping exactly the bits  $x_{j_\ell}$  and  $x_{k_\ell}$ . If  $e_{j_\ell}$  is an edge of  $G_x$  and  $e_{k_\ell}$  is not, then this mutation exchanges these two edges and the resulting graph  $G_y$  has a better fitness than  $G_x$  (note that  $G_y$  is connected since it contains  $G' - e_{j_\ell} + e_{k_\ell}$ ). Hence assume that  $G_x$  contains both  $e_{j_\ell}$  and  $e_{k_\ell}$ . Then  $G' - e_{j_\ell} + e_{k_\ell}$  is a connected subgraph of  $G$  having smaller weight than  $G'$ , in contradiction to our maximality assumption (that is, the set  $\{e_{i_1}, \dots, e_{i_r}\} \setminus \{e_{k_\ell}\} \cup \{e_{j_\ell}\}$  would have been a set with larger weight such that its removal creates a spanning tree). Hence the case that both  $e_{j_\ell}$  and  $e_{k_\ell}$  are in  $G_x$  does not occur.

Since an offspring is only accepted if it is connected and has not a larger  $g$ -value than  $x$ , we have

$$g(x) - E[g(z)] \geq \sum_{\ell=1}^r \Pr[A_\ell]w(e_{i_\ell}) + \sum_{\ell=1}^s \Pr[B_\ell](w(e_{j_\ell}) - w(e_{k_\ell})).$$

For all  $\ell$ , we have  $\Pr[B_\ell] = \frac{1}{m^2}(1 - \frac{1}{m})^{m-2} \geq \frac{1}{em^2}$  and  $\Pr[A_\ell] = \frac{1}{m}(1 - \frac{1}{m})^{m-1} \geq \frac{1}{em}$ .  
By construction,

$$\begin{aligned} \sum_{\ell=1}^r w(e_{i_\ell}) + \sum_{\ell=1}^s (w(e_{j_\ell}) - w(e_{k_\ell})) &= w(G_x) - w(G') + w(G') - w_{\text{opt}} \\ &= g(x). \end{aligned}$$

Hence  $g(x) - E[g(z)] \geq \frac{1}{em^2}g(x)$ , that is,  $E[g(z)] \leq (1 - \frac{1}{em^2})g(x)$  as claimed in (3).

Since we assumed that all edge weights are integers, the smallest positive value taken by our potential function  $g$  is  $s_{\min} = 1$ . Applying the multiplicative drift theorem (Theorem 3 in [21]), we conclude that the first time  $T$  that  $g(x) = 0$ , which is equivalent to saying that  $x$  encodes a minimum spanning tree, satisfies

$$E[T] \leq \frac{1 + \ln(g(x^{(0)})/s_{\min})}{1/(em^2)} = em^2(1 + \ln(f(x^{(0)}) - w_{\text{opt}})).$$

By the multiplicative drift theorem with tail bounds (Theorem 2), for each  $\lambda > 0$  we also have

$$\Pr[T > \lceil em^2(\lambda + \ln(f(x^{(0)}) - w_{\text{opt}})) \rceil] \leq \exp(-\lambda).$$

Let  $\epsilon > 0$ , let  $w(x^{(0)}) \geq (1 + \epsilon)w_{\text{opt}}$ , and let  $T_\epsilon$  be the first time that  $g(x) \leq \epsilon w_{\text{opt}} =: s_{\min}$ . Note that such an  $x$  is a  $(1 + \epsilon)$ -approximate solution, that is, it encodes a connected graph with weight at most  $(1 + \epsilon)w_{\text{opt}}$ . Let  $\tilde{g}(x)$  defined by  $\tilde{g}(x) = g(x)$  when  $g(x) \geq s_{\min}$  and  $g(x) = 0$  otherwise. Then (3) immediately translates to  $E[\tilde{g}(z)] \leq (1 - \frac{1}{em^2})\tilde{g}(x)$  and the multiplicative drift theorems give

$$\begin{aligned} E[T_\epsilon] &\leq \frac{1 + \ln(\tilde{g}(x^{(0)})/s_{\min})}{1/(em^2)} = em^2 \left( 1 + \ln \left( \frac{f(x^{(0)}) - w_{\text{opt}}}{\epsilon w_{\text{opt}}} \right) \right) \\ &\leq em^2(1 + \ln(1/\epsilon) + \ln(f(x^{(0)})/w_{\text{opt}})) \end{aligned}$$

and

$$\Pr[T_\epsilon > \lceil em^2(\lambda + \ln(1/\epsilon) + \ln(f(x^{(0)})/w_{\text{opt}})) \rceil] \leq \exp(-\lambda)$$

for all  $\lambda > 0$ . □

## 6.2 Comparison of Runtime Bounds

The classical analysis of the  $(1 + 1)$  EA on the MST problem in [19] as well as Theorem 12 both assume integral edge weights. For the SA, we will without loss of

generality assume the same, resulting in a smallest edge weight of at least 1. With  $T_{SA}(\epsilon)$  being the time for SA to obtain a  $(1 + \epsilon)$ -approximation, where  $\epsilon$  is a constant, Corollary 5 with  $T_0 = w_{\max}$  and  $w_{\min} \geq 1$  then states that

$$T_{SA}(\epsilon) = O\left((mn \ln(n))^{1+1/\epsilon+o(1)}(\ln \ln n + \ln(w_{\max}))\right) \tag{4}$$

(with high probability). We emphasize that this is only an upper bound. We do not have any lower bounds on the runtime of SA, and upper bounds and lower bounds for the  $(1 + 1)$  EA on the MST problem are apart by a polynomial factor. In lack of good lower bounds, we are therefore going to compare the upper bound for SA in (4) to an upper bound on the expected approximation time of the  $(1 + 1)$  EA. Such a comparison shows which of the two algorithms has the stronger runtime guarantee. It cannot formally prove anything about the true ranking of the algorithms.

The bound from Theorem 12 depends on the value of the initial search point  $x^{(0)}$ , assuming it to describe a connected graph. Our result for SA in Corollary 5 depends on the extremal edge weights but not on the initial search point. Again, since it seems rather difficult to obtain a result for SA depending on the initial search point, we assume a worst-case perspective and compare the two algorithms with respect to the worst-case bound  $f(x^{(0)}) \leq mw_{\max}$ . Moreover, we assume  $mw_{\max} \geq w_{\text{opt}}^\kappa$  for some constant  $\kappa > 1$ . Hence, the bound on the initial  $f$ -value is assumed significantly larger than the optimum so  $(1 + \epsilon)$ -approximations (for constant  $\epsilon$ ) are not too easy to find.

According to Theorem 12, the time  $T_{(1+1) \text{ EA}}(\epsilon)$  for the  $(1 + 1)$  EA to find a  $(1 + \epsilon)$ -approximation satisfies

$$E[T_{(1+1) \text{ EA}}(\epsilon)] \leq em^2(1 + \ln(1/\epsilon) + \ln(mw_{\max}/w_{\text{opt}})).$$

Using our assumptions, in particular that  $\epsilon$  and  $\kappa$  are constants, the right-hand side is no less than

$$\begin{aligned} &em^2(1 + \ln(1/\epsilon) + (1 - 1/\kappa) \ln(mw_{\max})) \\ &= \Omega(m^2(\ln(m) + \ln(w_{\max}))). \end{aligned} \tag{5}$$

We note that we have derived a *lower* bound on the upper bound on  $E[T_{(1+1) \text{ EA}}(\epsilon)]$  from Theorem 12 since we want to identify situations where this upper bound is larger than the upper bound in (4).

Finally, we carefully compare (4) with (5). Interestingly, the first bound, i. e., the one for SA, can be better for not too sparse graphs. The main reason is that (4) essentially grows like  $(mn)^{1+1/\epsilon}$ , while (5) grows like  $m^2$  (ignoring logarithmic factors). We will work out the asymptotic difference more closely now.

Assume that  $m = \Omega(n^{1+\beta})$  for some constant  $\beta \in (0, 1]$ . Then, comparing the factor  $(mn \ln(n))^{1+1/\epsilon+o(1)}$  from (4) to the factor  $m^2$  from (5) (and noting that the following factor in parentheses is lower in (4)), we essentially have a relative speed-

up of SA compared to  $(1 + 1)$  EA by at least

$$\frac{m^2}{(mn \ln(n))^{1+1/\epsilon}} = \frac{m^{1-1/\epsilon}}{(n \ln(n))^{1+1/\epsilon}},$$

where we ignored the  $o(1)$  in the exponent since, in the following, we can simply increase the constant  $\beta$  by an additive term of  $o(1)$  to adjust for the missing term. Now, plugging in the bound on  $m$ , the bound on the speed-up becomes

$$\frac{n^{(1+\beta)(1-1/\epsilon)}}{(n \ln(n))^{1+1/\epsilon}} = (\ln(n))^{-1-1/\epsilon} n^{(1+\beta)(1-1/\epsilon)-1-1/\epsilon} = (\ln(n))^{-1-1/\epsilon} n^{\beta(1-1/\epsilon)-2/\epsilon},$$

which becomes  $n^{\Omega(1)}$  if

$$\beta > \frac{2/\epsilon}{1-1/\epsilon},$$

where we recall that  $\beta$  is a constant in the range  $(0, 1]$ . Already for  $\epsilon > 3$ , there are feasible  $\beta$ ; e. g., for  $\epsilon = 4$ , it suffices that  $\beta > 2/3$ .

Altogether, for constant-factor approximations with  $\epsilon > 3$ , there is a sufficiently dense graph class such that the upper bound on  $T_{SA}(\epsilon)$  becomes asymptotically smaller than the bound on  $E[T_{(1+1) EA}(\epsilon)]$  (assuming  $mw_{\max} \geq w_{\text{opt}}^k$ ). In this sense, applying SA may be more efficient to obtain constant-factor approximations to the MST than applying the  $(1 + 1)$  EA.

### 6.3 A Hybridization of SA and $(1 + 1)$ EA

As we have seen in the previous subsection, SA may be more efficient at computing approximate solutions to the MST problem than the  $(1 + 1)$  EA. This suggests a hybrid approach where the  $(1 + 1)$  EA is started from an approximate solution obtained from SA. We now analyze in which situations the overall runtime to find an MST with such a hybrid approach is superior to the time taken by the  $(1 + 1)$  EA started with a random solution as in the classical work. To be precise, since we have no lower bounds on these runtimes, we only compare the runtime guarantees obtainable by our results, that is, we study under which conditions we obtain stronger guarantees via a hybridization.

The analysis from the previous subsection indicates that a runtime advantage of SA compared to the  $(1 + 1)$  EA (for computing approximate solutions) only seems to exist for dense graphs. Then the term  $m^2$  appearing in the bound for the  $(1 + 1)$  EA from Theorem 12 outweighs the term  $T_{SA} = (mn \ln(n))^{1+1/\epsilon+o(1)} (\ln \ln n + \ln(w_{\max}/w_{\min}))$  in the bound for SA from Corollary 5. In the following, we assume that  $\epsilon$  is a sufficiently large constant and  $m$  grows sufficiently faster than  $n$  such that  $T_{SA} = o(m^2)$  holds. To ease the presentation, we shall also assume that  $w_{\text{opt}} = \omega(1)$ , which is a very natural assumption.

Let us now consider the following hybridization:

- (a) We start MA with  $T_0 = w_{\max}$  and run it for  $T^*$  steps to achieve a solution of quality no larger than  $(1 + \epsilon)w_{\text{opt}}$  with high probability, with  $T^*$  defined in Corollary 5.
- (b) We initialize the  $(1 + 1)$  EA with the solution of SA at time  $T^*$  and let it run until it has obtained an MST.

We bound the sum of the (expected) times spent by the two algorithms. First, we have  $T^* = O((mn \ln n)^{1+1/\epsilon+o(1)} (\ln \ln n + \ln(w_{\max}/w_{\min})))$ , which bounds the time spent by SA. Let us assume that the solution at that time is of quality no less than  $w^* = (1 + \epsilon)w_{\text{opt}}$ , which happens with high probability. According to Theorem 12, with high probability (take  $\lambda \in \omega(1) \cap o(w_{\text{opt}})$ ), the  $(1 + 1)$  EA starting with a solution of quality  $w^*$  finds the optimum in time

$$O(m^2(1 + \ln(w^* - w_{\text{opt}}))) = O(em^2 \ln w_{\text{opt}}),$$

where we used that  $\epsilon = O(1)$ . Together with  $T^*$ , with high probability, we have a total optimization time of

$$O((mn \ln n)^{1+1/\epsilon+o(1)} (\ln \ln n + \ln(w_{\max}/w_{\min})) + m^2 \ln w_{\text{opt}}).$$

By comparison, the runtime guarantee from [19] for the  $(1 + 1)$  EA started with a uniformly random solution is

$$O(m^2 (\ln n + \ln(w_{\max}/w_{\min}))).$$

Essentially, via the hybrid approach we replace the  $\ln(w_{\max}/w_{\min})$  in the dominant term of the runtime with  $\ln w_{\text{opt}}$  (still assuming sufficiently dense graphs and sufficiently large  $m$  such that  $(mn \ln n)^{1+1/\epsilon+o(1)} (\ln \ln n + \ln(w_{\max}/w_{\min})) = o(m^2)$  holds). This is not a drastic improvement; however, if  $w_{\max}$  is much larger than  $w_{\text{opt}}$  (e. g., since the graph has a very heavy edge that does not appear in MSTs), the runtime bound for the hybrid approach is better than for the plain  $(1 + 1)$  EA.

## 7 Conclusions

We have shown that simulated annealing is a polynomial-time approximation scheme for the minimum spanning tree problem, thereby proving a conjecture by Wegener [3]. Our analyses use state-of-the-art methods and have led to improved results in the case of  $(1 + \epsilon)$ -separated weights, where simulated annealing yields an optimal solution with high probability. Our main result is one of the rare examples where simple randomized search heuristics, with a straightforward representation and objective function, serve as polynomial-time approximation scheme.

Since the runtime analysis of simulated annealing is still underrepresented in the theory of randomized search heuristics, our understanding of its working principles is still limited. In particular, we do not have a clear characterization of the fitness landscapes in which its non-elitism, along with a cooling schedule, is more efficient than global search. The study of the Metropolis Algorithm for the DLB problem in [9]



and our analysis on the minimum spanning tree problem might indicate that landscapes with many, but easy to leave local optima are beneficial; however, more research is needed to support this conjecture.

**Acknowledgements** This work was supported by a public grant as part of the Investissements d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, and a grant by the Independent Research Fund Denmark (DFF-FNU 8021-00260B).

**Funding** Open access funding provided by Royal Danish Library.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Sasaki, G.H., Hajek, B.: The time complexity of maximum matching by simulated annealing. *J. ACM* **35**, 387–403 (1988)
2. Jerrum, M., Sorkin, G.B.: The Metropolis algorithm for graph bisection. *Discret. Appl. Math.* **82**, 155–175 (1998)
3. Wegener, I.: Simulated annealing beats Metropolis in combinatorial optimization. In: *Automata, Languages and Programming, ICALP 2005*, pp. 589–601. Springer, Berlin (2005)
4. Jansen, T., Wegener, I.: A comparison of simulated annealing with a simple evolutionary algorithm on pseudo-Boolean functions of unitation. *Theoret. Comput. Sci.* **386**, 73–93 (2007)
5. Franzin, A., Stützle, T.: Revisiting simulated annealing: a component-based analysis. *Comput. Oper. Res.* **104**, 191–206 (2019)
6. Doerr, B., Rajabi, A., Witt, C.: Simulated annealing is a polynomial-time approximation scheme for the minimum spanning tree problem. In: *Proc. of GECCO '22*, pp. 1381–1389. ACM Press, (2022)
7. Jansen, T.: Simulated annealing. In: Auger, A., Doerr, B. (eds.) *Theory of Randomized Search Heuristics*. World Scientific Publishing, Singapore (2011)
8. Giel, O., Wegener, I.: Evolutionary algorithms and the maximum matching problem. In: *Symposium on Theoretical Aspects of Computer Science, STACS 2003*, pp. 415–426. Springer, Berlin (2003)
9. Wang, S., Zheng, W., Doerr, B.: Choosing the right algorithm with hints from complexity theory. In: *International Joint Conference on Artificial Intelligence, IJCAI 2021*, pp. 1697–1703. *ijcai.org*, (2021)
10. Lissovoi, A., Oliveto, P.S., Warwicker, J.A.: On the time complexity of algorithm selection hyperheuristics for multimodal optimisation. In: *Conference on Artificial Intelligence, AAAI 2019*, pp. 2322–2329. AAAI Press, Washington (2019)
11. Doerr, B.: Exponential upper bounds for the runtime of randomized search heuristics. *Theoret. Comput. Sci.* **851**, 24–38 (2021)
12. Droste, S., Jansen, T., Wegener, I.: Dynamic parameter control in simple evolutionary algorithms. In: *Foundations of Genetic Algorithms, FOGA 2000*, pp. 275–294. Morgan Kaufmann, Burlington (2000)
13. Oliveto, P.S., Paixão, T., Heredia, J.P., Sudholt, D., Trubenová, B.: How to escape local optima in black box optimisation: when non-elitism outperforms elitism. *Algorithmica* **80**, 1604–1633 (2018)
14. Neumann, F., Witt, C.: *Bioinspired Computation in Combinatorial Optimization - Algorithms and Their Computational Complexity*. Springer, Berlin (2010)
15. Witt, C.: Worst-case and average-case approximations by simple randomized search heuristics. In: Diekert, V., Durand, B. (eds.) *Proc. of STACS 2005. Lecture Notes in Computer Science*, vol. 3404, pp. 44–56. Springer, Berlin (2005)
16. Corus, D., Oliveto, P.S., Yazdani, D.: Artificial immune systems can find arbitrarily good approximations for the NP-hard number partitioning problem. *Artif. Intell.* **274**, 180–196 (2019)

17. Horoba, C.: Exploring the runtime of an evolutionary algorithm for the multi-objective shortest path problem. *Evol. Comput.* **18**, 357–381 (2010)
18. Neumann, F., Sutton, A.M.: Parameterized complexity analysis of randomized search heuristics. In: Doerr, B., Neumann, F. (eds.) *Theory of Evolutionary Computation - Recent Developments in Discrete Optimization*. Springer, Berlin (2020)
19. Neumann, F., Wegener, I.: Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoret. Comput. Sci.* **378**, 32–40 (2007)
20. Kirkpatrick, S., Gelatt, C.D., Jr., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**, 671–680 (1983)
21. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. *Algorithmica* **64**, 673–697 (2012)
22. Doerr, B., Goldberg, L.A.: Adaptive drift analysis. *Algorithmica* **65**, 224–250 (2013)
23. Lehre, P.K., Witt, C.: Tail bounds on hitting times of randomized search heuristics using variable drift analysis. *Comb. Probab. Comput.* **30**, 550–569 (2021)
24. Hoorfar, A., Hassani, M.: Inequalities on the Lambert W function and hyperpower function. *J. Inequal. Pure Appl. Math.* **9**, 5–9 (2008)
25. Kano, M.: Maximum and  $k$ -th maximal spanning trees of a weighted graph. *Combinatorica* **7**, 205–214 (1987)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.