Using Markov-Chain Mixing Time Estimates for the Analysis of Ant Colony Optimization

Dirk Sudholt
CERCIA, University of Birmingham
Birmingham, B15 2TT, UK

ABSTRACT

The Markov chain Monte Carlo paradigm has developed powerful and elegant techniques for estimating the time until a Markov chain approaches a stationary distribution. This time is known as mixing time. We introduce the reader into mixing time estimations via coupling arguments and use the mixing of pheromone models for analyzing the expected optimization time of ant colony optimization. We demonstrate the approach for plateaus in pseudo-Boolean optimization and derive upper bounds for the time until a target set is found. We also describe how mixing times can be estimated for MMAS ant systems on shortest path problems.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Algorithms, Performance, Theory

Keywords

Ant colony optimization, Markov chain Monte Carlo, mixing time, coupling, pseudo-Boolean optimization, shortest path problems

1. INTRODUCTION

Ant colonies in nature are capable of solving complex optimization problems, such as finding shortest paths between their nest and a food source. Certain ant species achieve this goal by a simple communication mechanism. These ants deposit pheromones on the ground while searching for food. These chemicals attract other ants, which then tend to follow pheromone trails. The more promising the food source, the more pheromone is deposited. Also, the shorter the path, the quicker it is invested with pheromones as more and more ants are attracted to following the path. Although the cognitive capabilities of a single ant are very limited, a whole ant colony exhibits intelligent behavior known as *swarm intelligence*.

This kind of swarm intelligence has been transferred to an optimization paradigm known as *ant colony optimization* (ACO). ACO algorithms have been successfully applied to the traveling salesman problem (TSP), the Quadratic Assignment Problem, network routing problems, and many other combinatorial problems [7]. The search of an ant is modeled by a random walk on a graph. An ant starts from a source node in search for a target node that represents food. On each edge a certain amount of artificial pheromone is deposited. At each node each ant chooses which edge to take next. This choice is made probabilistically and according to the amount of pheromone placed on the edges. Pheromones can thus be seen as a collective memory of an ant colony.

As in real ant colonies, the pheromones evaporate over time. The amount of evaporation is determined by the so-called *evaporation factor* ρ, $0 < \rho < 1$. In every pheromone update on every edge a ρ-fraction of the pheromone evaporates, i. e., if the edge contains pheromone τ, the remaining amount of pheromone is $(1-\rho) \cdot \tau$ and then eventually new pheromone is added. Intuitively, a large evaporation factor implies that the impact of previously laid pheromones diminishes quickly and new pheromones have a large impact on the system. Small evaporation factors, on the other hand, imply that the system only adapts slowly to new pheromones.

The current state of an ACO algorithm depends crucially on the pheromone values. As pheromones depend on a long history of past solutions, ACO algorithms are, in general, harder to analyze than other search heuristics such as evolutionary algorithms (cf. the survey by Witt [39]). The current population in an evolutionary algorithm typically is Markovian. Contrarily, the set of the most recent solutions of an ACO algorithm generally is not Markovian as new solutions crucially depend on pheromone values. However, if we consider a stochastic process where the current pheromones are included as part of the current state then we can obtain a Markovian model of an ACO algorithm. In contrast to Markov-chain models of evolutionary algorithms, such a Markov chain contains continuous components as pheromones are real valued.

We contribute to the methods used for the analysis of ACO and other randomized search heuristics using ideas from the powerful Markov chain Monte Carlo (MCMC) paradigm [17]. This paradigm has been widely used to design efficient sampling algorithms via simulations of simple Markov

FOGA'11, January 5–9, 2011, A-6867 Schwarzenberg, Austria.

chains. The most famous applications include approximating the permanent of a matrix [18], sampling proper colorings of a graph [14] and computing the volume of convex bodies [26].

As common in MCMC, we regard an ACO algorithm as a Markov chain that—under some conditions—converges to a stationary distribution that represents an equilibrium state of the system. Properties of the system, such as the optimization time or other performance measures, can then be assessed using knowledge on the stationary distribution and the time until the system has come close to the stationary distribution. The latter time is known as *mixing time*.

The task of estimating mixing times for ACO is non-trivial as a Markov chain model of ACO must reflect the pheromone model—a probabilistic model with continuous components. In particular, the number of states attained by an ACO system may be (countably) infinite. A second difficulty is that, unlike for Markov chains for the Metropolis algorithm or the (1+1) EA on a plateau the Markov chain of an ACO algorithm is not time-reversible: while it may be possible to move from state x to state y, the reverse operation is, in general, impossible. Decreasing a pheromone of τ yields a value of $(1-\rho)\tau$. But a reinforcement of the latter value yields pheromone $(1-\rho)^2\tau+\rho$, which in general is not equal to τ.

Nevertheless, we show in the following that it is possible to apply mixing time estimations and MCMC techniques to ACO and that this can be used for bounding the expected optimization time. We are confident that the presented techniques will find further applications for other probabilistic model-building algorithms and non-elitist search heuristics. This in particular includes evolutionary algorithms using more sophisticated selection mechanisms as cut selection for the environmental selection. This topic is highly relevant as there is a trend towards theoretical analyses of non-elitist evolutionary algorithms [13, 19, 23, 35].

After surveying previous work in Section 2, Section 3 introduces techniques for estimating the mixing time. This part can serve as a reference for researchers working on theory of randomized search heuristics. Section 4 then discusses possible implications that can be drawn from mixing times and stationary distributions. Section 5 demonstrates these techniques in an application to pseudo-Boolean optimization and Section 6 explains how the mixing time can be estimated for MMAS-type algorithms for shortest path problems in directed acyclic graphs. We finish with concluding remarks in Section 7.

2. PREVIOUS WORK

This work does not claim to be the first one using Markov chain Monte Carlo techniques in the context of randomized search heuristics. Markov chain techniques including arguments on stationary distributions have already been used for the analysis of evolutionary algorithms (see, e. g., Mitavskiy, Rowe, Wright, and Schmitt [27] and the references therein). In terms of running time analysis, for instance, Garnier, Kallel, and Schoenauer [9] used recurrence arguments based on stationarity for the analysis of the simple (1+1) EA on a needle function. Due to the lack of time-reversibility, this approach cannot be transferred to non-trivial ACO algorithms.

Also ad hoc mixing time approaches have been presented before in the context of evolutionary algorithms, where the "distance" between the current distribution and the stationary distribution was bounded by ad hoc arguments, see the proof of Theorem 10 in Doerr, Gnewuch, Hebbinghaus, and Neumann [4] or Lemma 3 in Lässig and Sudholt [22]. Markov chain Monte Carlo techniques have recently also been used to establish conditions for the success of the Metropolis algorithm in the context of optimization by Sanyal, S, and Biswas [38]. The Metropolis algorithm is a very convenient algorithm for MCMC techniques as for this algorithm it is very easy to compute the stationary distribution.

The first running time analyses of ACO were presented independently by Gutjahr [10] (a technical report appeared in 2006) and Neumann and Witt [30, 32]. These studies focused on simple pseudo-Boolean functions, after this kind of work was explicitly demanded in a survey by Dorigo and Blum [6].

Neumann and Witt [30], Doerr, Neumann, Sudholt, and Witt [5], and Doerr and Johannsen [3] studied a simple algorithm 1-ANT that constructs a pseudo-Boolean solution according to a straightforward construction graph where an ant makes independent choices for each bit. The 1-ANT records the best solution found so far. In case a new solution is found which is not worse, the new solution replaces the old one and pheromones are updated with respect to the new solution. This mechanism implies that each new best-so-far solution leads to only one pheromone update. The mentioned studies have shown that in case ρ is too small this leads to a stagnation behavior as the knowledge gained through improvements cannot be adequately stored in the pheromones. There is a phase transition from polynomial to exponential optimization times for decreasing ρ.

In a different line of research, Gutjahr and Sebastiani [11] and Neumann, Sudholt, and Witt [36] studied an algorithm called MMAS, where the current best-so-far solution is reinforced in every generation. This holds regardless of whether the best-so-far solution has been changed or not. This means that the algorithm might reinforce the same solution over and over again, until the best-so-far solution is replaced. In stark contrast to the 1-ANT, the increased greediness of MMAS leads to polynomial upper bounds on simple pseudo-Boolean functions.

Recently, Neumann, Sudholt, and Witt [37] investigated a variant of MMAS with so-called *iteration-best update*. Instead of recording the current best-so-far solution, the algorithm creates λ ant solutions in each iteration and it reinforces the best one. This is similar to comma strategies in evolutionary computation. A surprising result is that $\lambda = 2$ ants are sufficient for optimizing OneMax in $O(n \log n)$ expected iterations. This result only holds if ρ is chosen appropriately. In particular, the authors prove that if ρ is too large (with respect to a trade-off between ρ and λ) then the expected optimization time is exponential, for every function with a unique global optimum. This phase transition is surprising as the effect is opposite to the phase transition for the 1-ANT; for iteration-best it is essential to choose small values of ρ.

Besides these results also analyses for hybridization with local search [34] and for ACO in combinatorial optimization have appeared. Neumann and Witt [31] investigated ACO algorithms for finding minimum spanning trees. They considered two different construction procedures and proved that for one procedure the use of heuristic information leads to a performance that is better than the performance of a

simple evolutionary algorithm [29], in terms of the number of function evaluations.

Zhou [40] considered ACO for very simple instances of the TSP. This study was significantly extended by Kötzing, Neumann, Röglin, and Witt [21] who considered two different construction procedures and presented an average-case result for the performance of ACO. Kötzing, Lehre, Oliveto, and Neumann [20] investigated the performance of ACO for the minimum cut problem, but they only presented negative results for pheromone-based construction procedures.

Finally, Attiratanasunthron and Fakcharoenphol [2] and Horoba and Sudholt [15] considered ACO for the classical problem of finding shortest paths in graphs. The former authors presented an ant system n-ANT and proved that it can solve the single-destination shortest paths problem on directed acyclic graphs in $O((m\Delta\ell\log(\Delta\ell))/\rho)$ expected iterations. Here m is the number of edges, Δ is the maximum degree in the graph and ℓ is, loosely speaking, the maximum number of edges on any shortest path in the graph. This bound was later improved to $O(\Delta\ell^2 + (\ell\log(\Delta\ell))/\rho)$ iterations by Horoba and Sudholt [15] for a modified ant system $\text{MMAS}_{\text{SDSP}}$, when all shortest paths are unique. This algorithm is faster than n-ANT (when comparing upper bounds) and it overcomes the limitation to acyclic graphs. For the all-pairs shortest path problem a simple interaction mechanism between ants searching for different destinations leads to a remarkable speed-up, if ρ is not too large. This is another example where a slow adaptation of pheromones leads to the best known performance guarantees.

The last authors also presented an extension for stochastic shortest paths [16]. In this setting all edge weights are subject to noise that reflects possible delays and the task is to discover the real shortest paths despite the noise. They proved that in some settings the noise can mislead the search so that the system needs exponential time for finding decent approximate shortest paths.

3. MARKOV CHAINS AND MIXING TIMES

After having described previous work on the analysis of ACO, we start off with an introduction into mixing time techniques from Markov chain Monte Carlo. The basics described in this section can also be found in various book chapters and text books [1, 17, 24].

Mixing time estimations have deep roots in the Markov chain Monte Carlo paradigm. MCMC is a simple yet powerful technique for sampling that has found many applications (see Liu [25] for a detailed coverage of applications). Assume we have a space Ω and a positive weight function $w : \Omega \to \mathbb{R}^+$. The goal is to sample $x \in \Omega$ with a probability that is proportional to its weight, i. e., with probability $\pi(x) = w(x)/Z$ where Z is the sum of all weights, $Z := \sum_{x\in\Omega} w(x)$. The problem is that often Z is unknown.

The idea of Markov chain Monte Carlo is to construct a Markov chain (X_t) on the space Ω that converges to the desired distribution π. More precisely, we require that the probability of the current state being x at time t converges to $\pi(x)$ as $t \to \infty$, regardless of the initial state: for all $x_0 \in \Omega$ $\text{Prob}\,(X_t = x \mid X_0 = x_0) \to \pi(x)$. In such a setting one can simulate the Markov chain for a sufficiently long time and then take the current state of the chain as an approximate sample. A crucial question is how long we need to simulate the Markov chain in order to get close to π. This time is called the *mixing time* of the Markov chain.

The mixing time is directly related to the efficiency of a sampling algorithm. Therefore, MCMC has developed powerful and interesting techniques for bounding the mixing time.

A sequence $(X_t) = X_0, X_1, \ldots$ of elements from Ω is a Markov chain if for all $x_0, \ldots, x_t$

$$\begin{aligned}&\text{Prob}\,(X_t = x_t \mid X_0 = x_0, \ldots, X_{t-1} = x_{t-1})\\ &= \text{Prob}\,(X_t = x_t \mid X_{t-1} = x_{t-1})\,.\end{aligned}$$

In words, the current state depends only on the previous state. The Markov chain can therefore be described by transition probabilities: let $P(x, y)$ denote the probability of moving to state y given that the current state is in x. Then $P^t(x, y)$ denotes the probability of moving from x to y in exactly t steps.

Note that we can describe the Markov chain solely via its transition probabilities. A convenient way to do so is to take a matrix P of all transition probabilities. Given a current distribution x_t at time t then $x_{t+1} = x_t P$ describes the distribution at time $t + 1$.

Under certain conditions a Markov chain converges to a fixed distribution, known as *stationary distribution*.

Definition 1. *A probability distribution π is a stationary distribution for P if $\pi = \pi P$.*

Once the chain has reached the stationary distribution, the current state will always be distributed according to π. We also say that the chain is at stationarity.

Definition 2. *A Markov chain is called* irreducible *if there is a positive probability of reaching any target state from each state in finite time. Formally: for all states x, y there exists some $t \in \mathbb{N}_0$ such that $P^t(x, y) > 0$.*

For irreducible chains it might be the case that certain states may only be reachable periodically. For instance, we might have a Markov chain where with an initial state x_0 some state x only has positive probability if t is odd. In this case we would not expect the Markov chain to converge to a stationary distribution. The opposite term is *aperiodicity*.

Definition 3. *A Markov chain P is aperiodic if for all states x, y we have $\gcd\{t\colon P^t(x, y) > 0\} = 1$.*

A Markov chain is aperiodic if it is irreducible and if there is at least one state $z \in \Omega$ with a positive self-loop probability, i. e., $P(z, z) > 0$. The reason is that from any state x there is a positive probability of reaching z in finite time, spending an arbitrary finite number of steps in the self-loop, and then moving on to y. This excludes the existence of periods.

For finite Markov chains it is known that irreducibility and aperiodicity implies that the Markov chain has a unique stationary distribution. In this case the Markov chain is called *ergodic*. As we will have to deal with countably infinite state spaces, we need one further condition. We have to know that some states can be revisited in finite time, formalized by the term *positive recurrence*.

Definition 4. *State x is recurrent if $\sum_{t\geq 1} P^t(x, x) = 1$. A recurrent state x is positive recurrent if its expected time to return to x from x is finite: $\sum_{t\geq 1} t \cdot P^t(x, x) < \infty$.*

The following theorem is known as the fundamental theorem of Markov chains. The precise formulation is taken from

Mitzenmacher and Upfal [28, Theorem 7.11]; it states under which conditions a Markov chain with a finite or countably infinite state space converges to a stationary distribution.

THEOREM 1 (FUNDAMENTAL THEOREM). *Any irreducible aperiodic Markov chain belongs to one of the following two categories:*

1. *the chain is ergodic—for any pair of states i and j the limit $\lim_{t\to\infty} P^t(j,i)$ exists and is independent of j, and the chain has a unique stationary distribution $\pi_i = \lim_{t\to\infty} P^t(j,i)$ or*
2. *no state is positive recurrent—for all states i and j, $\lim_{t\to\infty} P^t(j,i) = 0$, and the chain has no stationary distribution.*

The distance between the current distribution and the stationary distribution is measured as follows.

DEFINITION 5 (TOTAL VARIATION DISTANCE). *For two probability distributions μ and ν on Ω the* total variation distance *is defined as*

$$||\mu - \nu|| = \frac{1}{2}\sum_{x\in\Omega} |\mu(x) - \nu(x)| = \max_{A\subseteq\Omega} |\mu(A) - \nu(A)|.$$

The total variation distance is thus the maximum difference in probability for any set $A \subseteq \Omega$.

DEFINITION 6 (MIXING TIME). *Consider an ergodic Markov chain starting in x with stationary distribution π. Let $p_x^{(t)}$ denote the distribution of the Markov chain after t steps. Let $t_x(\varepsilon)$ be the time until the total variation distance between the current distribution and the stationary distribution has decreased to ε: $t_x(\varepsilon) = \min\{t\colon ||p_x^{(t)} - \pi|| \le \varepsilon\}$. Let $t(\varepsilon) := \max_{x\in\Omega} t_x(\varepsilon)$ be the worst-case time until this happens.*

The mixing time t_{mix} of the Markov chain is then defined as $t_{\text{mix}} := t(1/(2e))$.

The constant $1/(2e)$ is chosen somewhat arbitrarily; any other constant strictly smaller than $1/2$ would lead to essentially the same results. In fact, when considering a time of αt_{mix} for some $\alpha > 1$ then the total variation distance decreases exponentially with α. This shows that an arbitrary precision can be achieved by letting the Markov chain run sufficiently long. The following statement is well known; a proof is given, e. g., in Levin et al. [24, page 55].

LEMMA 2. $t(\varepsilon) \le t_{\text{mix}} \lceil \log(1/\varepsilon) \rceil$.

One way to derive upper bounds on the mixing time of a Markov chain is by means of couplings. Given a Markov chain P a *coupling* is a pair process of two random processes (X_t, Y_t). Each process represents a copy of the original Markov chain, i. e., the transition probabilities for both X_t and Y_t are specified by P. However, X_t and Y_t can follow a joint distribution, so that X_t and Y_t need not be independent. For instance, we can let X_t and Y_t make the same random decisions, thus sharing the same source of randomness, as long as we do not change the transition probabilities. In fact, the goal of a coupling is to couple both Markov chains such that both chains at some point of time attain the same state. Whenever this happens, the two processes will always have the same state—we say that then X_t and Y_t have coupled.

DEFINITION 7 (COUPLING). *A coupling of a Markov chain P is a pair process (X_t, Y_t) where both X_t and Y_t, viewed in isolation, are copies of the Markov chain P and for all t it holds that if $X_t = Y_t$ then $X_{t+1} = Y_{t+1}$.*

The random time until the two processes have coupled is called the *coupling time*: $T_{xy} = \min\{t\colon X_t = Y_t \mid X_0 = x, Y_0 = y\}$. The worst-case coupling time for all x and y is related to the total variation distance between the current distribution and the stationary distribution as follows.

LEMMA 3.

$$\max_{x\in\Omega} ||p_x^{(t)} - \pi|| \le \max_{x,y\in\Omega} \text{Prob}\,(T_{xy} > t)$$

This immediately implies that the worst-case coupling time gives an upper bound on the mixing time.

COROLLARY 4.

$$t(\varepsilon) \le \min\{t\colon \max_{x,y\in\Omega} \text{Prob}\,(T_{xy} > t) \le \varepsilon\}.$$

Intuitively, the reason why the worst-case coupling time is an upper bound for the mixing time is that it gives an upper bound for the time until a Markov chain X with an arbitrary initialization has coupled with another Markov chain Y that is at stationarity. The reason for this is that the stationary distribution is a convex combination of point distributions (i. e. the initial state has some fixed value with probability 1).

4. HOW TO USE MIXING TIME ESTIMATIONS

Let us discuss how estimations of the mixing time can be used for the analysis of randomized search heuristics. The goal of MCMC is the design of efficient sampling algorithms. A randomized search heuristic can be regarded as an algorithm trying to sample the set OPT of global optima.

THEOREM 5. *Consider a randomized search heuristic that can be represented by an ergodic Markov chain with stationary distribution π. Let* OPT *be the set of global optima and let t_{mix} denote the mixing time on the considered problem. Then the expected optimization time is at most $t_{\text{mix}} \cdot O(\log(1/\pi(\text{OPT})))/\pi(\text{OPT})$.*

PROOF. For any initial state we have that after $t := t(\pi(\text{OPT})/2)$ steps the total variation distance between the current distribution $p_x^{(t)}$ and the stationary distribution π has decreased to at most $||p_x^{(t)} - \pi|| \le \pi(\text{OPT})/2$. By the definition of total variation distance this means that $|p_x^{(t)}(\text{OPT}) - \pi(\text{OPT})| \le \pi(\text{OPT})/2$ and hence $p_x^{(t)}(\text{OPT}) \ge \pi(\text{OPT})/2$. So, after t steps the probability of being in a global optimum is at least $\pi(\text{OPT})/2$. If this is not the case we wait for another t steps and repeat the arguments. Along with Lemma 2, the expected optimization time is at most

$$2t(\pi(\text{OPT})/2)/\pi(\text{OPT}) \le 2t_{\text{mix}} \cdot \lceil \log(2/\pi(\text{OPT})) \rceil / \pi(\text{OPT}).$$

□

Note that we have to wait for the mixing time every time we are unsuccessful in sampling OPT. This is because the condition of not having found OPT implies a bias for the distribution of states in the next step. Hence, we have to

wait for another mixing time steps to "reset" the current distribution towards a distribution that is close to stationarity.

Another use of mixing time arguments is the following. Assume we are interested in maximizing some performance criterion g that only depends on the current state of the algorithm. (An adaptation towards minimization is straightforward.) In the case of population-based algorithms one might think of, for instance, g describing the average, best, or worst fitness in the current population, or the closeness of the current population to some target point. After mixing, this performance measure will be close to the performance at stationarity.

THEOREM 6. *Let $g\colon \Omega \to \mathbb{R}$ be an arbitrary performance measure for the current state of the algorithm. Let $g_{\min}$ and $g_{\max}$ be its minimal and maximal values, respectively, and $\alpha \geq 1$. After $t_{\mix} \cdot 2\alpha$ iterations we have $\mathrm{E}(g) \geq \mathrm{E}_\pi(g) - 2^{-\alpha} \cdot (g_{\max} - g_{\min})$ where $\mathrm{E}_\pi(\cdot)$ denote the expectation of a random variable drawn according to π.*

PROOF. Note that $t(2^{-\alpha}) \leq t_{\mix} \lceil \log(2^\alpha) \rceil \leq t_{\mix} \cdot 2\alpha$, hence after $t_{\mix} \cdot 2\alpha$ steps the total variation distance between the current distribution and π is no more than $2^{-\alpha}$. Pessimistically estimating that the g-value is at $g_{\min}$ with the remaining probability $2^{-\alpha}$,

$$\begin{aligned}
\mathrm{E}(g) &= \sum_{z \in \Omega} \mathrm{Prob}(g = z) \cdot z \\
&\geq \sum_{z \in \Omega} (1 - 2^{-\alpha}) \cdot \mathrm{Prob}(\pi = z) \cdot z + 2^{-\alpha} g_{\min} \\
&= (1 - 2^{-\alpha}) \cdot \mathrm{E}_\pi(g) + 2^{-\alpha} g_{\min} \\
&\geq \mathrm{E}_\pi(g) - 2^{-\alpha} g_{\max} + 2^{-\alpha} g_{\min}.
\end{aligned}$$

□

We see that mixing time estimates can be useful in various ways. In the next sections we show some concrete applications to ACO algorithms.

5. PLATEAUS IN PSEUDO-BOOLEAN OPTIMIZATION

5.1 ACO for Pseudo-Boolean Optimization

As a first application we consider MMAS for pseudo-Boolean optimization, i. e., the maximization of a pseudo-Boolean function $f\colon \{0,1\}^n \to \mathbb{R}$. Many classical combinatorial problems can be modeled as a pseudo-Boolean function. Examples are MAXSAT or selection problems like Knapsack, Vertex Cover, Minimum Spanning Trees or Maximum Matchings, where a bit indicates whether an object, vertex, or edge is selected or not.

A solution, i. e., a bit string of length n, can be obtained by letting an artificial ant traverse a so-called *construction graph* and mapping the path chosen by the ant to binary values. We use the following natural construction graph. In addition to a start node v_0, there is a node v_i for every bit i, $1 \leq i \leq n$. This node can be reached from v_{i-1} by two edges. The edge $e_{i,1}$ corresponds to setting bit i to 1, while $e_{i,0}$ corresponds to setting bit i to 0. The former edge is also called a *1-edge*, the latter is called *0-edge*. An example of a construction graph for $n = 5$ is shown in Figure 1.

In a solution construction process an artificial ant sequentially traverses the nodes $v_0, v_1, \ldots, v_n$. The decision which edge to take is made according to pheromones on the edges. Formally, we denote pheromones by a function $\tau\colon E \to \mathbb{R}_0^+$. From v_{i-1} the edge $e_{i,1}$ is then taken with probability $\tau(e_{i,1})/(\tau(e_{i,0}) + \tau(e_{i,1}))$.

MMAS (see Algorithm 1) starts with an equal amount of pheromone on all edges: $\tau(e_{i,0}) = \tau(e_{i,1}) = 1/2$. Moreover, we ensure that $\tau(e_{i,0}) + \tau(e_{i,1}) = 1$ holds, i. e., pheromones for one bit always sum up to 1. This implies that the probability of taking a specific edge equals its pheromone value; in other words, pheromones and traversal probabilities coincide. The algorithm also keeps track of the current best-so-far solution x^*. In every iteration, first a new solution x is constructed and it replaces x^* if its objective function value is not worse. Finally, the pheromones are updated with respected to the new best-so-far solution.

Algorithm 1 MMAS

```
Set τ(e) := 1/2 for all e ∈ E.
Construct a solution x*.
Update pheromones w. r. t. x*.
loop
    Construct a solution x.
    if f(x) ≥ f(x*) then x* := x.
    Update pheromones w. r. t. x*.
end loop
```

The pheromone update works as follows. Let $\mathrm{P}(x^*)$ denote the path of edges that have been chosen in the creation of x^*. First, a ρ-fraction of all pheromones evaporates and a $(1-\rho)$-fraction remains. Next, some pheromone is added to edges that are part of $\mathrm{P}(x^*)$. To prevent pheromones from dropping to arbitrarily small values, it is common practice to restrict all pheromones to a bounded interval. The precise interval is chosen as $[\tau_{\min}, 1 - \tau_{\min}]$ for some parameter $0 < \tau_{\min} \leq 1/2$. Pheromone bounds are essential to prevent the pheromones from diverging to values arbitrarily close to 0 or 1. The following analysis of the mixing time hinges on these pheromones bounds as a diverging system does not necessarily mix.

In previous work $\tau_{\min}$ was set to $1/n$, inspired by standard mutations in evolutionary computation where for every bit an evolutionary algorithm has a probability of $1/n$ of reverting a wrong decision. Our results will be more general to highlight the impact the choice of $\tau_{\min}$ has on performance.

Depending on whether an edge e is contained in $\mathrm{P}(x^*)$, the pheromone values τ are updated to τ' as follows:

$$\begin{aligned}
\tau'(e) &= \min\{(1-\rho)\cdot\tau(e) + \rho,\ 1 - \tau_{\min}\} && \text{if } e \in \mathrm{P}(x^*), \\
\tau'(e) &= \max\{(1-\rho)\cdot\tau(e),\ \tau_{\min}\} && \text{if } e \notin \mathrm{P}(x^*).
\end{aligned} \tag{1}$$

Note that the pheromones on all 1-edges suffices to describe all pheromones as pheromones for the two edges for each bit sum up to 1.

5.2 An Upper Bound for MMAS on Needle

Randomized search heuristics are frequently confronted with *plateaus* in the search space. Plateaus are regions of equal fitness. As the fitness does not give any useful hints as to where to find better solutions, the best strategy is typically to perform a kind of random walk to explore the plateau. The largest possible plateau in a non-trivial setting

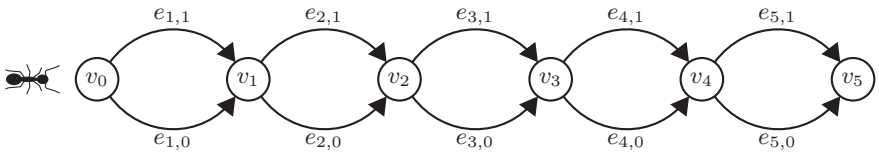

Figure 1: Construction graph for pseudo-Boolean optimization with $n = 5$ bits.

is encountered in the function

$$\textsc{Needle}(x) := \begin{cases} 1 & \text{if } x = x_{\text{OPT}}, \\ 0 & \text{otherwise,} \end{cases}$$

where x_{OPT} is the unique global optimum. It is known that every black-box algorithm needs on average at least $\Omega(2^n)$ function evaluations, where the average is taken over all possible x_{OPT} [8]. Neumann et al. [36] have shown the following upper bound for MMAS on Needle.

Theorem 7 (Neumann et al. [36]). *The expected optimization time of MMAS with $\tau_{\min} = 1/n$ on Needle is bounded from above by $O((n/\rho)^2(\log n) \cdot 2^n)$.*

The proof of Theorem 7 in [36] is tailored towards finding a single search point. W. l. o. g. assume that 1^n is the needle. Neumann et al. [36] then argue that, for every initial pheromone value, once a pheromone for a specific bit reaches a value of at least $1/2$ at some point of time t then the probability of creating a bit value 1 at this bit at any fixed point of time $t' \geq t$ is at least $1/2$. Intuitively, the probability of creating a 1 converges to $1/2$ from above. The expected time until all bits have reached a value of $1/2$ at least once is bounded by $O((n/\rho)^2 \log n)$ and then the probability of constructing the needle in the next iteration is 2^{-n}. In case the needle has not been found, the arguments have to be repeated from scratch as the event of not finding the needle introduces a bias towards low pheromones.

It should be remarked that in [33], a preliminary version of [36], a coupling argument was used. However, the coupling was trivial as two independent Markov chains were considered. The result was limited to constant values of ρ as the estimation of the coupling time breaks down for smaller ρ.

In the following we use coupling arguments with a more thorough coupling and mixing time estimates to re-prove Theorem 7 and to prove a more general result that overcomes the limitation of the needle being a single search point. Despite the larger generality, we emphasize that the focus is on the new proof method and not so much on the result itself.

Consider a generalized Needle function where the needle is an arbitrary target set S.

$$\textsc{Needle}_S(x) := \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{otherwise,} \end{cases}$$

hence S reflects all global optima.

In the remainder of this section we will prove the following result.

Theorem 8. *The expected number of iterations until MMAS finds the optimum on $\textsc{Needle}_S$ for non-empty S is bounded from above by $O((\rho\tau_{\min})^{-2}(\log n) \cdot 2^n/|S|)$.*

Note that until MMAS finds the needle, it behaves as if it was optimizing the constant function that attains value 0 everywhere. Instead of looking at the function Needle, we will consider this constant function. On the constant function, the algorithm behaves independently from the best-so-far search point. Therefore, we can model the current state of the algorithm by looking at the n pheromones on the 1-edges only. We also speak of a *pheromone vector* and refer to the pheromones on the 1-edges.

The following set describes our state space.

Definition 8. *Let $\mathcal{V}$ denote the set of all pheromone vectors τ such that $P^t(\tau^*, \tau) > 0$ for some $t \in \mathbb{N}$ and some $\tau^* \in \{\tau_{\min}, 1 - \tau_{\min}\}^n$.*

The set $\mathcal{V}$ might be infinitely large for appropriate ρ in case there is always one decision—increasing or decreasing a pheromone value—leading to a pheromone value that has not been attained before. However, the size of $\mathcal{V}$ is at most countably infinite as for every bit at most two new pheromone values can be reached in one iteration.

We assume in the following that MMAS is initialized with a pheromone vector in $\mathcal{V}$. Note that this differs from the common initialization where all pheromones are set to $1/2$. We will first prove Theorem 8 with this modified initialization and at the end of this section we will argue how to prove the result for the original initialization.

The first step for applying mixing time arguments is to show that the Markov chain fulfills all necessary requirements.

Lemma 9. *Assume MMAS starts with a pheromone vector in $\mathcal{V}$. The Markov chain representing the pheromone vector of MMAS on the constant function is ergodic.*

Proof. Let τ_x, τ_y be two pheromone vectors in $\mathcal{V}$. By definition of $\mathcal{V}$, there is a vector in $\{\tau_{\min}, 1 - \tau_{\min}\}$, w. l. o. g. $\tau^* = (\tau_{\min}, \dots, \tau_{\min})$, such that τ_y can be reached from τ^* in t steps with positive probability.

If MMAS is currently in state τ_x, there is a positive probability that MMAS first decreases all pheromones to arrive at $(\tau_{\min}, \dots, \tau_{\min})$ in finite time by always creating the search point 0^n. And there is a positive probability of reaching τ_y in t further steps. Hence, the Markov chain is irreducible. It is also aperiodic as there is a positive self-loop probability for all states where all pheromones are at some pheromone border.

By Theorem 1 the chain belongs to either of the two stated categories. To prove that it is ergodic we need to show that there is at least one positive recurrent state. Consider the state $\tau^* = (\tau_{\min}, \dots, \tau_{\min})$ again. The state is clearly recurrent because of the mentioned self-loop. It is also positive recurrent since for any state we arrive at τ^* after having reinforced the solution 0^n at least $\ln(1/\tau_{\min})/\rho$ times. The last term follows from the fact that after evaporating pheromones $\ln(1/\tau_{\min})/\rho$ times any pheromone value τ_i has de-

creased towards $\tau_{\min}$:

$$(1-\rho)^{\ln(1/\tau_{\min})/\rho} \cdot \tau_i \leq e^{-\ln(1/\tau_{\min})} \cdot \tau_i \leq \tau_{\min}.$$

The probability of always creating the solution 0^n in this time span is at least $\tau_{\min}^{n \cdot \ln(1/\tau_{\min})/\rho}$. As this holds for every state, the expected time until we return to τ^* is bounded by $\tau_{\min}^{-n \cdot \ln(1/\tau_{\min})/\rho} + \ln(1/\tau_{\min})/\rho < \infty$. □

Now we know by the fundamental theorem of Markov chains that there exists a stationary distribution π for the pheromone vectors of MMAS on the constant function. By symmetry of bits, we know that all entries of π must follow the same distribution. We do not know the probability density function for this distribution. Actually, it turns out that we do not need to know the probability density function. The following insight on the symmetry of bit values is sufficient for our purposes.

Observation 10. *Let π denote the stationary distribution for MMAS on the constant function. Then π is symmetric in the following sense: if τ_y results from τ_x by "inverting" pheromones on an arbitrary set of bits (inverting means replacing pheromone z by $1-z$) then $\pi(\tau_x) = \pi(\tau_y)$.*

A simple yet essential conclusion from this observation is the following.

Lemma 11. *If the Markov chain for MMAS on the constant function is at stationarity, the next ant solution is drawn uniform from $\{0,1\}^n$.*

Proof. By Observation 10 we have $\mathrm{E}_\pi\left(\tau_i^t\right) = 1/2$, where τ_i^t denotes the random pheromone of the i-th bit at time t. The probability of creating bit value 1 for bit i in the next solution construction is

$$\sum_p p \cdot \mathrm{Prob}_\pi\left(\tau_i^t = p\right) = \mathrm{E}_\pi\left(\tau_i^t\right) = 1/2.$$

As this holds independently for all bits, the next constructed solution is uniform. □

Now we know what the stationary distribution looks like, we can focus on estimating the time until we get close to it. We will basically show that the mixing time of MMAS on the constant function is $O((\rho\tau_{\min})^{-2} \log n)$. In order to prove this, we design the following coupling of two Markov chains X and Y.

The idea is to let X and Y share random decisions for constructing ant solutions whenever possible, without changing the original Markov chains. In each iteration we choose a random vector $r = (r_1, \ldots, r_n)$ where all entries r_i are chosen from the uniform distribution $U[0,1]$. This vector shall serve as a shared source of randomness for both X and Y. Let $\tau^X = (\tau_1^X, \ldots, \tau_n^X)$ be the pheromone vector corresponding to the current state of X. Construct an ant solution for X as follows. For all i, if $r_i < \tau_i^X$ the bit i is set to 1 and to 0 otherwise. Hence, the bit is set to 1 with probability τ_i^X. Then the pheromones τ^X are updated with respect to this ant solution. The same is done for Y with the same random vector r, but with respect to the pheromone vector τ^Y. Note that this way both X and Y are faithful copies of the original process, when viewed in isolation.

It is obvious that once X and Y attain the same pheromone value at some bit i, they will always have equal pheromone values for this bit. (This is not necessarily true in settings where some kind of selection is involved, but in our setting equal bit values imply that the same bit value will be reinforced.) This allows us to focus on the coupling time of a single bit, i. e., the time until the pheromone values become equal on one bit. Once all bits have been coupled, X and Y will always have the same pheromone vector.

What is the coupling time for a single bit? It turns out that the pheromone borders are essential for this estimation. Assume w. l. o. g. that $0 < \tau_i^X < \tau_i^Y < 1$ and let us assume for a moment that there were no pheromone borders. Consider the distance $d_i(X,Y) := \tau_i^Y - \tau_i^X$ between the pheromones. If $r_i > \tau_i^Y$ then both pheromones are multiplied by $(1-\rho)$ and the distance also decreases by a factor of $(1-\rho)$. Symmetrically, if $r_i < \tau_i^X$ then the distance is changed to $(1-\rho)\tau_i^Y + \rho - ((1-\rho)\tau_i^X + \rho) = (1-\rho)d_i(X,Y)$. In the last case, if $\tau_i^X < r_i < \tau_i^Y$ then the distance increases to $(1-\rho)\tau_i^Y + \rho - (1-\rho)\tau_i^X = (1-\rho)d_i(X,Y) + \rho$. The best case is therefore a compression of the distance by $(1-\rho)$. But even then a positive distance can never become 0 as the state space is continuous.

A conclusion drawn from these insights is that the pheromones have to meet at some border in order to couple. Gladly, the following simple observation helps. The distance between X and Y on bit i can be compressed by $(1-\rho)$, but it cannot become negative. In other words, τ_i^Y cannot become smaller than τ_i^X; it always holds that $\tau_i^Y \geq \tau_i^X$. So, when τ_i^X hits the upper pheromone border $1 - \tau_{\min}$, we must have $\tau_i^X = \tau_i^Y$. The same holds when τ_i^Y hits the lower pheromone border $\tau_{\min}$. The coupling time for a bit thus boils down to estimating the time until an arbitrary pheromone value first hits a specific border.

Lemma 12. *The expected coupling time for a specific bit is bounded by $O((\rho\tau_{\min})^{-2})$.*

In order to prove Lemma 12 we apply second-moment arguments. The following lemma is a simplification of Lemma 7 in [36].

Lemma 13 (Neumann, Sudholt, and Witt [36]). *Consider a stochastic process $\{X_t\}_{t\geq 0}$ on a bounded subset of $\mathbb{R}_0^+$. Let $\mathfrak{F}_t$ denote $X_0, \ldots, X_t$, $E_{t+1} := \mathrm{E}(X_{t+1} \mid \mathfrak{F}_t)$ and*

$$\Delta_{t+1} = \mathrm{E}\left((X_{t+1} - E_{t+1})^2 \cdot \mathbb{I}_{X_{t+1} < E_{t+1}} \mid \mathfrak{F}_t\right)$$

with $\mathbb{I}_F$ being the indicator function of event F.

Given $\alpha \in \mathbb{R}$, define $T := \min\{t \colon |X_t - X_0| \geq \alpha \mid X_0\}$. If

1. *$\{X_t\}_{t\geq 0}$ is a supermartingale (i. e. $E_{t+1} \leq X_t$) and*
2. *there exists $\delta > 0$ such that $\Delta_t \geq \delta$ for all $1 \leq t < T$*

then $\mathrm{E}(T) \leq 1 + (2X_0 + \alpha) \cdot \alpha/\delta$.

Proof of Lemma 12. By the discussion preceding this lemma, it suffices to estimate the time until the smaller pheromone τ_i^X hits the upper border or the larger pheromone τ_i^Y hits the lower pheromone border. Note that in at least one of these cases the considered pheromone has distance at least $1/2 - \tau_{\min}$ from the considered border.

Now consider a single pheromone τ_i and assume w. l. o. g. that initially $\tau_i > 1/2$. Then the coupling time is bounded by the time until the pheromone first hits $\tau_{\min}$. If $\tau_i < 1/2$ the argument is symmetric; in case $\tau_i = 1/2$ we just wait one iteration and then continue with one of the above cases.

When waiting until the lower pheromone border is reached, we imagine a modified process where the pheromone is not

capped at the lower pheromone border. Let τ_i^t denote the random pheromone at time t, then the pheromone is a supermartingale: if the upper pheromone border is not hit then $\tau_i^{t+1} = (1-\rho)\tau_i^t + \rho$ with probability τ_i^t and $\tau_i^{t+1} = (1-\rho)\tau_i^t$ with probability $1-\tau_i^t$, hence

$$\mathrm{E}\left(\tau_i^{t+1} \mid \tau_i^t\right) = (1-\rho)\cdot\tau_i^t + \tau_i^t\cdot\rho = \tau_i^t.$$

Obviously, taking into account the upper pheromone border $1-\tau_{\min}$, this can only decrease the expected next pheromone, hence

$$\mathrm{E}\left(\tau_i^{t+1} \mid \tau_i^t\right) \le \tau_i^t.$$

As initially $\tau_i^0 > 1/2$ applying Lemma 13 with $\alpha := \tau_i^0 - \tau_{\min}$ the random time T in this lemma describes the random time until τ_i first reaches a value of or below $\tau_{\min}$. The first condition of the lemma has already been verified, so it remains to prove a lower bound on the one-sided variance.

Observe that $\mathrm{Prob}\left(\tau_i^{t+1} \le (1-\rho)\tau_i^t \mid \tau_i^t\right) \ge 1-\tau_i^t$ as the pheromone is decreased with probability $1-\tau_i^t$. Hence, the one-sided variance (i. e. the variance when considering only decreasing pheromone) is at least

$$(1-\tau_i^t)\cdot(\rho\tau_i^t)^2 \ge (1-\tau_{\min})\cdot(\rho\tau_{\min})^2.$$

Invoking Lemma 13 yields that the expected time until the lower pheromone border is reached is at most

$$1 + (2\tau_i^0+\alpha)\cdot\alpha/((1-\tau_{\min})(\rho\tau_{\min})^2) = O((\rho\tau_{\min})^{-2}).$$

□

Now we can finally assemble all the pieces to prove Theorem 8.

Proof of Theorem 8. We first estimate the worst-case coupling time T where the worst case is taken over all possible pheromone vectors from $\mathcal{V}$ for X and Y. By Lemma 12 the expected time for coupling a single bit is bounded by $c(\rho\tau_{\min})^{-2}$ for some constant $c > 0$. By Markov's inequality the probability that the bit will not have coupled within $2c(\rho\tau_{\min})^{-2}$ iterations is at most $1/2$. As Lemma 12 holds for arbitrary initial pheromones, we can repeat this argument for an arbitrary number of $k \in \mathbb{N}$ phases of $2c(\rho\tau_{\min})^{-2}$ iterations each. The probability that after all phases a bit has not coupled is then bounded by 2^{-k}.

The straightforward way to continue is to choose $k := \log(2^n/|S|) + \log n + 1$. Taking the union bound over all bits yields $\mathrm{Prob}\left(T > 2ck\cdot(\rho\tau_{\min})^{-2}\right) \le |S|/2^{n+1}$. By Lemma 3 this is also a bound on the total variation distance. A solution drawn uniformly at random hits S with probability $|S|/2^n$. Along with Lemma 11 the probability of sampling the set S in the next constructed solution is at least $|S|/2^n - |S|/2^{n+1} = |S|/2^{n+1}$. Hence after $2ck(\rho\tau_{\min})^{-2} = O((\log(2^n/|S|) + \log n)(\rho\tau_{\min})^{-2})$ iterations the next constructed ant solution hits S with probability $\Omega(|S|/2^n)$. Repeating this argument until the algorithm is successful yields an upper bound of $O((\log(2^n/|S|)+\log n)(\rho\tau_{\min})^{-2}\cdot 2^n/|S|)$.

If $\log(2^n/|S|) = \omega(\log n)$ this way of arguing does not give the claimed bound. In this situation it actually makes sense to exploit the independence of bits once more and to argue about single bits directly. We consider the pheromone process and the coupling for a single bit only. Let T_i be the random time until bit i has coupled. Setting $k := \log n$ yields $\mathrm{Prob}\left(T_i > 2c(\log n)\cdot(\rho\tau_{\min})^{-2}\right) \le 2^{-k} = 1/n$. By Lemma 3 the total variation distance between the pheromone distribution on bit i and the stationary distribution on bit i then has decreased to at most $1/n$. This means that the probability for creating bit value 1 at this bit in the next constructed ant solution is at least $1/2 - 1/n$. As this holds for all bits independently, the probability of constructing any specific target point is at least

$$\left(\frac{1}{2}-\frac{1}{n}\right)^n = 2^{-n}\cdot\left(1-\frac{2}{n}\right)^n = \Omega(2^{-n}).$$

Hence the probability of finding $|S|$ after $2c(\log n)(\rho\tau_{\min})^{-2}$ iterations is $\Omega(|S|/2^n)$. This proves the claim. □

At this point we can also answer the question how we can deal with an initialization of pheromones outside $\mathcal{V}$. Our estimation of mixing times states that the expected time until the algorithm reaches some pheromone vector in $\mathcal{V}$ is bounded by $O((\rho\tau_{\min})^{-2}\cdot\log n)$ as we only have to wait until every bit has hit some pheromone border at least once to reach $\mathcal{V}$. This additional waiting time vanishes in the bound from Theorem 8, hence our analysis generalizes to arbitrary initial pheromone vectors.

5.3 Further Conclusions

The preceding theorem can be immediately applied to function where plateaus appear as subspaces of $\{0,1\}^n$. One such example was presented by Gutjahr and Sebastiani [11]: the function NH-OneMax consists of the Needle function on $k = \log n$ bits and the function OneMax on $n-k$ bits, which can only be optimized if the needle has been found on the needle part. The function is defined as

$$\text{NH-OneMax}(x) = \left(\prod_{i=1}^{k} x_i\right)\left(\sum_{i=k+1}^{n} x_i\right).$$

Theorem 8 then gives another proof of a result published in Neumann et al. [36]: replacing n by k, the expected time until MMAS finds the needle is $O((\rho\tau_{\min})^{-2}(\log k)\cdot 2^k) = O((\rho\tau_{\min})^{-2}(\log\log n)\cdot n)$. Including the expected time until the OneMax part is optimized gives an upper bound for NH-OneMax.

The analysis from the preceding subsection has also shown that bits quickly attain a "random" state where the probability of constructing bit value 1 is close to 1/2. The following result may be useful in a more complex context.

Lemma 14. *Call a bit* relevant *with respect to MMAS and the current generation if flipping the bit would result in a different solution becoming the next best-so-far solution.*

If there is a bit that has not been relevant for MMAS in the past $(\rho\tau_{\min})^2\alpha$ iterations then the probability of constructing bit value 1 in the next constructed solution is within $1/2 - 2^{-\Omega(\alpha)}$ and $1/2 + 2^{-\Omega(\alpha)}$.

Results of this kind have already found applications for evolutionary algorithms in pseudo-Boolean optimization, see the proof of Theorem 10 in [4] and Lemma 3 in [22].

6. MIXING OF MMAS ALGORITHMS FOR SHORTEST PATHS

We present another example where mixing time arguments can be used in a more complex setting: ACO for shortest path problems. Previous work focused on MMAS algorithms with best-so-far update. With such an update mechanism, the underlying pheromone model is, in general,

not irreducible as the algorithm will never leave the set of global optima. We therefore focus on MMAS algorithms with iteration-best update, where the best ant solution in the current iteration is reinforced. This implies that the best solution found so far can be lost and the system may converge to a non-trivial stationary distribution. Note that the best-so-far MMAS algorithm from Section 5 is, in fact, an iteration-best algorithm on the considered needle functions.

The motivation for analyzing iteration-best MMAS algorithms is manifold. This update scheme is often used in practice. The recent study by Neumann et al. [37] has shown that iteration-best MMAS systems can be surprisingly effective in pseudo-Boolean optimization. For shortest paths the analyses by Horoba and Sudholt [16] have shown that sticking to the best-so-far path might not be a good choice when dealing with noise. In a stochastic setting, this might encourage a risk-seeking behavior that leads to exponential running times on certain instances [16] as the ants may stick to a sub-optimal path with a large variance. This effect might become even stronger when, unlike in [16], negative noise is allowed. In this sense iteration-best ant systems might be more robust than best-so-far ant systems. Finally, iteration-best is closer to natural ants; it is interesting to see whether less powerful ants can perform as well as ants with a memory.

6.1 MMAS for Shortest Paths

We consider the single-destination shortest path problem on directed acyclic graphs. The task is to find shortest paths from every vertex to a single destination. For simplicity, we assume that the destination can be reached from every vertex as otherwise we can simply add edges with very large weights to obtain this property, without changing the shortest paths.

Assume an n-vertex weighted directed acyclic graph $G = (V, E, w)$ where $w(e)$ denotes the weight of edge e. We define a *path* of length ℓ from u to v as a sequence of vertices $(v_0, \dots, v_\ell)$ where $v_0 = u$, $v_\ell = v$, and $(v_{i-1}, v_i) \in E$ for all i with $1 \le i \le \ell$. As the graph is acyclic, a path cannot contain the same vertex twice. For convenience, we also refer to the corresponding sequence of edges as path. Let $\deg(u)$ denote the out-degree of a vertex u and $\Delta(G)$ or simply Δ denote the maximum out-degree of any vertex $u \in V$. Let $\mathrm{diam}(G)$ denote the diameter of G, that is, the maximum number of edges on any path in G.

We describe the considered iteration-best MMAS algorithm and thereby remark that many of the following arguments transfer to other MMAS algorithms that use the same path construction procedure and the same pheromone update mechanism. The selection of the new paths to be reinforced can be different.

In each iteration from every vertex λ ants start heading for the destination. Each ant chooses a path by performing a random walk through the graph according to pheromones on the edges; the probability of taking an edge is proportional to the amount of pheromone. Algorithm 2 gives a formal description of this procedure.

Afterwards, for each vertex a best path among the λ constructed paths is chosen for an update of the pheromones. As in previous ant systems for shortest paths a local update rule is used: an ant at vertex u is responsible for updating the edges leaving its start vertex u. In the beginning, pheromones $\tau\colon E \to \mathbb{R}_0^+$ are initialized such that all edges leaving some vertex u receive the same amount of pheromone: if $e = (u, \cdot)$ then $\tau(e) = 1/\deg(u)$. Afterwards, in each iteration the pheromones are updated exactly as in Equation (1), where $P(x^*)$ is redefined to the set of edges that are chosen to be reinforced (i. e. the first edges on any path to be reinforced).

Algorithm 2 Path Construction from u to v

```
Initialize i ← 0, p_0 ← u, and V_1 ← {p ∈ V \ {p_0} | (p_0, p) ∈ E}
while p_i ≠ v and V_{i+1} ≠ ∅ do
    i ← i + 1
    Choose a vertex p_i ∈ V_i according to probabilities
    τ((p_{i-1}, p_i)) / Σ_{p∈V_i} τ((p_{i-1}, p))
    V_{i+1} ← {p ∈ V \ {p_0, ..., p_i} | (p_i, p) ∈ E}
end while
return (p_0, ..., p_i)
```

Note that in this setting for a vertex with a large degree it may happen that the sum of pheromones on the outgoing edges exceeds 1. This happens if many low pheromones τ_i for which $(1-\rho)\tau_i < \tau_{\min}$ holds are set to $\tau_{\min}$ in the pheromone update (1), instead of being set to $(1-\rho)\tau_i$. However, Horoba and Sudholt [15] have shown that when $\tau_{\min} \le 1/\Delta$ holds—we assume that this condition always holds since it is required for the initialization to work properly—then probabilities and pheromones only differ by a factor of at most 2.

6.2 The Mixing Time of MMAS Algorithms

We explain how the mixing time of MMAS algorithms can be bounded. Consider a coupling of two Markov chains X, Y with pheromone vectors τ_X, τ_Y of arbitrary initial entries. In the coupling in each iteration λ ants are sent whose random walks are coupled as follows. Consider a setting where an ant has reached the same vertex in X and Y. Let p_i^X and p_i^Y be the probabilities of taking the i-th edge from the current vertex for X and Y, respectively. Then the j-th ants in X and Y with probability $\min\{p_i^X, p_i^Y\}$ both take the i-th edge. With the remaining probability $1 - \sum_i \min\{p_i^X, p_i^Y\}$ the ants make choices so that each ant makes random decisions according to the original path construction. This generalizes the coupling from Section 5. Note that whenever all pheromones at some vertex u are the same, all ants in X and Y make the same decisions at u.

We now consider an incremental coupling for the graph. W. l. o. g. let $V = \{1, \dots, n\}$ and assume that the vertices are topologically ordered with respect to the destination n. Let G_i denote the subgraph of vertices $\{i, \dots, n\}$. Once all pheromones on outgoing edges for these vertices are the same in X and Y we say that X and Y have coupled in G_i. As the vertices are due to a topological ordering, X and Y will always make the same decisions for any ant traversing G_i. Also the pheromones on edges in G_i will be the same in X and Y forever since only ants that start in G_i will change pheromones on edges of G_i.

Assuming that X and Y have coupled in G_{i+1}, we then estimate the time until X and Y have coupled in G_i. As in the previous analysis, we rely on pheromone borders for pheromones to attain equal values in X and Y. We call an edge e *saturated* if $\tau(e) = 1 - \tau_{\min}$. The idea is that once X and Y have both saturated the same edge at i, they will

have the same pheromone values on all outgoing edges. This is verified in the following lemma.

LEMMA 15. *Consider a vertex u with a saturated outgoing edge e. Then for all other outgoing edges e' incident to u it holds that $\tau(e') = \tau_{\min}$.*

PROOF. Assume $\deg(u) > 1$ as otherwise there is nothing to prove. We prove the lemma by showing that

$$\sum_{e=(u,\cdot)\in E} \tau(e) \;\le\; 1 + (\deg(u) - 2)\tau_{\min}.$$

This implies the claim since all $\deg(u) - 1$ non-saturated edges must have pheromone at most $\tau_{\min}$ for the upper bound to hold. Note that this upper bound improves upon Lemma 1 in Horoba and Sudholt [15] where only an upper bound of $1 + \deg(u)\tau_{\min}$ was proven.

We first show the following. Consider two pheromones τ_1, τ_2 and let τ_1', τ_2' denote their respective values after one pheromone update. If $\tau_1 + \tau_2 = 1$ and exactly one of these pheromones is reinforced then we claim that $\tau_1' + \tau_2' = 1$. In case τ_1' and τ_2' are not capped by pheromone borders, we have $\tau_1' + \tau_2' = (1-\rho)(\tau_1 + \tau_2) + \rho = 1 - \rho + \rho = 1$. Otherwise, it is easy to verify that both pheromones hit pheromone borders: $\tau_1' = 1 - \tau_{\min} \Leftrightarrow \tau_2' = \tau_{\min}$ and the same holds when τ_1' and τ_2' are swapped. In this case we clearly have $\tau_1' + \tau_2' = 1 - \tau_{\min} + \tau_{\min} = 1$.

This argument proves the claim for $\deg(u) = 2$. For $\deg(u) > 2$ let $\tau_1, \dots, \tau_{\deg(u)}$ denote all pheromones on outgoing edges of u in order of decreasing pheromones. Assume inductively that the sum of pheromones is bounded by $1 + (\deg(u) - 2)\tau_{\min}$; note that this property is true at initialization as there the sum equals 1. The sum of next pheromones is maximized if as many non-reinforced pheromones are at the lower border $\tau_{\min}$ as possible. This corresponds to $\tau_3 = \cdots = \tau_{\deg(u)} = \tau_{\min}$ and either τ_1 or τ_2 being reinforced. Due to our assumption, $\tau_1 + \tau_2 \le 1$, hence by our previous arguments $\tau_1' + \tau_2' \le 1$. The sum of new pheromones is hence at most $1 + (\deg(u) - 2)\tau_{\min}$. □

Using this lemma, we arrive at the following result.

THEOREM 16. *Consider an iterative coupling of two Markov chains X and Y representing an MMAS algorithm on G. Let T_i be the random time (or a random variable that stochastically dominates it) until X and Y have saturated the same edge outgoing from i, given that X and Y have coupled in G_{i+1}. Then the mixing time of the algorithm is bounded by $O(\sum_{i=1}^{n-1} \mathrm{E}(T_i))$ and $O(\max\{\mathrm{E}(T_i)\} \cdot \mathrm{diam}(G) \log n)$.*

PROOF. By Lemma 15, $\mathrm{E}(T_i)$ is an upper bound on the expected coupling time for G_i under the given circumstances. Therefore, $\sum_{i=1}^{n-1} \mathrm{E}(T_i)$ is an upper bound on the expected coupling time for G. Using Corollary 4, along with Markov's inequality, the first upper bound on the mixing time follows.

The second bound follows from a layering argument: consider the set L_j of all vertices having a maximum number of j edges on their paths to the destination. Observe that when all vertices in $L_1, \dots, L_{j-1}$ have coupled we can use the previous arguments to estimate the time until a fixed vertex in L_j becomes coupled. The expected time for this is at most $\max\{\mathrm{E}(T_i)\}$ and by Markov's inequality the probability of not having coupled in $2\max\{\mathrm{E}(T_i)\}$ iterations is at most $1/2$. Considering $(\log n) + 1$ periods of this length each, the probability of not having coupled is at most $1/(2n)$ for any fixed vertex in L_j. By the union bound, the probability that any vertex in L_j is not coupled is at most $1/2$. As the expected number of trials is at most 2, the bound $2((\log n)+1) \cdot 2\max\{\mathrm{E}(T_i)\}$ follows. Adding these times for all $\mathrm{diam}(G)$ layers proves the second bound. □

A complete analysis of iteration-best MMAS algorithms is beyond the scope of this work. The purpose here is to illustrate how couplings can be used for bounding the mixing time in a setting where not all components are independent. The ansatz presented here may be used as a basis for future analyses. So we only sketch how one could continue to bound the coupling time of single vertices and arrive at a complete analysis.

If the ant system is parametrized so that it shows a rather chaotic behavior, which occurs for large ρ, one might hope to saturate a specific edge simply because the ant system might by chance only construct ant solutions that start with this edge, until the edge becomes saturated. This resembles the notion of a *landslide sequence* for pheromones used in [37]. Indeed, it is not difficult to prove a polynomial bound for the mixing time of an iteration-best MMAS system with λ ants if $\lambda/\rho = O(\log n)$.

On the other hand, in a setting where there is good guidance towards finding shortest paths and shortest paths are unique, we might experience a drift towards increasing pheromones on the unique first edge of a shortest path. This, in turn, might be transformed into an upper bound for the time until the edge becomes saturated.

Drift analysis can also be key for analyzing the stationary distribution. Already the famous work by Hajek [12] has established occupation time bounds for all considered states. Simply speaking, in the presence of a drift the probability of being in a specific state decreases exponentially with the distance from the state the drift is pointing to. This can then in turn be used to lower-bound the probability of being at a large distance and to show that the closest states (i. e., high pheromones on the edges of a shortest path) have a large probability mass in the stationary distribution. With this argument, an upper bound on the expected time until shortest paths are found can be shown.

This way of reasoning might also prove useful for iteration-best MMAS in pseudo-Boolean optimization as the upper bound in Neumann et al. [37] is based on a positive drift for all bits.

7. CONCLUSIONS

Mixing time estimations and coupling techniques can be applied to the analysis of randomized search heuristics. We have demonstrated this for ant colony optimization where the Markov chain reflects a pheromone model with continuous components. We have seen how MMAS deals with plateaus in pseudo-Boolean spaces and subspaces and discussed how to estimate the mixing time of MMAS systems for shortest paths in directed acyclic graphs.

We expect the presented techniques to find further applications for the analysis of bio-inspired search heuristics. This includes further probabilistic model-building algorithms such as estimation-of-distribution algorithms as well as particle swarm optimization and evolutionary algorithms. The latter class of algorithms is particularly relevant in the light of recent considerations of non-elitist environmental selection schemes [13, 19, 23, 36]. As these algorithms naturally

converge to a non-trivial stationary distribution (in a sense that stationarity is not equivalent to being in a global optimum), Markov chain Monte Carlo techniques may lead to running time estimations where common techniques for elitist algorithms break down.

Acknowledgments

The author was partially supported by a postdoctoral fellowship from the German Academic Exchange Service while visiting the International Computer Science Institute in Berkeley, CA, USA and EPSRC grant EP/D052785/1. The author thanks Alistair Sinclair for an exciting lecture on Markov chain Monte Carlo techniques and Thomas Sauerwald for references and insightful discussions on whether the results in Section 5 can be improved. Thanks also go to Benjamin Doerr for advice on Markov chains with countably infinite states and to Christine Zarges and Daniel Johannsen for many useful comments on the presentation.

References

[1] D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. Monograph in preparation, `http://stat-www.berkeley.edu/users/aldous/RWG/book.html`, 2010.

[2] N. Attiratanasunthron and J. Fakcharoenphol. A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs. *Information Processing Letters*, 105(3):88–92, 2008.

[3] B. Doerr and D. Johannsen. Refined runtime analysis of a basic ant colony optimization algorithm. In *Proceedings of the Congress of Evolutionary Computation (CEC '07)*, pages 501–507. IEEE Press, 2007.

[4] B. Doerr, M. Gnewuch, N. Hebbinghaus, and F. Neumann. A rigorous view on neutrality. In *Proceedings of the Congress of Evolutionary Computation (CEC '07)*, pages 2591–2597. IEEE Press, 2007.

[5] B. Doerr, F. Neumann, D. Sudholt, and C. Witt. On the runtime analysis of the 1-ANT ACO algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '07)*, pages 33–40. ACM, 2007.

[6] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344:243–278, 2005.

[7] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.

[8] S. Droste, T. Jansen, and I. Wegener. Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory of Computing Systems*, 39(4): 525–544, 2006.

[9] J. Garnier, L. Kallel, and M. Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):173–203, 1999.

[10] W. J. Gutjahr. First steps to the runtime complexity analysis of ant colony optimization. *Computers and Operations Research*, 35(9):2711–2727, 2008.

[11] W. J. Gutjahr and G. Sebastiani. Runtime analysis of ant colony optimization with best-so-far reinforcement. *Methodology and Computing in Applied Probability*, 10: 409–433, 2008.

[12] B. Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied Probability*, 14:502–525, 1982.

[13] E. Happ, D. Johannsen, C. Klein, and F. Neumann. Rigorous analyses of fitness-proportional selection for optimizing linear functions. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '08)*, pages 953–960. ACM, 2008.

[14] T. P. Hayes, J. C. Vera, and E. Vigoda. Randomly coloring planar graphs with fewer colors than the maximum degree. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 450–458. ACM, 2007.

[15] C. Horoba and D. Sudholt. Running time analysis of ACO systems for shortest path problems. In *Proceedings of Engineering Stochastic Local Search Algorithms (SLS '09)*, volume 5752 of *LNCS*, pages 76–91. Springer, 2009.

[16] C. Horoba and D. Sudholt. Ant colony optimization for stochastic shortest path problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 1465–1472, 2010.

[17] M. Jerrum and A. Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems*, pages 482–520. PWS Publishing, 1996.

[18] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697, 2004.

[19] J. Jägersküpper and T. Storch. When the plus strategy outperforms the comma strategy – and when not. In *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI '07)*, pages 25–32. IEEE, 2007.

[20] T. Kötzing, P. K. Lehre, P. S. Oliveto, and F. Neumann. Ant colony optimization and the minimum cut problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 1393–1400. ACM, 2010.

[21] T. Kötzing, F. Neumann, H. Röglin, and C. Witt. Theoretical properties of two ACO approaches for the traveling salesman problem. In *Seventh International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '10)*, volume 6234 of *LNCS*, pages 324–335. Springer, 2010.

[22] J. Lässig and D. Sudholt. The benefit of migration in parallel evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 1105–1112, 2010.

[23] P. K. Lehre and X. Yao. On the impact of the mutation-selection balance on the runtime of evolutionary algorithms. In *FOGA '09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 47–58. ACM, 2009.

[24] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[25] J. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2002.

[26] L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006.

[27] B. Mitavskiy, J. E. Rowe, A. H. Wright, and L. M. Schmitt. Quotients of Markov chains and asymptotic properties of the stationary distribution of the Markov chain associated to an evolutionary algorithm. *Genetic Programming and Evolvable Machines*, 9(2):109–123, 2008.

[28] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[29] F. Neumann and I. Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoretical Computer Science*, 378(1): 32–40, 2007.

[30] F. Neumann and C. Witt. Runtime analysis of a simple ant colony optimization algorithm. In *Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC '06)*, volume 4288 of *LNCS*, pages 618–627. Springer, 2006.

[31] F. Neumann and C. Witt. Ant Colony Optimization and the minimum spanning tree problem. In *Proceedings of Learning and Intelligent Optimization (LION '07)*, volume 5313 of *LNCS*, pages 153–166. Springer, 2008.

[32] F. Neumann and C. Witt. Runtime analysis of a simple ant colony optimization algorithm. *Algorithmica*, 54(2): 243–255, 2009.

[33] F. Neumann, D. Sudholt, and C. Witt. Comparing variants of MMAS ACO algorithms on pseudo-Boolean functions. In *Proceedings of Engineering Stochastic Local Search Algorithms (SLS '07)*, volume 4638 of *LNCS*, pages 61–75. Springer, 2007.

[34] F. Neumann, D. Sudholt, and C. Witt. Rigorous analyses for the combination of ant colony optimization and local search. In *Proceedings of the Sixth International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '08)*, volume 5217 of *LNCS*, pages 132–143. Springer, 2008.

[35] F. Neumann, P. S. Oliveto, and C. Witt. Theoretical analysis of fitness-proportional selection: landscapes and efficiency. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 835–842. ACM, 2009.

[36] F. Neumann, D. Sudholt, and C. Witt. Analysis of different MMAS ACO algorithms on unimodal functions and plateaus. *Swarm Intelligence*, 3(1):35–68, 2009.

[37] F. Neumann, D. Sudholt, and C. Witt. A few ants are enough: ACO with iteration-best update. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 63–70. ACM, 2010.

[38] S. Sanyal, R. S, and S. Biswas. Necessary and sufficient conditions for success of the metropolis algorithm for optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 1417–1424. ACM, 2010.

[39] C. Witt. Rigorous runtime analysis of swarm intelligence algorithms – an overview. In *Swarm Intelligence for Multi-objective Problems in Data Mining*, volume 242/2009 of *Studies in Computational Intelligence*, pages 157–177. Springer, 2009.

[40] Y. Zhou. Runtime analysis of an ant colony optimization algorithm for TSP instances. *IEEE Transactions on Evolutionary Computation*, 13(5):1083–1092, 2009.