Ants Easily Solve Stochastic Shortest Path Problems

Benjamin Doerr
Max Planck Institute for Informatics
Saarbrücken, Germany
doerr@mpi-inf.mpg.de

Ashish Ranjan Hota
Indian Institute of Technology
Kharagpur, India
hota.ashish@acm.org

Timo Kötzing
Max Planck Institute for Informatics
Saarbrücken, Germany
koetzing@mpi-inf.mpg.de

ABSTRACT

The recent theoretical analysis (Horoba, Sudholt (GECCO 2010)) of an ant colony optimizer for the stochastic shortest path problem suggests that ant system experiences significant difficulties when the input data is prone to noise. In this work, we propose a slightly different ant optimizer to deal with noise.

We prove that, under mild conditions, it finds the paths with shortest expected length efficiently, despite the fact that we do not have convergence in the classic sense. To prove our results, we introduce a stronger drift theorem that can also deal with the situation that the progress is faster when one is closer to the goal.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Theory, algorithms

Keywords

stochastic shortest path, running time analysis, theory

1. INTRODUCTION

Ant colony optimization (ACO) is a powerful bio-inspired optimization meta-heuristic. It has been applied successfully to a wide range of optimization problems, not restricted to the natural application area of graph algorithms. A key feature of these algorithms is that they do not stick to a fixed set (population) of solution candidates, but evolve a probability distribution whose mass ideally converges to favorable parts of the search space. For this reason, it seems likely that ACO approaches work robustly in computationally harder settings, for example, those involving uncertainty. Unfortunately, the seemingly first and still only rigorous runtime analysis of ACO on a problem with uncertainty [22] shows that a natural adaption of the Min-Max Ant System (MMAS) to the stochastic single-source shortest path problem experiences significant difficulties in finding (in expectation) optimal solutions.

In this work, we show how to overcome many of the difficulties demonstrated there. The key difference in this work is that we reevaluate the best-so-far solution whenever it is compared to a newly found one. This avoids being mislead by an exceptional, too optimistic evaluation. A consequence of the reevaluation is that we do not have convergence in the pheromone values anymore. From the view-point of application, this is not a problem, as taking a solution with close to maximum pheromone value safely identifies the optimum.

In the mathematical analysis performed in this paper, however, the absence of convergence poses some challenges. To overcome them, we use drift analysis [19, 20]. Since the drift we encounter is not monotone with the distance to the optimum, we prove the first drift theorem not needing this assumption. We are optimistic that this will find more applications in the near future.

1.1 Ant Colony Optimization

The ant colony optimization (ACO) method was suggested by Dorigo in his Ph.D. thesis [8] and has since found numerous applications, see, e.g., [9]. The basic idea is to mimic the foraging behavior of an ant colony, which builds on ant leaving pheromone traces on paths leading to food sources. Since paths to rich food sources are used more often, the pheromone trail becomes stronger here, attracting other ants to follow this route. While this paradigm naturally is suited for optimization problems involving shortest paths, routes, etc., it can also be applied to problems without a natural graph structure by adding a suitable *construction graph*. What is generally seen as a strength of ACO systems is that they do not keep a sample of the search space (population) as do many evolutionary algorithms, but rather evolve a probability space defined on the solution space.

The downside of this richer approach is that theoretical analyses of ACO systems are seemingly very difficult. While a well-developed theory of evolutionary algorithms exists for many years, it was not before the year 2000 that convergence results were proven [11, 12] and it took until 2007 for the first theoretical run-time analyses for ACO systems to appear [15, 16, 7, 17]. These results clearly show that analyzing even very simple ACO algorithms is much more difficult than analyzing simple evolutionary algorithms, and even more recent papers like [24] seemingly found no simple analysis method.

In this work, we consider ACO in the presence of noisy

GECCO'12, July 7–11, 2012, Philadelphia, Pennsylvania, USA.

data (optimization in the presence of stochastic uncertainties is sometimes called stochastic programming or stochastic combinatorial optimization). While there is good empirical evidence that using ACO in stochastic settings makes sense (for this we refer to the survey [3] and the references therein), a clear theoretical understanding of this aspect is still missing.

Indeed, Gutjahr in [13, 14] proposed the S-ACO algorithm for stochastic problems and proved convergence under moderate conditions. A main ingredient of this algorithm is that in the pheromone update, the observed quality of a new solution is not simply compared with the (at that time observed) quality of the best-so-far solution, but rather both solution candidates are re-sampled an increasing number of times to get more stable estimates.

On the other hand, the single rigorous run-time analysis for stochastic problems [22] does not suggest that ACO methods are particularly suitable in stochastic optimization. Let us describe their approach in more detail. The work [22] studies how shortest paths can be found using ACO when the edge lengths are stochastic, that is, there is random noise added to the evaluation of any edge length. Like previous works performing a run-time analysis on shortest path problems [1, 21], an ACO system was used where one ant starts in each node in parallel. Each of these ants only puts pheromone on the first edge of its walk (which is an outgoing edge from its starting node). Otherwise, the ACO system builds on the Max-Min Ant System [27], in which the current solution is compared to the best-so-far seen solution and the pheromone update is made with respect to the better of the two.

Also, to make ant walks and paths to be found lead into the same direction, not the classic single-source shortest path problem is regarded, but instead the single-destination version. This is no loss, since one variant can be turned into the other by reverting all directions. For this problem formulation, both [1] and [21] show good run-time results for the ACO system analyzed. Hence, the non-stochastic version seems to be well tractable with the ACO algorithm proposed.

For the stochastic setting, [22] basically uses the same ant system. Thus, when an ant produces a path, its stochastic length is evaluated and compared with the best-so-far seen length of a path from this starting point. The better of the two is reinforced by the pheromone system and stored as best-so-far solution and solution value. This system seems to have difficulties optimizing stochastic shortest paths. The authors present an example graph where, with high probability, the ants get stuck on an inferior solution (see Section 4 of this paper for a simple graph demonstrating the same effect). The authors also show that the ACO system suggested can compute approximate solutions. However, the approximation ratio, that is, the factor by which the solution computed is longer than the optimal one, is exponential in the length (number of edges) of the longest path. This makes no sense for most real-world graphs.

In this work, we overcome the difficulties seen in [22] via a simple modification. Analyzing the proofs there, we find the main difficulty is that an ant might be exceptionally lucky and evaluate a path to have a relatively small length. If this happens, the ant will prefer this path in the future, despite the actual quality of this solution not being so good. The simple way to overcome this is to reevaluate the best-so-far solution whenever it is compared to an actual solution. Clearly, this makes the comparison more fair, with both solutions "just having a single try" to evaluate shorter. This approach differs from the one of the S-ACO algorithm, where a relatively large number of reevaluations is done to not only get fairness, but also relatively stable evaluations. In a nutshell, the outcome of our work is that a single reevaluation often is enough.

The results we obtain justify this positive opinion. The ant system with reevaluation easily optimizes the example graph for which the previous approach failed. More generally, we show that for suitable choices of the system's parameters, the ACO system finds all shortest paths that, in direct comparison, have a chance of slightly more than $1/2$ of being evaluated shorter. For reasonable probability distributions for the noise, this implies that the ants find the paths with the shortest expected length.

While from the view-point of application the new system provides little difficulties, for the analysis this is different. Since the new system does not "lock in" the best-so-far solution (as often did the previous, whether actually better or not), we do not have such a strong convergence of the pheromone values. In fact, we do not have convergence at all, because it may always happen that the path which is better in expectation is evaluated to be longer than an alternative one. Of course, we do have the property that the aimed at solution often looks better than the alternatives, and this is enough to show that its pheromone values stay close to the maximum value. Hence, we may read off the optimal solution from those edges that carry close to maximum pheromone.

The true difficulty is in the mathematical analysis of the ACO process. Since we do not have convergence, we only observe a weaker drift behavior towards high pheromone values on preferred edges. *Drift analysis*, as introduced to our field by He and Yao [19, 20], is an advanced method to deal with such situations. Unfortunately, in our case the drift displays an uncommon behavior, being strongest for medium range pheromone values and weaker at the extremes. This makes all of the *additive drift* method of He and Yao, best for uniform drift, the *multiplicative drift* method by Doerr, Johannsen and Winzen [6], best for drift proportional to the distance from the optimum, and the *variable drift* method of Johannsen [23] (cited, e.g., in [5]), applicable whenever the drift is monotone with the distance from the optimum, give only bad run-time bounds. For this reason, we prove a generalization of the variable drift theorem (in a sense, of all drift theorems so far) that does not need the assumption that the drift is non-decreasing with the distance from the goal. This drift theorem might be useful for future research, in particular, in the ACO field.

Due to space constraints, some proofs are omitted.

2. PROBLEM DESCRIPTION AND ALGORITHM

In this section, we give a formal definition of the stochastic shortest path problem we regard and present the ACO algorithm we propose and analyze. We also describe how both relate to the well-known bandits learning problem.

2.1 Stochastic Shortest Path Problems

As most previous works, we regard a variant of the single-

destination shortest path (SDSP) problem. Here, we are looking for shortest paths from any other vertex to a particular vertex called *sink*. Of course, by reverting the directions of all edges we can transform any problem instance (including the optimal solution) of the SDSP to the classic single-source shortest path problem and vice versa. So both problems are equivalent.

Definition 1 (Stochastic SDSP Problem). *Let (V, E) be a directed acyclic graph (DAG). Assume that there is a unique sink (a vertex without edges going out), and that there is a path from every other vertex to the sink. For each edge $e \in E$, let X_e be a random variable describing the stochastic length of e. Denote by $X = (X_e)_{e \in E}$ the family of all these. Then the triple $G = (V, E, X)$ is called* graph with stochastic edge weights *or* stochastic-weight graph.

For any (directed) path p consisting of the edges $E_p \subseteq E$, we let $X_p = \sum_{e \in E_p} X_e$ be the (random) length of the path p.

The aim of this investigation is analyzing to what extent ACO systems are capable of finding a shortest path from any vertex in G to the sink. As pointed out in the introduction, in contrast with the deterministic setting, it is not so clear what "shortest path" means in the context of stochastic path lengths. One possible goal could be to find paths with minimal expected length. The performance measure could be the first hitting time. As we will see later, our ACO system only achieves the similar aim of finding paths with good probability of appearing shorter than other alternatives. While pathological examples show that there is little correlation between the two goals, for many natural distributions the difference is small.

We would like to stress that our focus is on understanding how natural ACO systems behave in situations of uncertainty. It is absolutely clear that custom-tailored methods are better suited to overcome the particular uncertainties in our setting. For example, by sufficiently often sampling the lengths of all edges separately, one can derive an arbitrarily close approximation of the underlying length distribution. This would allow, e.g., replacing the stochastic edge lengths with the approximate expected edge lengths and then running a classic shortest path algorithm to compute expected shortest paths.

2.2 Our ACO Algorithm

The common theme of all ACO algorithms is that artificial ants do a random walk over a given graph. The random decisions are heavily influenced by pheromone values attached to the edges. The outcome of each walk (in fact, it will be always a path) is interpreted as a solution of the underlying problem. Depending on the fitness of this solution, the pheromone values are updated in a way that (hopefully) makes this solution and similar ones more likely to be found in future ant walks.

In this subsection, we first make precise the construction of paths via ant walks, describe the way pheromones are updated, and finally give the ACO algorithm resulting from these and further design choices. The path construction and the pheromone update rule follow the usual standards. The ACO algorithm considered is the so-called Max-Min Ant System (MMAS) as first introduced in [27]. As in [22], we adopt it to the SDSP problem by having one ant start in each vertex in parallel. This is justified by the fact that we are looking for shortest paths from each vertex to the sink. What constitutes the crucial difference to [22] is that we reevaluate the best-so-far solution for each ant walk (see line 8 in Algorithm 2).

Path construction: As previously mentioned, we start (in parallel) one ant a_v from every non-sink vertex v and let it construct a path from v to the sink. When situated at a non-sink vertex, the ant randomly chooses one outgoing edge with a probability proportional to the pheromone on that edge, and then traverses this edge. The ants repeat this step until the sink is reached (since the graph is acyclic and has a unique sink, this always happens, without multiple visits to a vertex). This path construction procedure is made precise in Algorithm 1.

Algorithm 1: Path Construction from u to the sink

```
1 Input: DAG G, start vertex u, pheromones τ;
2 i ← 0, p_0 ← u;
3 V_1 ← {p ∈ V | (p_0, p) ∈ E};
4 while V_{i+1} ≠ ∅ do
5 |   i ← i + 1;
6 |   choose p_i ∈ V_i with probability
  |   τ(p_{i-1}, p_i) / Σ_{p∈V_i} τ(p_{i-1}, p_i);
7 |   V_{i+1} ← {p ∈ V | (p_i, p) ∈ p};
8 return (p_0, ..., p_i);
```

Pheromone updates: Assume that the ant a_v has found a path p from v to the sink which we consider good. In this case, to increase the probability that this path is found again, we update the pheromone values on the edge outgoing from v, that lies on path p. We use the following rule for the pheromone update. Let $\tau : E \to [0, 1]$ describe our current pheromone values on the edges. Then the pheromone values $\tau' := \tau'(\tau, p)$ after this update satisfy

$$\tau'(e) = \begin{cases} \min((1-\rho)\tau(e) + \rho, \tau_{\max}), & \text{if } e \text{ is used in } p; \\ \max((1-\rho)\tau(e), \tau_{\min}), & \text{otherwise} \end{cases}$$

for all edges out-going from v. For all other edges e, we have $\tau'(e) = \tau(e)$. Note that $\tau_{\min}$ and $\tau_{\max}$ are parameters to the algorithm.

The MMAS ACO algorithm with reevaluation of stochastic lengths: The basic concept of the MMAS ACO algorithm is that after each run of an ant, a pheromone update is made. If the just-constructed path p is at least as good as the best-so-far seen path p^*, then the update is done with respect to p, otherwise with respect to p^*.

Since we have one ant a_v for each non-sink vertex v, we store a best-so-far solution p_v^* for each vertex and compare this with the solution constructed by a_v.

There is one difficulty to deal with for stochastic settings: there is no absolute quality of solutions. Due to the stochasticity, the same path may be evaluated with different lengths. Horoba and Sudholt [22] decided that they store the best-ever seen length with the best-so-far solution. However, as can be seen from analyzing their proofs, this leads to strange behavior. If a path was ever evaluated at an unexpectedly short length (as may rarely happen), then this path looks too good for the remainder of the ACO run.

To overcome this problem, we suggest to reevaluate the best-so-far solution whenever it is compared to another so-

lution. By this we hope to better capture the typical quality of a solution.

Algorithm 2 gives the complete Max-Min Ant System with reevaluation of the best-so-far solution.

Algorithm 2: Max-Min Ant System

```
Parameter: ρ;
Input: DAG G = (V, E);
initialize pheromones τ and best-so-far paths p*_1, ..., p*_n;
while termination criterion not met do
    for u = 1 to n in parallel do
        construct a simple path p_u from u to the sink
        w.r.t. τ;
        w ← evaluate(p_u);
        w* ← evaluate(p*_u);
        if w ≤ w* then p*_u ← p_u;
        update pheromones τ on all edges outgoing from
        u w.r.t. p*_u;
return (p_0, ..., p_i);
```

We will use the following lemma about pheromones in the MMAS taken from [21].

LEMMA 2 ([21, LEMMA 1]). *Let v be a vertex and τ be a the vector of pheromones in some iteration during some run of MMAS. Then the total pheromones τ_{total} on the edges going out from v is bounded from above by $1 + n\tau_{\min}$.*

In particular, any edge with pheromone τ is chosen with a probability at least $\tau/(1 + n\tau_{\min})$.

2.3 Bandit Setting

We consider the following definition for a stochastic online learning setting (see, for example, [4]). Given a finite set of possible *actions* and associated real-valued random variables $(X_a)_{a\in A}$ a *learner* is asked to repeatedly choose an action from A. With every choice $a \in A$, the learner incurs a *loss* drawn according to X_a (the draws of different iterations are assumed to be independent). The goal is to minimize the loss; the difference between the actual loss (after n iterations) and the loss of the optimal strategy of always choosing the action corresponding to the random variable with smallest expected value, is known as the *regret*.

Trivially, if the learner knows the distribution of the losses $(X_a)_{a\in A}$ at the start, then he can achieve an expected regret of 0. Instead, one looks at settings where only the set of possible actions and not the associated loss functions are given. There are several different settings for how a learner gathers information about the loss functions; the most interesting setting for our purposes is the *multi-armed bandit setting* [26], where, after each iteration, the learner gets to know the loss it incurred in the given iteration, but nothing about the alternatives. This poses a question regarding the exploration vs. exploitation trade off: should the learner choose an action known to have a small loss in order to keep the regret low, or try to find actions with even lower loss?

The case of finding the shortest link among m parallel links corresponds to choosing between one of m different actions. Since our ACO algorithm reevaluates its best-so-far solution, it requires choosing two actions for each iteration of the ACO.

The setting where several bandit problems are supposed to be solved in parallel, and feedback from one task might benefit the learner in other tasks (as in the stochastic single-destination shortest path problem), is called a *multitask bandit problem.*

For the multi-armed bandit setting, efficient algorithms are known which achieve a sublinear regret [25]. Furthermore, many variations of the settings of online learning have been studied, including with adversarial rather than stochastic loss [2], also in a setting of finding a shortest path [18]. One of the main interests of this work lies in finding efficient problem specific algorithms with low regret.

We consider MMAS optimizing m parallel links and compute the expected regret. Suppose one edge has deterministic weight 0 and all others deterministic weight 1. After an initial phase of finding the good edge and converging in pheromones to that edge, MMAS will sample a non-optimal edge with probability about $(1-n\tau_{\min})$ in every iteration. In each iteration where a non-optimal edge is chosen, MMAS incurs a regret of 1. Thus, the expected regret in t rounds is $\Theta(t(1 - \tau_{\max}))$, linear in the number of iterations. This shows that MMAS is inferior in the bandit setting to tailored algorithms.

Intuitively, MMAS focuses more strongly on exploration than problem specific algorithms. Note that MMAS will not converge to the right link at all if, for example, the link with highest expected value has a chance of less than $1/2$ of coming out higher when compared with any other link (see the example given in Proposition 8).

For the remainder of this paper we will not consider the bandit setting, but focus on the questions of whether and how fast MMAS converges to paths that are good in expectation.

3. PARALLEL LINKS

In this section, we analyze the most simple case of two nodes connected by several parallel edges. This will be the cornerstone of the results in the following sections.

Roughly speaking, we will show that, if there is an edge that looks shorter than any other edge with probability strictly greater than $1/2$, then this edge will attract the vast majority of the pheromone.

While seemingly a natural result, its proof is surprisingly hard. The main difference, e.g., compared to the previous work [22], is that now the preferred edge only gains pheromone at a higher rate than the other edges. In previous works, typically the MMAS once it found a good edge, it sticked to it and reinforced it until $\tau_{\max}$ was reached.

For this reason, we shall need arguments from drift analysis to bound the time needed until this process has brought the maximum pheromone to the preferred edge. Unfortunately, the progress of growth of pheromone on the preferred edge follows an uncommon pattern. It is strongest for mid-range pheromone values, but slow for both high and low pheromone values. To analyze such types of drift, we need to prove a variation of Johannsen's variable drift theorem that can deal with the situation that the drift is not monotonic with the distance to the optimum.

3.1 Result

In the remainder of this section, let $G = (V, E, X)$ be a stochastic-weight graph such that

- $V = \{v, s\}$,
- E consists of m parallel edges from v to the sink s,

- there is an $e \in E$ and a $\delta > 0$ such that for all $f \in E \setminus \{e\}$, $\Pr(X_e < X_f) \geq 0.5 + \delta$. We call e a *preferred* edge, since e looks shorter than each of the other edges with probability more than 0.5.

Note that the condition of being a preferred edge is more than just saying that X_e has the minimum expectation. Furthermore, it may also happen that none of the edges is preferred. Examples of this type can easily be constructed from the famous *non-transitive dice* example from elementary probability theory, see, e.g., Gardner [10].

Of course, we would hope that MMAS recognizes a preferred edge, if it exists, and ideally raises its pheromone value to the upper boundary $\tau_{\max}$. This is indeed true, though hard to prove.

THEOREM 3. *Suppose $\tau_{\max} + \tau_{\min} = 1$ and $2\rho \leq \tau_{\min}$. Then, after an expected number of $O(\frac{\log(1/\tau_{\min})}{\delta\rho})$ iterations, the pheromone value on the preferred edge e is $\tau_{\max}$.*

In Section 3.2 we give an important tool for our proof of Theorem 3, a new drift theorem allowing for drift which is not increasing with the distance to the optimum. Section 3.3 constitutes the proof of Theorem 3.

We will give a version of Theorem 3 applicable to arbitrary graphs in Section 5, namely Theorem 11; note, however, that Theorem 11 requires an additional restriction on δ which is not present in Theorem 3.

3.2 Proof via Improved Drift Analysis

The difficulty in proving Theorem 3 lies in the fact that, while e is preferred over other edges, this preference may be only slight. So, unlike with many proofs on the performance of MMAS, it does not happen that once e is found, MMAS does not reinforce the other alternatives anymore. Rather, with constant rate, it will happen that an alternative looks better, and that this is reinforced instead of e.

The current best method to deal with such situations is *drift analysis*, which was first introduced to the theory of evolutionary computation by He and Yao [19, 20]. In a nutshell, this method tries to suitably characterize the expected progress of a random process in one iteration. From this, via so-called drift theorems, one can deduce information about the expected time the process needs to reach a certain area in the state space.

While the (now so-called) *additive drift* of He and Yao is most suitable for situations in which the progress is uniform, the recently introduced multiplicative drift analysis [6] captures best the situation that the progress is proportional to the current distance from the goal. A generalization of this, applicable to the general situation that the progress is faster if the distance to the goal is larger, was proven in Johannsen's PhD thesis [23, Theorem 4.6] and is called *variable drift*.

Unfortunately, even this very general tool is not enough for our purposes. The problem is that the expected increase of the pheromone level of e is small both when e carries much pheromone (this stems from the pheromone update rule) and when e has little pheromone (because then the probability of sampling e is small). Hence, Johannsen's monotonicity assumption seems hard to fulfill.

We overcome this difficulty by proving a variable drift theorem that does not need the monotonicity assumption. We are optimistic that this new drift theorem will be useful both in other ACO problems and, more generally, for the analysis of other bio-inspired search heuristics.

THEOREM 4. *Let $(X_t)_{t\geq 0}$ be random variables describing a Markov process over a finite state space $0 \in S \subseteq \mathbb{R}_0^+$ and let $x_{\min} := \min\{x \in S \mid x > 0\}$. Furthermore, let T be the random variable that denotes the first point in time $t \in \mathbb{N}$ for which $X_t = 0$. Suppose that there exist $c \geq 1$, $d > 0$ and a continuous function $h : \mathbb{R}_0^+ \to \mathbb{R}^+$ such that*

- *for all $t < T$, $E(X_t - X_{t+1} \mid X_t) \geq h(X_t)$;*
- *for all $t < T$, $P(X_t - X_{t+1} \leq d) = 1$; and*
- *for all $x < y$ with $y - x \leq d$, we have $h(x) \leq c\, h(y)$.*

Then

$$E(T \mid X_0) \leq c\left(\frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(x)}dx\right).$$

Note that the case of $c = 1$ above just means that h is monotonic. For our purposes, we will need the statement for $c > 1$.

3.3 Analysis: Parallel Edges

Using the above new drift theorem, we can now analyze how the pheromone is accrued on the preferred edge e. As always in drift analysis, we have to define a suitable progress measure (potential function).

Recall that the state of the MMAS for our graph is fully described by the pheromone values of all edges and the best-so-far solution, which is one of the edges from E. For the purpose of our analysis, only the pheromone on e is important and whether e is the current best-so-far solution or not. We denote such a reduced state by a pair (τ, x), where τ is the present pheromone on e and x is set to 1 if e is the current best-so-far solution and 0 otherwise.

For such a state (τ, x), we define its *potential* by

$$g(\tau, x) = \tau + \rho x.$$

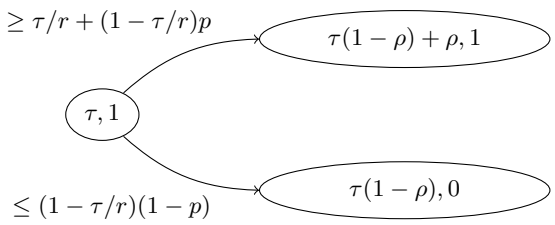
Hence, g basically measures the pheromone on e, giving a "bonus" of ρ when e is the current best-so-far solution. For this potential, we can show the following statement on its expected increase.

LEMMA 5. *Let $r = 1 + \varepsilon$ be an upper bound on the total pheromone on the edges of the graph. Let $\delta \geq \varepsilon$. Let MMAS be in a state with reduced state (τ, x). Let (τ', x') be (random variables denoting) the (reduced) state after one iteration, where the update of the pheromone values was made ignoring the pheromone boundaries $\tau_{\min}$ and $\tau_{\max}$. Then*

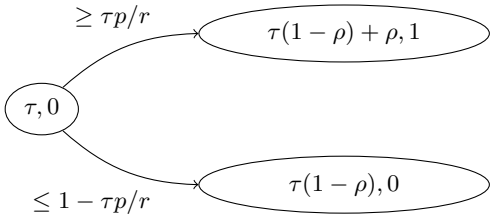
$$E[g(\tau', x') - g(\tau, x)] \geq \min(\tau, 1 - \tau)\rho\delta/2.$$

PROOF. To prove the lemma, we first make the underlying Markov chain precise, that is, given a state (τ, x), we find the transition probabilities of going to other states.

When $x = 1$, we sample e again with probability at least τ/r, which automatically implies $x = 1$ for the next state. If we sample some edge other than e (with probability at most $1 - \tau/r$), we can still have $x = 1$ in the next state with probability $p \geq 0.5 + \delta$, thanks to e having the better value in the direct comparison (see Definition 9). Thus, the probability of next going to a state with $x = 1$, given in the present state $x = 1$, is at least $\tau/r + (1 - \tau/r)p$. In this



(a) Best-So-Far=1



(b) Best-So-Far=0

Figure 1: Markov Chain

state, we have a positive reinforcement of the pheromone τ to $\tau(1-\rho)+\rho$.

With similar arguments, we find the probability of going to a state with $x = 0$, given the present has $x = 1$, is at most $(1-\tau/r)(1-p)$ and the pheromone value in that state is $\tau(1-\rho)$. In this case there is no positive reinforcement in the pheromone on e.

Similarly, if in the present state we have $x = 0$, then the probability of going to a new state with $x = 1$ is at least $\tau p/r$. With remaining probability of at most $1-\tau p/r$ we go to a state with $x = 0$. The pheromone values of the next states are the same as in the cases of $x = 1$. Fig. 1 depicts the different states and corresponding transitions of the Markov chain. The associated transition probabilities of going from one state to the other are also shown.

Given this chain, we can now prove the statement of the lemma. Assume first that $x = 1$. Then

$$\begin{aligned}
&E[g(\tau', x') - g(\tau, x)]\\
&= P(x' = 1)\ g(\tau(1-\rho)+\rho, 1) +\\
&\quad P(x' = 0)\ g(\tau(1-\rho), 0) -\\
&\quad g(\tau, 1)\\
&\geq (\tau/r + (1-\tau/r)p)(\tau(1-\rho)+2\rho) +\\
&\quad (1-\tau/r)(1-p)\tau(1-\rho) - (\tau+\rho)\\
&= \tau^2(1-\rho)/r + 2\tau\rho/r + \tau(1-\tau/r)p(1-\rho) +\\
&\quad 2(1-\tau/r)p\rho + \tau(1-\tau/r)(1-p)(1-\rho)\\
&\quad -(\tau+\rho)\\
&= \tau^2(1-\rho)/r + 2\tau\rho/r + \tau(1-\tau/r)(1-\rho)\\
&\quad +2(1-\tau/r)p\rho - (\tau+\rho)\\
&= 2\tau\rho/r + \tau(1-\rho) + 2(1-\tau/r)p\rho - (\tau+\rho)\\
&= (2p - 1 + \tau(2/r - 1 - 2p/r))\rho.
\end{aligned}$$

The last term is at least $(1-\tau)\rho\delta/2$ if the following two equations hold.

$$\begin{aligned}
2p - 1 &\geq \delta/2;\\
2p/r + 1 - 2/r &\geq \delta/2.
\end{aligned}$$

The first of these equations follows from $2p - 1 = \delta$; the second is implied by $\delta \geq \varepsilon$.

For $x = 0$, we have

$$\begin{aligned}
&E[g(\tau', x') - g(\tau, x)]\\
&= P(x' = 1)\ g(\tau(1-\rho)+\rho, 1) +\\
&\quad P(x' = 0)\ g(\tau(1-\rho), 0) -\\
&\quad g(\tau, 0)\\
&= (\tau p/r(\tau(1-\rho)+2\rho) + (1-\tau p/r)\tau(1-\rho) - \tau\\
&= \tau^2 p(1-\rho)/r + 2\tau p\rho/r + \tau(1-\rho) - \tau^2 p(1-\rho)/r - \tau\\
&= 2\tau p\rho/r - \tau\rho\\
&= \tau\rho(2p/r - 1).
\end{aligned}$$

The last term is at least $\tau\rho\delta/2$ if $2p/r - 1 \geq \delta/2$, which is implied by $\delta \geq \varepsilon$.

Thus, the drift is lower bounded by $\min(\tau, 1-\tau)\rho\delta/2$. □

For all t, let τ^t and x^t be the random variables denoting the pheromone in iteration t and the value of the best-so-far solution in iteration t, respectively. We define the random variables X^t by

$$X^t = \begin{cases} 1 - g(\tau^t, x^t) & \text{if } \tau^t < \tau_{\max} \\ 0 & \text{else} \end{cases}$$

for all t.

We want to use Theorem 4 with the drift computed in Lemma 5. To that end we let h be such that, for all z,

$$h(z) = \min(z, 1-z)\rho\delta/4.$$

Note that the drift computed in Lemma 5 does not directly apply to the Markov chain $(X^t)_t$, as the potential values of a state of the algorithm might be up to ρ different from its pheromone value τ. However, straightforward calculations with the result of Lemma 5 show that, for all z, $h(z)$ gives a lower bound for the drift of $(X^t)_t$.

The minimal x that we will use this on is $1 - \tau_{\max} - \rho = \tau_{\min} - \rho$, since $\tau_{\max} + \rho$ is the largest potential value of g. Furthermore we have, for all t,

$$g(\tau^t, x^t) - 2\rho \leq g(\tau^{t+1}, x^{t+1}) \leq g(\tau^t, x^t) + 2\rho.$$

This holds because, in one iteration, the pheromone value can change by at most ρ, and the value of x can change by at most 1, which changes the value of $g(\tau^{t+1}, x^{t+1})$ by at most 2ρ from the value of $g(\tau^t, x^t)$. A special case occurs when $\tau^{t+1} = \tau_{\max}$, the boundary case of the Markov chain $(X^t)_t$; the additional distance is $1 - \tau_{\max} - \rho = \rho$. However, it is easy to see that in this case we have $x^t = 1$, which means that the potential cannot be decreased by changing x from 0 to 1, so the maximum change in potential is 2ρ also in this case.

Thus, for the preconditions of Theorem 4, we need to show that there is a c' such that, for all x, y with $0 \leq x < y \leq 1 - \tau_{\min} = \tau_{\max}$ and $y - x \leq 2\rho$, $h(x) \leq c'h(y)$. We let $c' = 2$. Let x, y with $1 - \tau_{\max} - \rho \leq x < y \leq 1 - \tau_{\min}$ be such that $y - x \leq 2\rho$. The claim is trivial if $y \leq 1/2$, so

suppose $y > 1/2$. We have $2y - x \leq y + 2\rho \leq 1$; hence, $1 - x \leq 2(1 - y)$. Therefore,

$$h(x) \leq (1-x)\rho\delta/4 \leq 2(1-y)\rho\delta/4 = 2h(y).$$

This fulfills the preconditions of Theorem 4 as desired.

By Theorem 4, the expected time until the pheromone value on the preferred edge e is at $\tau_{\max}$, is at most

$$\begin{aligned}
&2\left(\frac{1}{\rho\delta} + \int_{\tau_{\min}-\rho}^{\tau_{\max}} \frac{4}{\min(x,(1-x))\rho\delta} dx\right) \\
&= \frac{2}{\rho\delta} + \frac{8}{\rho\delta}\left(\int_{\tau_{\min}-\rho}^{\frac{1}{2}} \frac{1}{x} dx + \int_{\frac{1}{2}}^{\tau_{\max}} \frac{1}{1-x} dx\right) \\
&\leq \frac{2}{\rho\delta} + \frac{16}{\rho\delta}\int_{\tau_{\min}-\rho}^{\frac{1}{2}} \frac{1}{x} dx \\
&= O\left(\frac{\log(1/\tau_{\min})}{\rho\delta}\right).
\end{aligned}$$

4. EXAMPLES

Next we give some illustrative examples of stochastic-weight graphs, where MMAS converges as desired (using the result of the previous section). The first example concerns normally distributed random variables as edge lengths. Together with the previous section, Proposition 6 shows that, for parallel links with normally distributed lengths, MMAS will converge in pheromone to the edge with shortest expected value (given δ large enough).

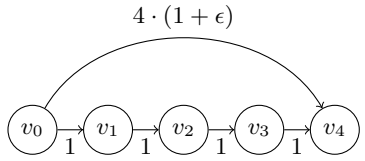
PROPOSITION 6. *Consider two normal distributed random variables $X \sim N(\mu, \sigma_1^2)$ and $Y \sim N(\mu(1+\epsilon), \sigma_2^2)$, where μ, ϵ, σ_1 and $\sigma_2 \in \mathbb{R}^+$. Then, $P(X \leq Y) = 0.5 + \delta$, where*

$$\delta = \Omega\left(\frac{\mu\epsilon}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right).$$

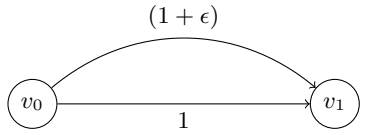
We denote the gamma distribution with shape parameter $k > 0$ and scale parameter $\theta > 0$ as $\Gamma(k, \theta)$.

The paper [22] gives a family of instances for stochastic-weight graphs where MMAS *without* reevaluation of the best-so-far path requires a superpolynomial number of iterations for finding the paths that are shortest in expectation. The random weights in these instances are based on the gamma distribution. The core difficulty for these instances lies in differentiating correctly whether to use a single long edge to the goal, or many short edges, which are just a bit shorter than the long edge in total; see Figure 2 for an illustration of two cases with $n \in \{1, 4\}$ short edges. The edge weights w are subject to a gamma distributed noise, such that the stochastic length is $w(1 + \Gamma(k, \theta))$, where the parameters k and θ are the same over all edges of the graph.

In [22] it was shown that MMAS without reevaluation will be deceived by the variance of the single long edge, and converge to it in pheromone values. In Proposition 7 we show, for many relevant parameters, that the sum of the short edges (corresponding to the random variable X in the proposition) is likely to come out better than the long edge (corresponding to the random variable Y). Together with the result from Section 3, this implies that many of the instances given in [22] which MMAS cannot solve without reevaluation, *can* be solved using reevaluation. In particular, we have that MMAS with reevaluation on instances exemplified in Figure 2 will converge (as desired) to the lower edges (following the lower chain depicted in Figure 2).



(a) Example graph with n=4



(b) Example graph with n=1

Figure 2: Example graphs

PROPOSITION 7. *Let m be a natural number and $k, \theta, \varepsilon > 0$. Suppose $k\varepsilon^2 \geq 16$. For $i \leq m$, let $X_i \sim 1 + \Gamma(k, \theta)$ be i.i.d. and let $X \sim \sum_{i=1}^{m} X_i$. Furthermore, let $Y \sim m(1 + \varepsilon)(1 + \Gamma(k, \theta))$ be independent of all X_i. Then*

$$P(X \leq Y) \geq 9/16.$$

In general, when comparing two random variables, a smaller expected value does not imply that it is more likely to come out smaller. Instead, there are two random variables where the first one has an expected value arbitrarily higher than the second, and still comes out better with an arbitrarily good probability. This is formalized in the following proposition.

PROPOSITION 8. *For all $c > 1$ and q with $0 < q < 1$, there are X, Y such that $E(X) = cE(Y)$ and $P(X < Y) = q$.*

5. GENERAL GRAPHS

In this section we formally analyze the behavior of MMAS on well-behaved stochastic-weight graphs.

DEFINITION 9. *Let $G = (V, E, X)$ be a stochastic-weight graph and let $\delta > 0$. For all $i \leq n$, we inductively define sets of edges A_δ^i such that $A_\delta^0 = \emptyset$ and an edge e from v to w is an element of A_δ^{i+1} if and only if there is a path p from v to the sink using only edges from $A_\delta^i \cup \{e\}$ such that p contains e and, for any path p' from v to the sink where p' does not use e, $P(X_p < X_{p'}) \geq 0.5 + \delta$. We call edges from A_δ^n δ-preferred edges.*

In Theorem 11 below we will show that the name "preferred edge" has its justification in the fact that MMAS will prefer to take these edges (after an initial phase to find these edges). But first we will give an example of preferred edges in a stochastic-weight graph where all edge weights are normally distributed; Proposition 10 shows that, in this case, the set of preferred edges corresponds to the tree of paths of shortest expected length.

PROPOSITION 10. *Let $G = (V, E, X)$ be a stochastic-weight graph such that, for all edges $e \in E$, X_e is normally distributed. Furthermore, suppose that the graph derived from G by replacing all random weights by their expected value has a unique shortest path tree S. Then, for δ small enough, the set of δ-preferred edges A^n_δ is the tree S.*

We now get to the main theorem of the paper, concerning the behavior of MMAS on general graphs.

THEOREM 11. *Let $G = (V, E, X)$ be a stochastic-weight graph with n vertices and suppose $\tau_{\min} = O(1/n^3)$ and $\rho = \tau_{\min}/2$. Then there is a constant c such that, for δ such that the set of δ-preferred edges $A^n_\delta \subseteq E$ is a tree and $\delta \geq cn^2\tau_{\min}$, after*

$$O\left(\frac{n\log(1/\tau_{\min})}{\delta\rho}\right)$$

iterations, MMAS on G has a pheromone of $1 - O(n\tau_{\min})$ on all edges of A^n_δ and $O(n\tau_{\min})$ on all others.

Acknowledgement

We thank one anonymous reviewer for pointing us to Gutjahr's work [13, 14] on the S-ACO algorithm, which we (and the authors of [22]) unfortunately overlooked. Ashish Ranjan Hota was gratefully supported by the German Academic Exchange Services (DAAD) through a WISE stipend for undertaking an internship at the Max Planck Institute for Informatics, Saarbrücken, Germany.

6. REFERENCES

[1] 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*, 105:88–92, 2008.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, 32:48–77, 2003.

[3] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8:239–287, 2009.

[4] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[5] B. Doerr, M. Fouz, and C. Witt. Sharp bounds by probability-generating functions and variable drift. In *Genetic and Evolutionary Computation Conference (GECCO'11)*, pages 2083–2090. ACM, 2011.

[6] B. Doerr, D. Johannsen, and C. Winzen. Multiplicative drift analysis. In *Genetic and Evolutionary Computation Conference (GECCO'10)*, pages 1449–1456. ACM, 2010.

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

[8] M. Dorigo. *Optimization, Learning and Natural Algorithms (in Italian)*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.

[9] M. Dorigo and T. Stützle. *Ant colony optimization*. MIT Press, 2004.

[10] M. Gardner. Nontransitive dice and other probability paradoxes. *Scientific American*, 223:110–114, 1970.

[11] W. J. Gutjahr. A graph-based ant system and its convergence. *Future Generation Comper Systems*, 16:873–888, 2000.

[12] W. J. Gutjahr. ACO algorithms with guaranteed convergence to the optimal solution. *Information Processessing Letters*, 82:145–153, 2002.

[13] W. J. Gutjahr. A converging ACO algorithm for stochastic combinatorial optimization. In *Symposium on Stochastic Algorithms: Foundations and Applications (SAGA'03)*, pages 10–25, 2003.

[14] W. J. Gutjahr. S-ACO: An ant-based approach to combinatorial optimization under uncertainty. In *ANTS*, pages 238–249. Springer, 2004.

[15] W. J. Gutjahr. Mathematical runtime analysis of aco algorithms: survey on an emerging issue. *Swarm Intelligence*, 1:59–79, 2007.

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

[17] 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.

[18] A. György, T. Linder, G. Lugosi, and G. Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8:2369–2403, 2007.

[19] J. He and X. Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127:57–85, 2001.

[20] J. He and X. Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3:21–35, 2004.

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

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

[23] D. Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics*. PhD thesis, Universität des Saarlandes, 2010. Available online at http://scidok.sulb.uni-saarland.de/volltexte/2011/3529/pdf/Dissertation_3166_Joha_Dani_2010.pdf.

[24] T. Kötzing, F. Neumann, D. Sudholt, and M. Wagner. Simple max-min ant systems and the optimization of linear pseudo-boolean functions. In *Foundations of Genetic Algorithms (FOGA'11)*, pages 209–218. ACM, 2011.

[25] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4 – 22, 1985.

[26] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

[27] T. Stützle and H. H. Hoos. MAX-MIN ant system. *Future Generation Computer Systems*, 16:889–914, 2000.