Technical University of Denmark

DTU

Runtime analysis of ant colony optimization on dynamic shortest path problems

Lissovoi, Andrei; Witt, Carsten

DTU Library
Technical Information Center of Denmark

Runtime Analysis of Ant Colony Optimization on Dynamic Shortest Path Problems[1]

Andrei Lissovoi[a], Carsten Witt[a]

[a]*DTU Compute, Technical University of Denmark, Denmark*

Abstract

A simple ACO algorithm called λ-MMAS for dynamic variants of the single-destination shortest paths problem is studied by rigorous runtime analyses. Building upon previous results for the special case of 1-MMAS, it is studied to what extent an enlarged colony using λ ants per vertex helps in tracking an oscillating optimum. It is shown that easy cases of oscillations can be tracked by a constant number of ants. However, the paper also identifies more involved oscillations that with overwhelming probability cannot be tracked with any polynomial-size colony. Finally, parameters of dynamic shortest-path problems which make the optimum difficult to track are discussed. Experiments illustrate theoretical findings and conjectures.

Keywords: Ant Colony Optimization; Shortest Paths; Dynamic Problems; Runtime Analysis

1. Introduction

Ant colony optimization (ACO) is a class of nature-inspired algorithms that is mostly used to solve combinatorial optimization problems. In recent years, runtime analysis of nature-inspired algorithms has advanced considerably (Auger and Doerr, 2011; Jansen, 2013; Neumann and Witt, 2010b). Even though the majority of results still apply to simple evolutionary algorithms, a lot of progress has also been made in the analysis of ACO. With respect to problems from combinatorial optimization, which is the classical domain of application for ACO, there are results on shortest paths (Attiratanasunthron and Fakcharoenphol, 2008; Sudholt and Thyssen, 2012a), minimum spanning trees (Neumann and Witt, 2010a), minimum cuts (Kötzing et al., 2010) and the traveling salesperson problem (Zhou, 2009; Kötzing, Neumann, Röglin and Witt, 2012).

Real-world optimization problems are not always static in nature. Often problem and goal of optimization are dynamic, i.e., they change over time. In these cases, it is important to find a solution that is "good" with respect to the current goal of optimization.

---

[1]A preliminary version of this work appeared in *GECCO '13: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, pages 1605-1612, ACM, New York, USA, 2013.

Email addresses: `andl@dtu.dk` (Andrei Lissovoi), `cawi@dtu.dk` (Carsten Witt)

Nature-inspired algorithms are often considered to be "robust" optimizers which can adapt to such dynamic problems if the underlying optimal solution does not change too quickly or extremely. In particular, many applications of evolutionary algorithms, as well as ACO, on dynamic problems are reported in the literature (Garlick and Barr, 2002; Xiang and Lee, 2008; Nguyen et al., 2012).

In this paper, we are concerned with ACO on dynamic problems. Our aim is to understand, using rigorous runtime analyses, the conditions under which ACO algorithms are able to track the optimum of a dynamically changing problem, i.e. maintain the ability to construct optimum, or close-to-optimum solutions while the fitness function changes. Such runtime analyses are motivated by related theoretical studies of evolutionary algorithms for dynamic problems (Rohlfshagen et al., 2009; Jansen and Schellbach, 2005; Droste, 2003). We have chosen the single-destination shortest path problem (SDSP) as object of our analysis as this is probably the combinatorial optimization problem that ACO has been understood best on. There are even runtime analyses of ACO on stochastic optimization problems (Sudholt and Thyssen, 2012b; Doerr et al., 2012; Feldmann and Kötzing, 2013), which, together with dynamic problems, can be subsumed under the term "optimization under uncertainty". However, methods for the analysis of stochastic optimization problems are not directly applicable to dynamic problems.

So far, there is only a single runtime analysis of ACO on dynamic problems. Kötzing and Molter (2012) compare a simple ACO algorithm and a simple evolutionary algorithm on a dynamic pseudo-boolean problem and show that the ACO algorithm can outperform the evolutionary one. To the best of our knowledge, our work is the first runtime analysis of ACO on a dynamic combinatorial optimization problem.

Our findings can be summarized as follows. A simple ACO algorithm based on the Max-Min Ant System (Stützle and Hoos, 2000) is studied on dynamic shortest path problems with increasing amount of dynamics. First it is analyzed how long it takes the system to adapt to a one-time modification of the graph, observing that if modifications to the weight functions occur slower than this, the dynamic optimization process can be treated as a series of adaptations to one-time changes. It is proved that two extreme cases can happen: only a single pheromone value or all pheromone values might need to be updated to adapt. Upper and lower bounds on the time required are proved. Then, more rapid periodic changes are studied by considering changing between two different functions in every iteration as an extreme case. Examples are shown where the changes exhibit enough locality for the system to track them reliably by updating independent components, maintaining at least a constant probability of constructing the optimum solution in any given iteration. Interestingly, this is possible by increasing the size of the ant colony moderately. The utility of a population for tracking problems was studied in evolutionary computation by Jansen and Schellbach (2005), but our result seems the first of this kind in the runtime analysis of ACO. It is then proved that a single ant started at each vertex is sufficient to track a slower oscillation in the same setting. Finally, an example is given where the problem changes globally, which makes it very unlikely that changes can be tracked quickly. Experiments supplement the theoretical findings, and we discuss properties of the dynamics that are related to the difficulty of the tracking problem.

The paper is structured as follows. Section 2 introduces notation and the algorithm λ-MMAS, which generalizes previously studied MMAS by introducing a larger population. Section 3 proves polynomial lower and upper bounds on the time for the system to adapt to one-time changes. Periodic changes with locality are studied in Section 4, where it is shown that the oscillating optimum can be tracked by the ant colony for a super-polynomial number of iterations, both for rapid oscillations using a larger colony, and for slower oscillations using a single ant. Section 5 describes a globally changing example that is conjectured to be difficult to track. Experiments are described in Section 6. We finish with some conclusions.

2. Preliminaries

The λ-MMAS algorithm (Algorithm 1) is a generalized version of the MMAS$_{\text{SDSP}}$ analyzed on shortest path problems in Sudholt and Thyssen (2012a), where the generalization is due to a use of a population (see Neumann et al., 2010 for a related algorithm in pseudo-boolean optimization). The algorithm allows for negative weights, but requires that all cycles in the graph are of strictly positive weight. We note that graphs with $\Delta < 2$ are not interesting for shortest path problems, and assume that $\Delta \geq 2$ throughout the rest of the paper: otherwise, each vertex has at most one outgoing edge, and finding the shortest paths is trivial.

Every iteration of the algorithm starts λ ants at each vertex, and each ant constructs a simple path through the graph, following arcs randomly with probability proportional to their pheromone values until it reaches the destination vertex t, or until it reaches a vertex from which no arcs lead to a vertex that has not already been visited. For each vertex v, the best-so-far path x_v^* is updated to the path of least weight of the λ paths constructed from v in the current iteration and the previous x_v^* path, and is then used to update the pheromone values on arcs leaving v.

The algorithm needs to start at least one ant at each vertex, even if the goal is to find the shortest path between a specific pair of vertices, in order to ensure that the shortest paths can be found in expected polynomial time. The issue is illustrated in Sudholt and Thyssen (2012a) using the graph shown in Figure 1: if the weight on the (r, t) arc is $n = |V|$ and all other arcs have unit weights, the vertex r is avoided by the s-t shortest path. An ant started at s will, with probability $1 - (1/2)^{n/2}$, make no more than $n/2$ steps towards t (each taken with probability $1/2$ as the pheromone values are initialized to $1/deg^+(v) = 1/2$, and sum to 1 for each vertex) before visiting r. Subsequent iterations will accept a path that makes additional steps towards t only if it reaches t without visiting r, which occurs with probability at most $(1/2)^{n/2}$ (as each of the additional $n/2$ steps toward t is made with probability at most $1/2$); discovery, acceptance, and pheromone updates based on paths that take even fewer steps towards t before visiting r will only decrease the probability of reaching t without visiting r over time.

Three parameters affect the behavior of the algorithm: the evaporation rate ρ controls the speed with which the pheromone values are updated, while the pheromone bounds $\tau_{\min}$ and $\tau_{\max}$ control how likely an ant is to deviate from a pheromone trail, balancing exploration with the ability to follow reinforced paths. As in Sudholt and Thyssen (2012a), we choose

3

Algorithm 1 The λ-MMAS algorithm on a directed graph $G = (V, \mathcal{A})$, with pheromone bounds $\tau_{\min}$ and $\tau_{\max}$, and evaporation rate ρ, where $t \in V$ is the destination vertex, $\deg^+(v)$ is the outdegree of vertex v, and $f(x_v)$ is the fitness value of a given path (sum of its arc weights if the path terminates at t, ∞ if it does not).

> Initialize $\tau_a \leftarrow 1/\deg^+(v)$ for all $a = (v, v') \in \mathcal{A}$
> **for** $i \leftarrow 1, 2, \ldots$ **do**
>   **for each** $v \in V$ **do**
>   **for** $j = 1, 2, \ldots, \lambda$ **do**
>   Let $x_{v,j}$ be an empty path starting at v
> $p \leftarrow v, \; S \leftarrow \big\{ (p, v') \in \mathcal{A} \mid v' \notin x_{v,j} \big\}$
>   **while** S is not empty and $p \neq t$ **do**
>   Pick arc $a = (p, h')$ from S with probability:
> $p_a = \tau_a / \sum_{s \in S} \tau_s$
>   Append a to $x_{v,j}$
> $p \leftarrow h', \; S \leftarrow \big\{ (p, v') \in \mathcal{A} \mid v' \notin x_{v,j} \big\}$
> $x_v \leftarrow \arg\min_{x_{v,j}} f(x_{v,j})$
>   **if** $i = 1$ **or** $f(x_v) < f(x_v^*)$ **then**
> $x_v^* \leftarrow x_v$
>   **for each** $a = (v, v') \in \mathcal{A}$ **do**
> $\tau_a \leftarrow \begin{cases} \min(\tau_{\max}, (1-\rho)\tau_a + \rho) & \text{if } a \in x_v^* \\ \max(\tau_{\min}, (1-\rho)\tau_a) & \text{otherwise} \end{cases}$

the pheromone bounds are chosen based on Δ, the maximum vertex out-degree, and ℓ, the maximum number of arcs in any shortest path in the graph:

$$\tau_{\min} = 1/(\Delta \ell) \qquad\qquad \tau_{\max} = 1 - \tau_{\min}$$

where $n = |V|$ can be used in place of Δ or ℓ if either value is unknown. With this choice of pheromone values, an ant is able to follow a reinforced path of $\ell - 1$ arcs (where the pheromone values on the arcs in the path are all $\tau_{\max}$, and on all other outgoing arcs from vertices visited by the path are $\tau_{\min}$) with probability at least $1/e$. This is proved in Sudholt and Thyssen (2012a) by considering the probability of not deviating from the path at each
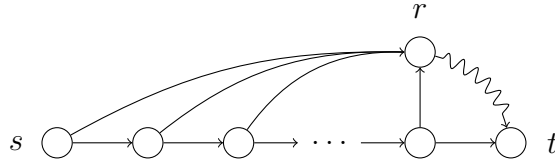

Figure 1: An example graph; the weight on the (r,t) arc affects whether the vertex r is visited or avoided by the shortest paths to t.

vertex, which is at least $(1 - \Delta\tau_{\min})^{\ell-1} \geq (1 - 1/\ell)^{\ell-1} \geq 1/e$, as the pheromones sum to at least $\tau_{\min} + \tau_{\max} = 1$ whenever there's an opportunity to deviate from the reinforced path.

The Lemma 1 is heavily used to study the probabilistic pheromone model, and is proved in Sudholt and Thyssen (2012a). As λ-MMAS selects the next arcs with probability equal to the arc's pheromone value divided by the sum of pheromone values on viable arcs leaving the vertex, the lemma allows us to bound the probability that an ant will follow any specific outgoing arc to a vertex it has not visited before by at least $\tau_{\min}/2$.

Lemma 1. *The sum of pheromone values on all outgoing arcs from any vertex is always at most 2.*

When the first shortest path from vertex v is constructed by an ant, it will be used as the best-so-far path x_v^* in every subsequent pheromone update. The following Lemma from Attiratanasunthron and Fakcharoenphol (2008) bounds the freezing time – the number of iterations of reinforcing a single arc before the pheromone values reach the pheromone bounds, at which point they will remain unchanged by subsequent pheromone updates until a new best-so-far path, using a different arc to leave v, is discovered. The freezing time is also a bound on the number of iterations between the discovery of a shortest path and the pheromone value on its first arc reaching $\tau_{\max}$.

Lemma 2. *If x_v^* is unchanged for $\ln(\tau_{max}/\tau_{min})/\rho$ iterations, the pheromone value on the first arc of x_v^* is τ_{max}, and equal to τ_{min} on all other arcs leaving v.*

Proof. Consider the effect of $\ln(\tau_{\max}/\tau_{\min})/\rho$ pheromone updates on the pheromone value on an arc. It is easy to see that that this number of pheromone updates is sufficient to reduce a pheromone value to $\tau_{\min}$, even if it was originally $\tau_{\max}$:

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

Consider two pheromone values, initially set to $\tau_{\max}$ and $\tau_{\min}$; while the first evaporates, the second is reinforced. The sum of the two values is initially 1, and remains 1 as multiplying both values by $(1-\rho)$ is exactly balanced by adding ρ to the second value. Thus the number of iterations required to increase a pheromone value from $\tau_{\min}$ to $\tau_{\max}$ is equal to the number of iterations required to reduce it from $\tau_{\max}$ to $\tau_{\min}$.

Therefore, $\ln(\tau_{\max}/\tau_{\min})/\rho$ iterations of reinforcing a single arc is sufficient to increase its pheromone value to $\tau_{\max}$, while also reducing the pheromone values on all other arcs exiting its source vertex to $\tau_{\min}$. □

The next section examines how λ-MMAS is able to handle a one-time change to the weight function, and motivates using $\lambda > 1$ ants to reduce the expected number of iterations needed to discover the shortest paths after a one-time change. Following sections illustrate the benefits and limitations of pheromone memory in a setting where the weight function is changed more frequently: it allows a relatively small number of ants to keep track of shortest paths if the differences between the weight functions are relatively minor, but requires a super-polynomial number of ants if the changes are more significant.

5

3. A one-time change

If the oscillation is sufficiently slow, λ-MMAS may be able to rediscover and re-freeze all of the shortest paths before the next change to the weight function occurs. If this is the case, the process can be treated as a series of one-time modifications to the weight function, with each modification followed by a number iterations during which λ-MMAS rediscovers the shortest paths, and a number of iterations during which the pheromones are frozen to favor the shortest paths, resulting in each ant having at least a constant probability of constructing the shortest path from its starting vertex.

The expected number of iterations λ-MMAS needs to discover the shortest paths of the new weight function depends both on how similar these shortest paths are to those of the previous weight function, and on the maximum number of arcs in any shortest path in the graph using the new weight function. The more similar the shortest paths, the fewer pheromone values need to be changed; and the fewer arcs in the longest shortest path, the more shortest paths can be discovered in parallel.

The proof of the following theorem is based on the analysis in Sudholt and Thyssen (2012a). The presentation here is simplified; a finer analysis in Sudholt and Thyssen (2012a) demonstrates that it is not necessary to wait for the full freezing time, removing the log factor from the freezing time component of the expected number of iterations.

Theorem 3. *With high probability, 1-MMAS will after $O(\ell^*/\tau_{min} + \ell \ln(\tau_{max}/\tau_{min})/\rho)$ iterations have discovered all shortest paths after a one-time change to the weight function given that such shortest paths are unique, where $\ell^* = \max(\ell, \log n)$ and ℓ is the maximum number of arcs in any shortest path to t in the new graph. This is also the expected number of iterations before all the shortest paths are rediscovered after the weight function is altered.*

Proof. Consider an arbitrary vertex v: the shortest path from v to t has at most $\ell \leq \ell^*$ arcs. λ-MMAS can discover the shortest path from v to t by constructing all of its subpaths (ending at t) in the order of increasing number of arcs. This allows shortest paths to be discovered by only constructing subpaths with one arc with a sub-τ_{max} pheromone value at a time, waiting for the pheromones to freeze, and repeating the process for the next subpath.

Let p_{path} be the probability that a specific ant constructs a specific k-arc path containing a single sub-τ_{max} arc. In order to discover the v-t shortest path, at most ℓ^* subpaths, all with $k \leq \ell^*$, need to be constructed (and frozen). Assuming the pheromone values on the shorter subpaths are frozen at τ_{max}, the probability of discovering a new shortest path in this fashion is at least p_{path}:

$$p_{path} \geq \frac{\tau_{min}}{2} \cdot (1 - \Delta\tau_{min})^{k-1} \geq \frac{\tau_{min}}{2e}$$

where $1 - \Delta\tau_{min} \geq 1 - 1/\ell$ is the probability of selecting the arc with pheromone value τ_{max} at any frozen vertex, recalling that the pheromone bounds τ_{min} and τ_{max} were chosen so as to ensure that the probability of following $\ell - 1$ such arcs was at least $1/e$.

The number of the desired subpaths discovered in this fashion can be bounded using a Chernoff bound: if every considered iteration is able to discover the next subpath with

6

probability p_{path}, the ℓ^* desired subpaths are discovered in $t = 10e\ell^*/\tau_{\text{min}}$ iterations with high probability. Let N_t be the number of subpath discoveries in t iterations, then $\mu = E(N_t) = t \cdot p_{\text{path}} \geq 5\ell^*$ (treating N_t as binomially distributed), and apply the Chernoff bound:

$$P(N_t < \ell^*) = P(N_t < (1 - 4/5) \cdot \mu)$$
$$< e^{-5\ell^* \cdot (4/5)^2/3} = O\big(n^{-16/15}\big)$$

inserting $\ell^* \geq \log n$ in the last step.

At most $\ell \ln(\tau_{\text{max}}/\tau_{\text{min}})/\rho$ additional iterations are required to freeze the pheromone values after each discovery to preserve the p_{min} lower bound on the probability of discovering the next subpath in each iteration. Thus a shortest path from an arbitrary v to t is discovered in at most $10e\ell^*/\tau_{\text{min}} + \ell \ln(\tau_{\text{max}}/\tau_{\text{min}})/\rho$ iterations with high probability. Shortest paths from all of the n vertices in the graph are also found with high probability, which can be shown by applying a union bound:

$$P(\min_{v \in V} N_t \geq \ell^*) \geq 1 - (1 - P(N_t < \ell^*))^n$$
$$\geq 1 - nP(N_t < \ell^*)$$
$$= 1 - O\big(n^{-1/15}\big)$$

Treating this as a success probability in a geometric distribution, it is clear that, in expectation, no more than $O(1)$ phases of $10\ell^*/\tau_{\text{min}} + \ell \ln(\tau_{\text{max}}/\tau_{\text{min}})/\rho$ iterations each are required. Thus, the expected number of iterations before all shortest paths are rediscovered after the weight function changed is $O(\ell^*/\tau_{\text{min}} + \ell \ln(\tau_{\text{max}}/\tau_{\text{min}})/\rho)$. $\qquad\square$

As remarked in Sudholt and Thyssen (2012a), the unique shortest paths constraint can be dropped at the cost of introducing a $\log n$ factor to the first component of the bound, by requiring in the analysis that all shortest paths of length i are found and frozen before λ-MMAS is considered to have a change to discover a shortest path of length $i + 1$; the expected time until all shortest paths of a particular length are found (given that all shorter paths have been found and frozen) is dominated by the coupon collector problem, and is therefore $O(\log n/\tau_{\text{min}})$; in total, there are no more than ℓ^* such phases.

To illustrate the potential effects of a one time change, consider the graph in Figure 1 and the weight functions w_1 and w_2 shown below; the shortest paths of w_1 visit r immediately, while w_2 avoids r if possible:

$$w_1(\alpha) = \begin{cases} -1 & \text{if } \alpha = (r, t) \\ 1 & \text{otherwise} \end{cases} \qquad\qquad w_2(\alpha) = \begin{cases} n & \text{if } \alpha = (r, t) \\ 1 & \text{otherwise} \end{cases}$$

If the pheromone values are frozen to the shortest paths using w_1, λ-MMAS will require a large number of iterations to rediscover all of the shortest paths when the weight function is changed to w_2. The proof of the following lower bound is also inspired by a related result in Sudholt and Thyssen (2012a), which allows for even finer bounds, but considers a uniform initialization of pheromone values, which is not given after λ-MMAS has frozen the pheromones to favor specific shortest paths.

7

Theorem 4. *One-time changes to the weight function may require an expected* $\Omega(\ell/\tau_{min})$ *iterations for 1-MMAS with* $\rho = 1$ *to discover all shortest paths, where* ℓ *is the maximum number of arcs in any shortest path to t in the new graph.*

Proof. The theorem is proved by example: we'll show that if, on the graph of Figure 1, all shortest paths under w_1 are discovered and frozen, $\Omega(\ell/\tau_{\min})$ iterations are required to rediscover the shortest paths after the weight function is switched to w_2. Given the high evaporation rate ($\rho = 1$), after the first iteration, all pheromone values are at the pheromone bounds, and switch between those instantly whenever new shortest paths are discovered by λ-MMAS. Assume for a moment that the optimization proceeds as in the proof of Theorem 3: only shortest paths with a single non-$\tau_{\max}$ arc are discovered until shortest paths from all vertices have been found.

The probability that the next subpath (with only one non-$\tau_{\max}$ arc) is discovered in a single iteration is at most $\tau_{\min}$. Let T_i be the number of iterations between the discoveries of subpaths $i-1$ and i; as $\Delta \geq 2$ and hence $\tau_{\min} \leq 1/2$, the probability that the next subpath is not discovered for at least $1/(2\tau_{\min})$ iterations is:

$$P(T_i > 1/(2\tau_{\min})) \geq (1 - \tau_{\min})^{1/(2\tau_{\min})}$$
$$= \sqrt{(1 - \tau_{\min})^{1/\tau_{\min}}} \geq 0.5$$

When the weight function is changed to w_2, the entire ℓ-arc longest shortest path needs to be rediscovered, changing the pheromone values on the outgoing arcs from each of the ℓ non-trivial vertices. Let N be the number of subpaths for which $T_i > 1/(2\tau_{\min})$, and $\mu = E(N) \geq \ell/2$. A Chernoff bound shows that at least a quarter of the subpaths will require more than $1/(2\tau_{\min})$ iterations each with overwhelming probability:

$$P(N \geq \ell/4) = 1 - P(N < (1 - 1/2)\mu)$$
$$> 1 - e^{-\mu/2^2/3} \geq 1 - e^{-\ell/24}$$

Thus, if no shortest paths with more than one non-$\tau_{\max}$ arc are discovered, $\Omega(\ell/\tau_{\min})$ iterations are required with overwhelming probability.

How likely is λ-MMAS not to find any shortest paths with more than one non-$\tau_{\max}$ arc? The probability p_{f} of discovering such a path is greatest at the beginning of the process, when shortest paths with 2, 3, ..., ℓ non-$\tau_{\max}$ arcs can be discovered; and the probability that no such paths are found for $1/\tau_{\min}^2$ iterations is at least a constant:

$$p_{\mathrm{f}} \leq \tau_{\min}^2 \sum_{i=2}^{\ell} \tau_{\min}^{i-2} < 2 \cdot \tau_{\min}^2$$
$$(1 - p_{\mathrm{f}})^{1/\tau_{\min}^2} \geq 1/16$$

assuming $\tau_{\min} \leq 1/2$ as before.

Thus, with at least constant probability, λ-MMAS does not discover any shortest paths with more than one non-$\tau_{\max}$ arc in $1/\tau_{\min}^2 \geq \ell/\tau_{\min}$ iterations, and $\Omega(\ell/\tau_{\min})$ iterations are

required to rediscover the shortest paths with overwhelming probability if only subpaths with a single non-$\tau_{\max}$ arc are discovered. Therefore, the expected number of iterations before the shortest paths are rediscovered by λ-MMAS (with $\lambda = 1$ and $\rho = 1$) after switching from w_1 to w_2 is $\Omega(\ell/\tau_{\max})$. $\qquad\square$

Conversely, there also exist combinations of weight functions for which the shortest paths may be rediscovered relatively quickly – for example, switching from w_2 to w_1 on the graph of Figure 1 is easier than the alternative, as a low $\ell = 2$ value reduces the interdependency between shortest paths, allowing them to be discovered in parallel without requiring ants to select more than one non-$\tau_{\max}$ arc at a time. Additionally, changing the weight of the (r, t) arc to values between the extremes considered in w_1 and w_2 would change only some of the shortest paths, reducing the amount of pheromone values that need to be updated.

These analyses suggest that λ-MMAS spends most of the iterations *waiting* to discover the next subpath. The expected number of iterations spent waiting can be reduced by increasing the number of ants simulated in each iteration, as expanded upon in Theorem 5; this is similar in effect to increasing the offspring population size in a $(1+\lambda)$ EA, as considered by Jansen et al. (2005).

Theorem 5. *Using $\lambda = 2e/\tau_{min}$ ants allows λ-MMAS to discover new shortest paths in expected constant time, allowing all shortest paths to be rediscovered in $O(\ell + \ell \ln(\tau_{max}/\tau_{min})/\rho)$ iterations.*

Proof. The probability that a single ant will construct a specific path with only one non-$\tau_{\max}$ arc is at least $\tau_{\min}/2e$, so the probability that at least one of λ ants constructs it is at least p_c:

$$p_c \geq 1 - (1 - \tau_{\min}/2e)^{2e/\tau_{\min}} \geq 1 - 1/e$$

Shortest paths up to ℓ arcs long may therefore be discovered using the same mechanism as in Theorem 3, with each subpath taking an expected $e/(e-1) = O(1)$ iterations to discover, and $\ln(\tau_{\max}/\tau_{\min})/\rho$ additional iterations to freeze (as the freezing rate is unaffected by the increased number of ants simulated in each iteration). The total number of iterations required to rediscover and freeze all shortest paths is, per linearity of expectation, at most $O(\ell + \ell \ln(\tau_{\max}/\tau_{\min})/\rho)$. $\qquad\square$

While increasing the number of ants simulated does not reduce the amount of work performed by the algorithm, it does allow the work to be parallelized to a greater extent, as each ant in an iteration can be simulated independently of others, only needing to compare the fitness values of the constructed paths at the end of the iteration.

The theorems presented in this section used ℓ, the maximum number of arcs in any shortest path to t, to bound the complexity of rediscovering the shortest paths after a weight function change. In the examples considered so far, this has been accurate; though there are graphs where it is an overestimation (see e.g. Figure 2). More precise bounds could instead consider, for each shortest path, the number of vertices along the path at which an ant would have opportunity to deviate from the shortest path, and for which the shortest path was altered by the weight function change.
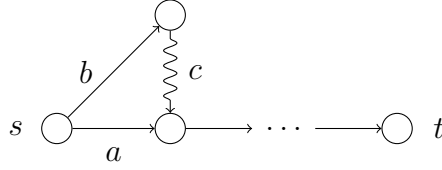
Figure 2: Changing the weight of arc c controls which of the a and b arcs is on the shortest path from s to t.

4. Periodic local changes

If the weight functions change more frequently than the expected number of iterations required to rediscover and re-freeze the shortest paths after each change, the process can no longer be treated as a series of recoveries from one-time changes. Taken to the extreme, the weight functions could change in *every* iteration of λ-MMAS. In this section, we consider such rapid oscillation between two weight functions the shortest paths of which are extremely similar, differing only in the choice of outgoing edge from a single vertex in the graph. Somewhat surprisingly, λ-MMAS, with only a constant number of ants, maintains at least a constant probability of constructing the optimum shortest path solution in each iteration in this setting. We'll then show that if the oscillation between these two weight functions is slower (though not as slow as to allow analysis as a series of one-time changes), 1-MMAS is also able to track the optimum.

Throughout this section, λ-MMAS is considered able to track the optimum solution if it is able to maintain a constant probability of constructing the shortest paths in any given iteration, i.e. the best possible probability of success when a starting a constant number of ants at each vertex. In the local changes setting, only a single choice of outgoing edge changes between the shortest paths of the two different weight functions. λ-MMAS must therefore keep the pheromone values on the oscillating edges within a constant distance of $1/2$, as approaching the pheromone bounds would reduce the probability that the correct shortest paths is constructed in every iteration to a sub-constant value. We note that adjusting the pheromone bounds directly, to keep the pheromones close to $1/2$, would be counterproductive: such bounds would reduce the probability of ants following the non-oscillating portion of the shortest paths, which would've been reinforced, to a sub-constant value when alternative paths are available. Additionally, a low evaporation rate is required in order for λ-MMAS to be able to keep the pheromone values on the oscillating edges close to $1/2$ for significant amounts of time, as, intuitively, a too-high evaporation rate causes λ-MMAS to forget that an arc was part of a shortest path sooner rather than later.

This section considers oscillation between two similar weight functions, shown below, on the graph in Figure 2.

$$w_1(\alpha) = \begin{cases} 1 & \text{if } \alpha = c \\ 0 & \text{otherwise} \end{cases} \quad w_2(\alpha) = \begin{cases} -1 & \text{if } \alpha = c \\ 0 & \text{otherwise} \end{cases}$$

The graph and weight functions are chosen such that the longest shortest path has at least $n - 2$ arcs, the shortest paths are only changed at a single vertex, which isolates the

10

simplest possible dynamic shortest path problem, and the simulated ants only have the opportunity to deviate from the shortest path at a single vertex. The latter limit on the opportunities to deviate from the shortest paths is used only to simplify the analysis, and could be relaxed at the cost of increasing the required number of ants by a constant factor, as discussed at the end of this section; thus, similar results could be obtained for less artificial graphs, as long as the oscillation could be isolated to a single triangle in the graph.

In this situation, it is possible for λ-MMAS to use relatively few ants and still maintain the ability to correctly identify the shortest path in each iteration with at least constant probability for a super-polynomial number of iterations.

Lemma 6. *Consider switching between the weight functions w_1 and w_2 on the graph of Figure 2 during every iteration. Starting $\lambda = 4$ ants at each vertex is sufficient to ensure that λ-MMAS, with $\rho \in o(1/\log n)$, will keep the pheromone values on outgoing arcs from s in the $[0.3, 0.7]$ range for a super-polynomial number of iterations with respect to n with a probability super-polynomially close to 1.*

Proof. Let $\tau'_t = \min(\tau_t(a), \tau_t(b))$ be the minimum of the pheromone values on the arcs a and b after iteration t. Consider the effect of two sequential pheromone updates on τ'_t:

$$-\rho < \tau'_{t+2} - \tau'_t < 2\rho \tag{1}$$

as evaporation may at most reduce the smaller value by $\rho/2$ (as $\tau_t(a) + \tau_t(b) = 1$, and hence $\tau'_t \leq 1/2$), and reinforcement may at most increase the smaller value by ρ.

If the arc with the smaller pheromone value τ'_t is only reinforced in the second of two pheromone updates, the effect is:

$$\tau'_{t+2} - \tau'_t \geq \tau'_t(1-\rho)^2 + \rho - \tau'_t \geq \rho - 2\rho\tau'_t \tag{2}$$

It is convenient to scale the minimum pheromone values, such that a constant number of pheromone updates alter the scaled values by at most a constant:

$$g(\tau_t) = \min(\tau_t(a), \tau_t(b))/\rho$$

Consider the expected change $\Delta_t(\tau'_t) = g(\tau_{t+2}) - g(\tau_t)$ of these scaled values two pheromone updates after an update that favored reinforcing the arc with the τ'_t pheromone value. For a lower bound, assume that the τ'_t arc is never reinforced during the next iteration (when it is not on the shortest path), and let p_f be the probability that it is also not reinforced during the iteration after the next (when it is on the shortest path), which yields:

$$E(\Delta_t(\tau'_t)) > (1 - p_f) \cdot (1 - 2\tau'_t) + p_f \cdot (-1)$$
$$= 1 - 2p_f(1 - \tau'_t) - 2\tau'_t$$

by applying g to the bounds in (1) and (2). For the weight functions w_1 and w_2, $p_f \leq (1 - \tau'_t(1-\rho))^\lambda$. The drift is at least 0.02 when $0.3 \leq \tau'_t \leq 0.4$, $\lambda = 4$, and $\rho \leq 0.05$.

The simplified drift theorem (Oliveto and Witt, 2011, 2012) can then be applied to the scaled values when the pheromones are within this region, treating two iterations of

11

λ-MMAS as a single step in a Markov process over the scaled values, ensuring that the arc with the minimum pheromone value was always favored by the last iteration in the step. This requirement simplifies the following analysis, but may mean that the pheromone values enter the considered drift region during a "wrong" iteration; to accommodate this, we will adjust the size of the drift region by the equivalent of one iteration, pessimistically assuming that the iteration resulted in the pheromone values being brought closer to the bounds.

The drift is greater than 0.02 within a region specified by $a = (3/10)/\rho$ and $b = (4/10)/\rho$. As mentioned perviously, the length of the drift region may need to be reduced by the equivalent of a single pheromone update to ensure that the last pheromone update favored reinforcing the τ'_t arc, so the length of the drift region is $\ell \geq b - a - 1 = \omega(\log n)$. This region then satisfies the first requirement of the simplified drift theorem.

As a consequence of (1), the process cannot make large changes to the scaled values in a single iteration. The second requirement of the simplified drift theorem is satisfied by setting $r(\ell) = 4$ and $\delta = 1$:

$$P(|\Delta_t| \geq j) \leq \frac{r(\ell)}{(1+\delta)^j} = 2^{2-j}$$

as $P(|\Delta_t| > 2) = 0$, and the right side of the inequality is at least 1 for $j \leq 2$.

Then, per the simplified drift theorem, there exists a constant c^* such that with probability $1 - 2^{-\omega(\log n)}$, the Markov process first reaches a state $X_t \leq a$ (and hence first encounters a $\tau'_t \leq 3/10$) after at least $2^{c^* \omega(\log n)}$ iterations. $\qquad \square$

Given that the pheromone values remain within this range for a super-polynomial number of iterations, the probability that the correct shortest path x^*_s is constructed also remains bounded for a super-polynomial number of iterations.

A more detailed analysis than the approximations used in Lemma 6 is needed to show whether $\lambda = 3$ ants are also sufficient. This setting is revisited as part of an experiment in Section 6.

Theorem 7. *Consider switching between the weight functions w_1 and w_2 on the graph of Figure 2 during every iteration. 4-MMAS with evaporation rate $\rho \in o(1/\log n)$ will with probability super-polynomially close to 1 be able to find the correct s-t shortest path with at least a constant probability in each iteration for any polynomial number of iterations.*

Proof. Let $\tau'_t = \min(\tau_t(a), \tau_t(b))$ after iteration t; then, the probability that x^*_s is correct is at least the probability of any ant constructing a path through the arc with pheromone value τ'. As long as τ'_t is at least a constant, e.g. 0.3, this probability is also at least a constant:

$$P(x^*_s \text{ is correct} \mid \tau'_t \geq 0.3) \geq 1 - (1 - 0.3)^4 = 0.7599$$

Per Lemma 6, $\tau'_t \geq 0.3$ for a super-polynomial number of iterations with probability super-polynomially close to 1, which proves the theorem. $\qquad \square$

The proof relies on a special property of the considered weight functions: only the choice made at s affects whether the ant constructs a correct shortest path to t. If the weight function or graph were changed to require the ants to follow a pheromone-reinforced path after leaving s, $\lambda = 4$ ants may be insufficient to ensure a positive drift. Using the usual pheromone bounds, the probability of an ant successfully following any pheromone-reinforced path is at least $1/e$, so the failure probability can be adjusted accordingly:

$$p_{\mathrm{f}} \leq (1 - \tau_t'(1 - \rho) \cdot 1/e)^{\lambda}$$

With this adjustment, $\lambda = 12$ ants are sufficient to ensure that the drift is greater than 0.003 for the same constraints on τ_t' and ρ as were used in Lemma 6.

4.1. Tracking slower local changes

The preceding section has shown that an ant colony can be useful when tracking rapid oscillations, as increasing the number of ants increases the amount of exploration the λ-MMAS algorithm performs before altering pheromone values. If the evaporation rate is also sufficiently low, during rapid oscillation between two similar shortest paths, λ-MMAS is able to keep the pheromone values on the affected arcs close to 0.5, essentially being able to remember that both outgoing arcs have been part of the shortest path recently.

A colony with only a single ant cannot track *rapid* oscillation in this fashion, but as the following theorem shows, it can keep the pheromone values within a constant range if the oscillation is sufficiently slow (i.e. each shortest path remains the optimum for at least $T \in \omega(\log n)$ iterations). With only a single ant, the pheromones are instead kept within a constant range by the difference in magnitude of the pheromone updates which change the values toward the middle, and those toward the extremes.

Theorem 8. *Consider switching between the weight functions w_1 and w_2 on the graph of Figure 2 every $T = 2/\rho$ iterations. 1-MMAS with an evaporation rate $\rho \in o(1/\log n)$ will keep the pheromone values within the $[0.01, 0.99]$ range for an expected super-polynomial number of iterations.*

Proof. Consider the optimization process as a series of phases, with alternating phases using w_1 and w_2 as the weight functions. Then, at the beginning of an optimization phase, at iteration t, let $\tau_t(a)$ be the pheromone value on the arc favored by (i.e. on the shortest path of) the weight function used in the phase that has just began. We shall show that if $\tau_t(a) \in [0.03, 0.5]$, then with super-polynomially high probability, the pheromone value on the non-favored arc will be in this range at the end of this optimization phase, i.e. $\tau_{t+T}(b) \in [0.03, 0.5]$.

If the new shortest path is discovered immediately, the pheromone value $\tau_t(b)$ will not be reinforced in the following T iterations. Recall that $\tau_t(a) + \tau_t(b) = 1$, so $\tau_t(b) \geq 0.5$, which after T evaporating pheromone updates will be reduced to no less than 0.03, given that $\rho \leq 0.1$, i.e. n is large enough:

$$\tau_{t+T}(b) \geq \tau_t(b)(1 - \rho)^T$$
$$\geq 0.5(1 - \rho)^{2/\rho} \geq 0.03$$

13

So regardless of when the new shortest path is discovered, the pheromone value $\tau_{t+T}(b)$ cannot be reduced below the range acceptable for the next optimization phase.

Suppose that the new shortest path is not discovered immediately, so the pheromone value on arc a is evaporated until the shortest path is discovered. In the first $T/2$ iterations, the pheromone value on arc a cannot be reduced below 0.01, given that $\rho \leq 0.1$ as before:

$$\tau_{t+T/2}(a) \geq \tau_t(a)(1-\rho)^{T/2}$$
$$\geq 0.03(1-\rho)^{1/\rho} \geq 0.01$$

Thus the shortest path is discovered in the first $T/2$ iterations with super-polynomially high probability:

$$1-(1-0.01)^{T/2} \geq 1-0.99^{1/\rho} \geq 1-0.99^{\omega(\log n)} \geq 1-n^{-\omega(1)}$$

If the shortest path is discovered in the first $T/2$ iterations, the pheromone value on arc b will be evaporated for at least $T/2$ iterations:

$$\tau_{t+T}(b) \leq \tau_{t+T/2}(b)(1-\rho)^{T/2}$$
$$\leq 0.99(1-\rho)^{1/\rho} \leq 0.99/e < 0.5$$

and will therefore be within the acceptable range for the next optimization phase.

Therefore, if at the beginning of the phase, the pheromone value on arc favored by the phase is within $[0.03, 0.5]$, the pheromone value on the arc favored by the next phase will also be within this interval at the start of the next phase with super-polynomially high probability.

The first phase of the optimization process satisfies this precondition (as pheromone values are initialized s.t. $\tau_1(a) = \tau_1(b) = 0.5$); and as each phase has a super-polynomially small probability of failing, a failure occurs (potentially causing pheromone values to exit the $[0.01, 0.99]$ range) in an expected super-polynomial number of iterations. $\qquad\square$

This illustrates that if the oscillation is sufficiently slow, and the evaporation rate is sufficiently low, the pheromone update mechanism in MMAS can still prevent the pheromones from freezing for an expected super-polynomial number of iterations, even in a single-ant colony. More specifically, the magnitude of the change in the pheromone values of an update bringing pheromone values closer to $1/2$ is greater than that in an update bringing pheromone values closer to their bounds.

5. Periodic global changes

Using pheromones as an implicit memory, λ-MMAS is able to reliably construct shortest paths even with a constant number of ants when the oscillating weight functions only make slight alterations to the shortest paths in the graph. In this section, we consider an oscillation between two weight functions where the shortest paths at many vertices change, and there are many opportunities for an ant to deviate from the shortest path.
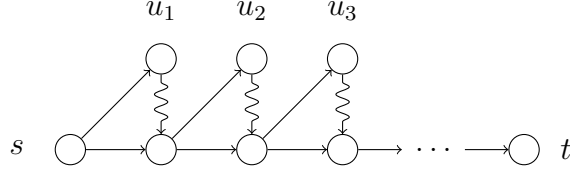
Figure 3: Multiple oscillating triangles on the path from s to t.

Suppose that there are k oscillating triangles on the path from s to t, as illustrated in Figure 3. The weight functions w_1 and w_2 can be extended such that the shortest path using w_1 avoids all of the vertices $u_1, \ldots, u_k$, while w_2 favors visiting all of those vertices. Let U be the set of arcs leaving the u_i vertices; the extended weight functions are then:

$$w_1(\alpha) = \begin{cases} 1 & \text{if } \alpha \in U \\ 0 & \text{otherwise} \end{cases} \qquad w_2(\alpha) = \begin{cases} -1 & \text{if } \alpha \in U \\ 0 & \text{otherwise} \end{cases}$$

When k is large, i.e. $k \in \omega(\log n)$, ensuring that the correct shortest path from s to t is constructed with at least constant probability in each iteration requires starting a super-polynomial number of ants at each vertex. While tracking the optimum during an oscillation between two dramatically different weight functions is difficult, this setting illustrates a limitation of the pheromone memory used by MMAS: it is unable to store distinct solutions (unlike e.g. a $(\mu + \lambda)$ EA, which might be able to store both shortest path trees in the population, and clone both in any given iteration if μ and λ are large enough).

Theorem 9. *Rapid oscillation between two weight functions may require as many as $2^{\Omega(n)}$ ants to be started at each vertex in order for λ-MMAS with $\rho \leq 0.5 - \Omega(1)$ to maintain at least a constant probability of constructing the shortest paths in each iteration.*

Proof. The theorem is proved by example: we'll show that on the graph of Figure 3 with $k = \Omega(n)$ triangles, with weight functions swapping between w_1 and w_2 during every iteration, λ-MMAS requires $\lambda = 2^{\Omega(k)} = 2^{\Omega(n)}$ ants to be started at each vertex to maintain a constant probability of constructing the shortest path from s in each iteration. Essentially, λ-MMAS is no better in this setting than exhaustive search during every iteration; this is somewhat expected as the shortest paths for w_1 and w_2 are substantially different, and λ-MMAS is not able to store distinct solutions in the pheromone memory.

Consider the pheromone state in an iteration of λ-MMAS. Classify triangle i as *favorable* with respect to the current weight function if the pheromone value on the correct shortest path arc from s_i is at least 0.5. If less than $k/2$ triangles are favorable, constructing the shortest path from s requires a number of ants that is exponential with respect to k, as at least some ant has to make the correct choice of outgoing arc in all of at least $k/2$ unfavorable triangles. Let p_s be the probability that one of the λ ants started at s constructs the correct shortest path, p_k be the probability that a particular ant correctly navigates the $k/2$ unfavorable triangles, and $p_1 \leq 0.5$ be the probability that a particular ant correctly

15

navigates a specific unfavorable triangle, then

$$p_s \le 1 - (1 - p_k)^\lambda \le 1 - (1 - (0.5)^{k/2})^\lambda \le \lambda \cdot 2^{-k/2}$$

To ensure that p_s is at least a constant, at least $2^{\Omega(k)}$ ants must be started from s.

If more than half the triangles are favorable in the current iteration, an exponential number of ants will be required to construct the shortest paths from s in the next iteration. Let $\tau \ge 0.5$ be the pheromone value on the correct shortest path arc in a triangle in the current iteration; then $1 - \tau$ is the pheromone value on the other arc in the same triangle. In the next iteration, that other arc will be on the shortest path, and its pheromone value will be at most $0.5 + \rho$. As long as ρ is upper-bounded by a constant less than 0.5, the derivation of p_s can be repeated using $p_1 \le 0.5 + \rho$ to show that an exponential number of ants must be started from s in the next iteration.

Thus, at least every second iteration requires λ to be exponential with respect to the number of triangles in order for λ-MMAS to have a constant probability of at least one ant constructing the shortest path from s to t. $\square$

A closer analysis may be able to show that in expectation, no more than some constant fraction of the triangles will favor a specific weight function. This would mean that an exponential number of ants is required in every iteration (rather than at least every second iteration as concluded in the proof of Theorem 9) to ensure a constant probability of the correct s-t shortest path being constructed in an arbitrary iteration.

In general, the smaller the evaporation rate, the less effect the first iteration has on the pheromone values, reducing the amount of bias towards w_1 introduced at the start of the optimization process. A smaller λ has a similar effect: ants started further away from t may not be able to find the full shortest path, which reduces the probability that the w_1-favored arc is reinforced. Subsequent iterations will, due to drift, reduce the bias introduced by the asymmetry at the start of the process; i.e. if more triangles favor w_1 than w_2, in expectation more w_1-favoring triangles will be reinforced towards w_2 during a w_2 iteration than w_2-favoring triangles towards w_1 during a w_1 iteration.

In any case, the example illustrates the nature of the difficulty: the pheromone memory isn't capable of separating the two weight functions, and thus does not scale well if the differences between the weight functions extend over multiple vertices.

6. Experiments

To illustrate the behavior of λ-MMAS in settings introduced in previous sections, a number of simulations were performed; all averages and medians presented in the figures in this section are computed based on data from 1000 simulations for each combination of MMAS parameters ρ and λ, and, where applicable, oscillation frequency.

In the local changes setting with a single triangle, λ-MMAS with $2 \le \lambda \le 4$ ants started at each vertex was simulated with different evaporation rates, recording the first iteration at which the pheromone values on the triangle exited the $[1/10, 9/10]$ range (i.e. were about to freeze) for the first time; the averages are shown in Figure 4. While it seems the $\lambda = 2$
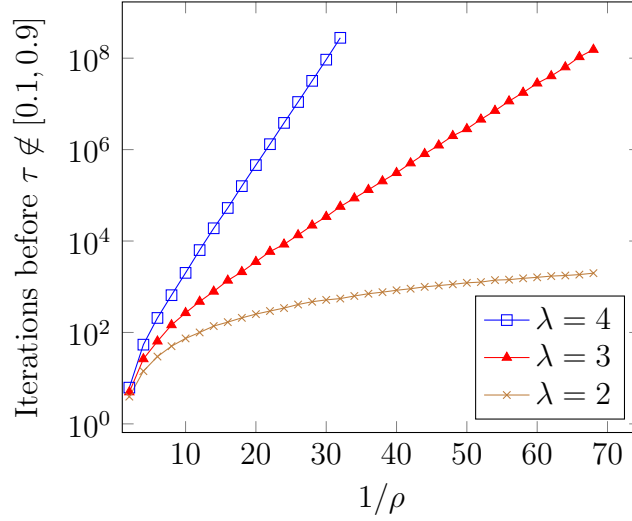
Figure 4: Number of iterations before pheromone values on the triangle leave $[0.1, 0.9]$ for the first time for varying evaporation rates for λ-MMAS, with $\lambda = 2, 3, 4$ ants started at each vertex during a single iteration.

curve is concave, which suggests that $\lambda = 2$ ants are not sufficient to prevent pheromones from freezing for a super-polynomial number of iterations, it seems that $\lambda = 3$ ants may be sufficient to achieve this.

For 1-MMAS in the local changes setting, the single-ant colony was simulated with different evaporation rates and phase lengths T, recording the first iteration at which pheromone values on the triangle arcs exited specific ranges; the results are shown in Figure 5. There appears to be a more significant difference between $T = 4$ and $T = 3$ than $T = 3$ and $T = 2$. Additionally, it seems that the pheromone values also remain within more restrictive ranges than $[0.01, 0.99]$ for significant numbers of iterations.

In the global changes setting with $k = 200$ triangles in series, λ-MMAS was simulated with $\lambda = 6$ and $\rho = 1/50$. The number of triangles with pheromone values outside the $[1/4, 3/4]$ range was recorded; Figure 6 displays the average number of triangles favoring with pheromone values outside this range, collated by which weight function the pheromone values were favoring. Notably, the number of triangles favoring either weight function increases at approximately equal rates, and eventually stabilizes, keeping approximately a third of the triangles in the graph within the $[1/4, 3/4]$ pheromone range.

Figure 7 presents the same metric in the global changes setting with the evaporation rate set to $\rho = 1$ to maximize the impact of the first iteration. Somewhat surprisingly, even with this extreme evaporation rate, the ratio between the number triangles frozen to favor w_1 and w_2 is initially only 3:2. As suggested in Section 5, the ratio decreases gradually over time, illustrating that the impact of the initial bias is counteracted, rather than compounded, by subsequent iterations.

The number of triangles favoring w_1 in Figure 6 is consistently greater than the number of triangles favoring w_2; this is an artifact of always recording the number of triangles after
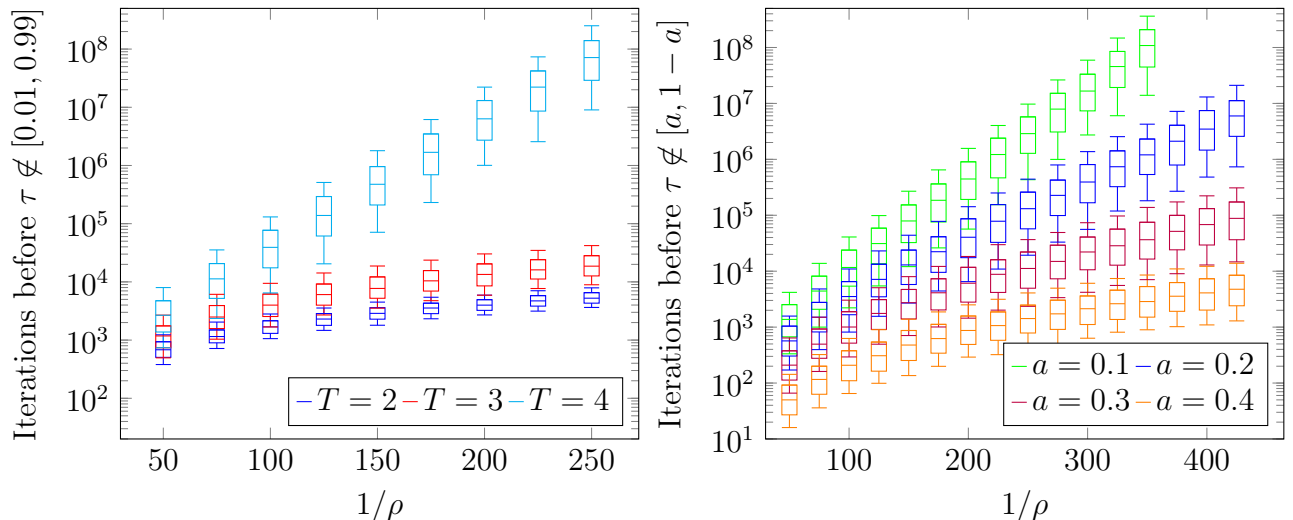
17

Figure 5: Local changes with slower oscillation periods; showing the median, quartiles, and 9th and 91st percentiles of the number of observations required to leave specific pheromone ranges. The left subfigure illustrates the effect of of phase length, while the right subfigure illustrates, for $T = 4$, the number of iterations during which 1-MMAS pheromone values remain within $[a, 1-a]$ for various constants a.

an iteration using the w_1 weight function. In 7, the data points alternate between iterations using w_1 and w_2, causing the lines to appear jagged.

7. Conclusions

We have studied a simple ACO algorithm called λ-MMAS for dynamic variants of the single-destination shortest paths problem. Building upon previous results for the special case of 1-MMAS, it was studied to what extent an enlarged colony using λ ants per vertex helps in tracking an oscillating optimum. We showed that λ-MMAS, even with constant number of ants per vertex, can deal with dynamic shortest paths problems where the shortest paths are changed infrequently (by rediscovering the shortest paths before the next change occurs), or changed rapidly between a small set of possible similar solutions (by keeping the pheromone values close to $1/2$ for the affected arcs). It has also been shown that even a single-ant colony can prevent the pheromone values from freezing for a sufficiently slow oscillation. However, we also identified an example where a fast oscillation between two weight functions that are sufficiently different is so hard to track that a super-polynomial number of ants is needed. Furthermore, we have discussed properties of the dynamics that make the problem hard. Experiments show that the theoretical results are also valid for small problem dimensions and illustrate effects that are not yet visible in theorems.

8. Future Work

In the future, the performance of λ-MMAS in settings with more complex weight function changes could be analyzed – for instance, generalizing the results to oscillations with constant
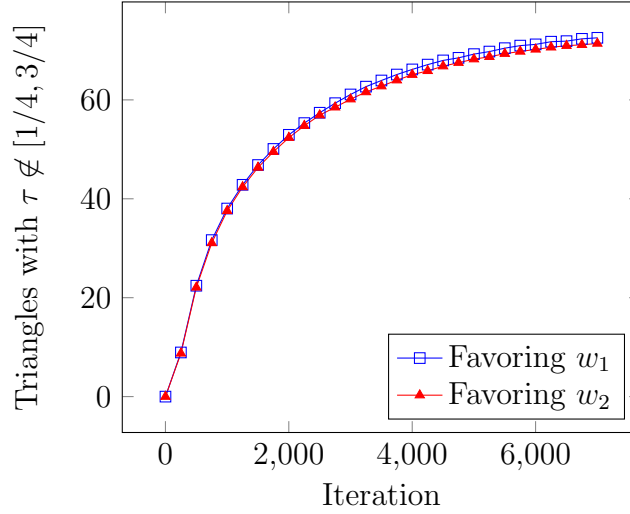
18

Figure 6: 200 triangles in series; average across 1000 simulations; $\lambda = 6, \rho = 1/50$.

phase lengths, oscillations between more than two weight functions, or using a less regular schedule than was considered here. Having different components of the graph oscillate at different frequencies could also pose an interesting challenge to ACO algorithms, one that could potentially be addressed by maintaining several sets of pheromones updated using different evaporation rates.

References

Attiratanasunthron, N., Fakcharoenphol, J., 2008. A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs. Information Processing Letters 105, 88–92.

Auger, A., Doerr, B. (Eds.), 2011. Theory of Randomized Seach Heuristics–Foundations and Recent Developments. World Scientific.

Doerr, B., Hota, A., Kötzing, T., 2012. Ants easily solve stochastic shortest path problems, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2012), ACM Press. pp. 17–24.

Droste, S., 2003. Analysis of the (1+1) EA for a dynamically bitwise changing OneMax, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2003), Springer. pp. 909–921.

Feldmann, M., Kötzing, T., 2013. Optimizing expected path lengths with ant colony optimization using fitness proportional update, in: Proc. of Foundations of Genetic Algorithms (FOGA 2013), ACM Press. pp. 65–74.

Garlick, R.M., Barr, R.S., 2002. Dynamic wavelength routing in wdm networks via ant colony optimization, in: Dorigo, M., Caro, G.D., Sampels, M. (Eds.), Ant Algorithms, Springer. pp. 250–255.

Jansen, T., 2013. Analyzing Evolutionary Algorithms: The Computer Science Perspective. Natural Computing Series, Springer.

Jansen, T., Jong, K.A.D., Wegener, I., 2005. On the choice of the offspring population size in evolutionary algorithms. Evolutionary Computation 13, 413–440.

Jansen, T., Schellbach, U., 2005. Theoretical analysis of a mutation-based evolutionary algorithm for a tracking problem in the lattice, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2005), ACM Press. pp. 841–848.

Kötzing, T., Lehre, P., Neumann, F., Oliveto, P., 2010. Ant colony optimization and the minimum cut problem, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2010), ACM Press. pp. 1393–1400.
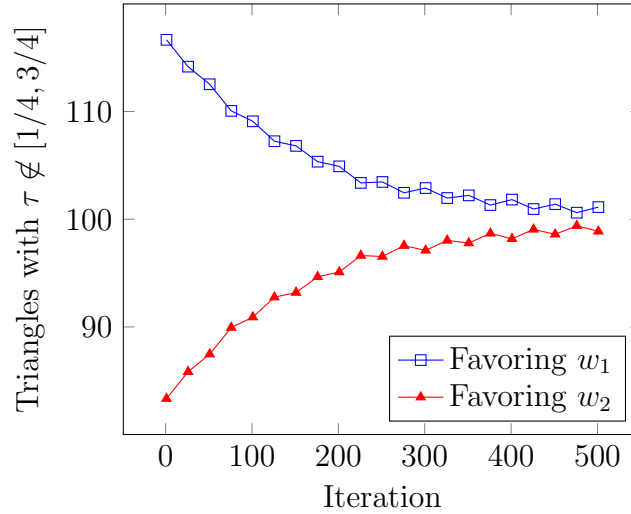
Figure 7: 200 triangles in series; average across 1000 simulations; $\lambda = 6, \rho = 1$.

Kötzing, T., Molter, H., 2012. ACO beats EA on a dynamic pseudo-boolean function, in: Proc. of Parallel Problem Solving from Nature (PPSN 2012), Springer. pp. 113–122.

Kötzing, T., Neumann, F., Röglin, H., Witt, C., 2012. Theoretical analysis of two ACO approaches for the traveling salesman problem. Swarm Intelligence 6, 1–21.

Neumann, F., Sudholt, D., Witt, C., 2010. A few ants are enough: ACO with iteration-best update, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2010), pp. 63–70.

Neumann, F., Witt, C., 2010a. Ant colony optimization and the minimum spanning tree problem. Theoretical Computer Science 411, 2406–2413.

Neumann, F., Witt, C., 2010b. Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity. Springer.

Nguyen, T.T., Yang, S., Branke, J., 2012. Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation 6, 1–24.

Oliveto, P.S., Witt, C., 2011. Simplified drift analysis for proving lower bounds in evolutionary computation. Algorithmica 59, 369–386.

Oliveto, P.S., Witt, C., 2012. Erratum: Simplified drift analysis for proving lower bounds in evolutionary computation. arXiv:1211.7184 Http://arxiv.org/abs/1211.7184.

Rohlfshagen, P., Lehre, P.K., Yao, X., 2009. Dynamic evolutionary optimisation: An analysis of frequency and magnitude of change, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2009), ACM Press. pp. 1713–1720.

Stützle, T., Hoos, H.H., 2000. MAX-MIN ant system. Future Generation Computer Systems 16, 889–914.

Sudholt, D., Thyssen, C., 2012a. Running time analysis of ant colony optimization for shortest path problems. Journal of Discrete Algorithms 10, 165–180.

Sudholt, D., Thyssen, C., 2012b. A simple ant colony optimizer for stochastic shortest path problems. Algorithmica 64, 643–672.

Xiang, W., Lee, H., 2008. Ant colony intelligence in multi-agent dynamic manufacturing scheduling. Engineering Applications of Artificial Intelligence 21, 73 – 85. URL: http://www.sciencedirect.com/science/article/pii/S0952197607000437, doi:http://dx.doi.org/10.1016/j.engappai.2007.03.008.

Zhou, Y., 2009. Runtime analysis of an ant colony optimization algorithm for TSP instances. IEEE Transactions on Evolutionary Computation 13, 1083–1092.