

Ant Colony Optimization and the Minimum Cut Problem

Timo Kötzing

Department 1: Algorithms and Complexity
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
koetzing@mpi-inf.mpg.de

Frank Neumann

Department 1: Algorithms and Complexity
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
fne@mpi-inf.mpg.de

Per Kristian Lehre

School of Computer Science
University of Birmingham
B15 2TT Birmingham, United Kingdom
p.k.lehre@cs.bham.ac.uk

Pietro S. Oliveto

School of Computer Science
University of Birmingham
B15 2TT Birmingham, United Kingdom
p.s.oliveto@cs.bham.ac.uk

ABSTRACT

Ant Colony Optimization (ACO) is a powerful metaheuristic for solving combinatorial optimization problems. With this paper we contribute to the theoretical understanding of this kind of algorithm by investigating the classical minimum cut problem. An ACO algorithm similar to the one that was proved successful for the minimum spanning tree problem is studied. Using rigorous runtime analyses we show how the ACO algorithm behaves similarly to Karger and Stein's algorithm for the minimum cut problem as long as the use of pheromone values is limited. Hence optimal solutions are obtained in expected polynomial time. On the other hand, we show that high use of pheromones has a negative effect, and the ACO algorithm may get trapped in local optima resulting in an exponential runtime to obtain an optimal solution. This result indicates that ACO algorithms may be inappropriate for finding minimum cuts.

Categories and Subject Descriptors

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

General Terms

Theory, Algorithms, Performance

Keywords

Ant Colony Optimization, Min-cut

1. INTRODUCTION

Ant colony optimization (ACO) [4] is a powerful class of general purpose algorithms that has been applied to a wide range of combinatorial optimization problems. While ACO

algorithms are successfully used in applications, the theoretical foundation of this kind of algorithm is not well developed. Increasing the theoretical foundation of ACO algorithms has been pointed out as one of the most important challenges in the research field of ant colony optimization [3]. The goal of this paper is to contribute to the theoretical understanding of ACO algorithms by carrying out rigorous runtime analyses on a classical combinatorial optimization problem, namely the minimum cut problem.

The analysis of ACO algorithms with respect to their runtime behavior started for some well-known pseudo-Boolean functions [8, 9, 16, 2, 14]. Such analyses treat ACO algorithms as randomized algorithms in a classical sense and analyze the number of iterations to obtain optimal or good solutions for a given problem. As in the case of evolutionary algorithms, such studies on pseudo-Boolean functions should point out basic properties of these algorithms and set the basis for analyses on more realistic problems. This path of research has already been followed for evolutionary algorithms in a very successful way. Based on results for different kinds of pseudo-Boolean functions [5, 11, 20], results have been obtained for different kinds of combinatorial optimization problems. An overview of the different results that have been obtained on the runtime of evolutionary algorithms for combinatorial optimization problems can be found in [17].

There are only few results on the rigorous runtime analysis of ACO algorithms for classical combinatorial optimization problems. To our knowledge, the only problems where rigorous results have been obtained are the computation of shortest paths [1, 10] and minimum spanning trees [15]. Recently, also an initial study for the traveling salesman problem has been carried out [21]; however, this paper only analyses some simple instances. All studies on classical combinatorial optimization problems are motivated by the wish to understand the basic working principles on natural examples. Such studies should be the basis for a deeper understanding of ACO algorithms that can be used later on for the development of even more successful algorithms.

In this paper, we examine the behavior of ACO on another well-known combinatorial optimization problem in a rigorous way. We study the well-known minimum cut problem in graphs. This problem can be solved by different kinds of algorithms in polynomial time. Several related problems,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '10, July 7–11, 2010, Portland, Oregon, USA.

Copyright 2010 ACM 978-1-4503-0072-8/10/07 ...\$10.00.

such as the minimum k -cut problem and the multiway cut problem, are NP-hard [19]. It has been shown in [18, 6] that general stochastic search algorithms, such as evolutionary algorithms and ant colony optimization, can achieve good results for difficult cutting problems. However, this is not necessary the case for the classical minimum cut problem. In a recent theoretical work, [13] proves that simple single-objective evolutionary algorithms cannot solve the minimum cut problem in expected polynomial time.

ACO algorithms construct solutions for a problem in a different way compared to evolutionary algorithms. In this paper we analyse whether this may be an advantage for ACO on the minimum cut problem. We use a similar approach to the one used for the minimum spanning tree problem examined in [15]: To obtain a candidate solution for the minimum-cut problem, we compute a forest consisting of $n - 2$ edges, which yields a graph consisting of exactly two connected components. The edges that lie between these two components constitute the cut that is given by such a solution.

We examine this approach with respect to a wide range of parameter settings that determine the importance of pheromone and heuristic information. It turns out that this ACO approach behaves similar to the well-known randomized algorithm of Karger and Stein [12] for computing a minimum cut, as long as the influence of pheromone values is not too big. As a consequence we get ACO algorithms that provably find optimal cuts in an expected polynomial number of iterations. On the other hand, we show that our ACO approach fails once a higher influence of pheromone values is allowed: There are graphs where provably an expected exponential number of iterations is needed to obtain an optimal solution.

In Section 2 we describe the ACO approach examined in this paper. Afterwards, we point out for which parameter settings our approach leads provably to an expected polynomial optimization time in Section 3. Finally, in Section 4, we discuss parameter settings and worst case instances where the stated approach fails.

2. THE ALGORITHM

The min-cut problem can be stated as follows. Given an undirected graph $G = (V, E)$ with $|V| = n$ vertices, m edges and a weight function $w : E \rightarrow \mathbb{R}_+$, the goal is to find a partitioning of the vertices into two sets $V_1 \neq \emptyset$ and $V_2 \neq \emptyset$ such that the total weight of the edges between them is minimal.

We study a simple ACO algorithm called MMAS* (see Algorithm 2), already analyzed in [15] for the minimum spanning tree (MST) problem. MMAS* stands for *Max-Min Ant System*, where the max-min refers to having upper and lower bounds on the *pheromone values* used (see below for a discussion on pheromone values). MMAS* works iteratively, creating one new candidate solution x in each iteration, and keeping track of the best-so-far solution x^* .

New candidate solutions are generated by a procedure **Construct** (see Algorithm 1). This procedure starts with an empty graph and iteratively introduces more and more edges between different components while not creating cycles. Conceptually, choosing an edge can be seen as contracting this edge. Finally, a graph with exactly two connected components is obtained. These two connected components are given as the output by the **Construct** procedure, and

the edges between them constitute the cut of this candidate solution.

A run of **Construct** on a graph $G = (V, E)$ returns a set E_0 of $n - 2$ edges of G . By the definition of the construction procedure **Construct**, the graph $G = (V, E_0)$ can be partitioned into two vertex sets $V_1 \neq \emptyset$ and $V_2 \neq \emptyset$ such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ and $(V_1 \times V_2) \cap E_0 = \emptyset$, i. e. there is no edge in E_0 having their endpoints in different partitions.

The procedure **Construct** is randomized and works as follows. We imagine an *artificial ant* to choose step by step components of a candidate solution. In our setting, the components to choose from are the edges from the edge set $\{e_1, \dots, e_m\}$ of the input graph G . The *construction graph* $C(G)$ is a directed graph on the $m + 1$ nodes $\{s, e_1, \dots, e_m\}$ with the designated start node s . Its edge set A of cardinality m^2 is given by

$$A := \{(e_i, e_j) \mid 0 \leq i \leq m, 1 \leq j \leq m, i \neq j\},$$

i. e., $C(G)$ is obtained from the complete directed graph by removing all self-loops and the edges pointing to s . We give a rough graphical depiction of $C(G)$ in Figure 1.

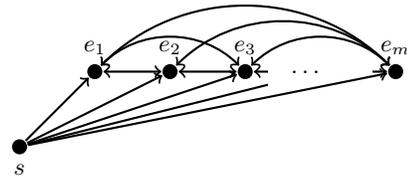


Figure 1: The Construction Graph $C(G)$.

To choose components for a candidate solution, we imagine the ant to traverse $C(G)$. When the artificial ant visits node e in $C(G)$, this corresponds to choosing the edge e for the partitioning. To ensure that a walk of the ant actually constructs a forest, we define the feasible neighborhood $N(e'_k)$ of construction node e'_k depending on the nodes e'_1, \dots, e'_k visited so far:

$$N(e'_k) := (E \setminus \{e'_1, \dots, e'_k\}) \setminus \{e' \in E \mid (V, \{e'_1, \dots, e'_k, e'\}) \text{ contains a cycle}\}.$$

We want the ant to choose an outgoing edge with a probability based on the *pheromone value* τ left on the edge by previous ants and on *heuristic information* η on the edge (the pheromone values change during the computation of MMAS*, the heuristic information does not). The complete algorithm to choose a new candidate solution can now be given as a procedure **Construct** as shown in Algorithm 1.

We want to use heuristic information that prefers edges in the construction procedure that have large weights. Hence, we set $\eta(e_i, e_j) = w(e_j)$.

The algorithm MMAS* consists of iteratively generating new solutions using the procedure **Construct**, keeping the best-so-far solution x^* , and updating the pheromone values in between each two iterations.

Initially, all pheromone values are the same such that the pheromone values sum up to 1 (i. e., each edge $(u, v) \in A$

Algorithm 1: The algorithm $\text{Construct}(C(G), \tau, \eta)$

```
1 function Construct on  $C(G), \tau, \eta$  is
2    $e'_0 \leftarrow s$ ;
3    $k \leftarrow 0$ ;
4   for  $k = 0$  to  $n - 3$  do
5      $R \leftarrow \sum_{y \in N(e'_k)} [\tau(e'_k, y)]^\alpha \cdot [\eta(e'_k, y)]^\beta$ ;
6     Choose one neighbor  $e'_{k+1}$  of  $e'_k$  where the
       probability of selection of any fixed  $y \in N(e'_k)$  is
        $\frac{[\tau(e'_k, y)]^\alpha \cdot [\eta(e'_k, y)]^\beta}{R}$ ;
7   return  $\{e'_1, \dots, e'_{n-2}\}$ ;
```

gets a pheromone value $\tau_{(u,v)} = 1/|A|$). Afterwards, an initial solution x^* is produced using **Construct**.

We let h and ℓ be two pheromone values such that $h \geq \ell$ (high and low). These values may depend on the size n of the input graph. In each iteration, MMAS* will update the pheromone values according to a procedure **Update** such that $\text{Update}(\tau, x^*) = \tau'$, where

$$\forall (e, e') \in A : \tau'_{(e,e')} = \begin{cases} h, & \text{if } e' \in x^*; \\ \ell, & \text{otherwise.} \end{cases}$$

In other words, all edges leading to nodes which were chosen in the best-so-far solution x^* have pheromone value h , all others have ℓ .

The *fitness* of a candidate solution $p = (v_1, \dots, v_{n-2})$ is measured by the cost of the edges that lie between the two partitions V_1 and V_2 , i. e. the cut corresponding to p is given by the edges that lie between V_1 and V_2 . The fitness is given by the following function

$$f(p) := \sum_{e \in E \cap (V_1 \times V_2)} w(e).$$

The complete algorithm MMAS* is given in Algorithm 2.

Algorithm 2: The algorithm MMAS*.

```
1 function MMAS* on  $G$  is
2    $\tau_{(u,v)} \leftarrow 1/|A|$ , for all  $(u, v) \in A$ ;
3    $x^* \leftarrow \text{Construct}(C(G), \tau, \eta)$ ;
4   Update( $\tau, x^*$ );
5   while true do
6      $x \leftarrow \text{Construct}(G, \tau, \eta)$ ;
7     if  $f(x) > f(x^*)$  then
8        $x^* \leftarrow x$ ;
9      $\tau \leftarrow \text{Update}(\tau, x^*)$ ;
```

3. UPPER BOUNDS

In this section, we show upper bounds on the expected optimization time of MMAS*, dependent on the α and β and on the pheromone bounds h and ℓ . To gain such results we consider the initial run first and extend our analysis later to the whole run of the algorithm.

LEMMA 1. *Choosing $\beta = 1$, the probability that MMAS* produces a min-cut in the initial run is at least $2/(n(n-1))$.*

PROOF. We follow the ideas used in the analysis of the randomized algorithm of Karger and Stein [12] for computing a min-cut. Let $i, 3 \leq i \leq n$ be the number of connected components in the run of the construction procedure and k be the value of a min-cut C . Then the total weight of the edges that lie between these i components is at least $ik/2$ as otherwise C is not a min-cut of G . In the initial run the probability to choose an edge of C when having i components is at most

$$\frac{k}{ik/2} = 2/i.$$

Hence, the probability of *not* choosing an edge of C is at least $1 - 2/i$ and the probability of choosing no edge of C in the initial run of the MMAS* is at least

$$\prod_{i=3}^n (1 - 2/i) = 2/(n(n-1)).$$

This gives the desired probability. \square

For the case $\alpha = 0$ and $\beta = 1$, all solutions constructed by MMAS* have the same chance of being optimal as the initial one. This leads to the following upper bound on the expected optimization time.

COROLLARY 1. *Let $\alpha = 0$ and $\beta = 1$. The expected time for MMAS* to find a min-cut for any given weighted graph is $O(n^2)$.*

PROOF. As the pheromones do not have any impact when $\alpha = 0$, the algorithm behaves as if $\ell = h$. The result then follows from Theorem 1. \square

In the following we extend our analysis to the case $\alpha = \beta = 1$. We show an upper bound on the expected optimization time for MMAS* that depends on the ratio of the two pheromone bounds h and ℓ .

THEOREM 1. *Let $\alpha = 1, \beta = 1$ and $0 < \ell \leq h$, and let $c_n = \frac{h}{\ell}$. Then the expected optimization time of the MMAS* is*

- $O\left(\frac{(n-2+2c_n)!}{(n-2)!(2c_n)!}\right)$;
- in particular, if $c_n = k$ constant, $O(n^{2k})$.

PROOF. Let C be a min-cut of total weight k and let $i, 3 \leq i \leq n$ be the number of connected components in the run of the construction procedure. In the worst case, all edges of C have pheromone value h and all other edges have pheromone value ℓ . The probability of choosing an edge of C in step i is upper bounded by

$$\begin{aligned} \frac{hk}{hk + (ik/2 - k)\ell} &= \frac{h}{h + (i/2 - 1)\ell} \\ &= 1 - \frac{(i/2 - 1)\ell}{(i/2 - 1)\ell + h} \\ &= 1 - \frac{i - 2}{i - 2 + 2(h/\ell)}. \end{aligned}$$

Hence, the probability of *not* choosing an edge of C is at least $\frac{i-2}{i-2+2c_n}$ and the probability of choosing no edge of C during the construction of a new solution is at least

$$\prod_{i=3}^n \frac{i-2}{i-2+2c_n} = \frac{(n-2)!(2c_n)!}{(n-2+2c_n)!},$$

which leads to an expected optimization time of $O(n^{2c_n})$ if $c_n = k$ constant. \square

As a corollary, Theorem 1 implies that MMAS* is able to compute a minimum cut in expected polynomial time if the ratio between h and ℓ is not too large.

4. LOWER BOUNDS

In this section we will show lower bounds on the optimization time of MMAS* for a wide range of parameter settings. To show some of these lower bounds, we will use the following family of graphs.

DEFINITION 1. For all $n > 2$, let $V_n = \{1, \dots, n\} \times \{0, 1\}$ be the vertex set and let H_n be the complete graph on V_n with E_n the set of all of its edges. We assign weights to the edges of H_n such that, for all $(a, i), (b, j) \in V_n$,

$$w_n((a, i), (b, j)) = \begin{cases} 1, & \text{if } i = j; \\ n - 2, & \text{else if } a = 1 = b; \\ 0, & \text{otherwise.} \end{cases}$$

We let $\mathcal{H} = (H_n, w_n)_{n \in \mathbb{N}}$ denote the family of these graphs.

The weighted graph (H_n, w_n) is depicted in Figure 2.

Note that the weight on all edges *inside* each clique K_n is 1, and all and only the weight-0 edges are omitted. We will call the only edge of weight $n - 2$ the ‘‘heavy edge’’ e . Obviously, the optimal min-cut for (H_n, w_n) is just the heavy edge and the heavy edge never creates a cycle.

4.1 Lower Bounds for $\alpha = 1$ and $\beta = 1$

We start by considering MMAS* weighing heuristic information and pheromone information equally and point out where this parameter setting leads with high probability to an exponential optimization time.

THEOREM 2. Let $\alpha = 1$, $\beta = 1$ and $0 < \ell \leq h$, and let $c_n = \frac{h}{\ell}$ and $n \geq 5$. There are graphs where the optimization time of the MMAS* is 1 with probability $O(n^{-2})$, and, if the optimization time is strictly bigger than 1 and $c_n > n$, then the optimization time is $\left(\frac{c_n}{n}\right)^n$ with probability $1 - \left(\frac{n}{c_n}\right)^{\Omega(n)}$.

PROOF. We consider the family of graphs \mathcal{H} . The part about the optimization time of 1 with probability $O(n^{-2})$ follows from Lemma 1. Suppose now MMAS* is *not* successful in the initialization. Hence, the heavy edge e and $2n - 3$ other edges, have pheromone value h , while the less than $2n^2$ remaining edges have pheromone value ℓ .

Whenever e has pheromone value h , the probability of choosing e again in a step of the ant is *lower* bounded by

$$\begin{aligned} \frac{h(n-2)}{h(n-2) + (2n-3)h + 2\ell n^2} &= 1 - \frac{2\ell n^2}{h(3n-5) + 2\ell n^2} \\ &= 1 - \frac{2n^2}{c_n(3n-5) + 2n^2}. \end{aligned}$$

Hence, the probability of *not* choosing e in a run of the ant when $n \geq 5$, is at most

$$\left(\frac{2n^2}{c_n(3n-5) + 2n^2}\right)^{2n-2} \leq \left(\frac{2n^2}{c_n(3n-5)}\right)^{2n-2} \leq \left(\frac{n}{c_n}\right)^{2n-2}.$$

By union bound, the probability that the minimum cut has not been found within $\left(\frac{c_n}{n}\right)^n$ iterations is at least

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

As a corollary to Theorem 2, we get that, for $\alpha = 1$, $\beta = 1$ and $\frac{h}{\ell} \geq n(1 + \varepsilon)$, for any constant $\varepsilon > 0$, there are graphs where the expected optimization time of MMAS* is bounded below by an exponential function.

4.2 Lower Bounds for $\beta > 1$

In the following, we study the impact of the choice of β in greater detail. We have already seen that $\beta = 1$ may lead to an expected polynomial optimization time if $\alpha = 0$ holds. On the other hand, for $\beta = 1$ and $\alpha = 1$ there are graphs where MMAS* needs with high probability an exponential number of steps to obtain an optimal solution. As a consequence, we ask whether larger values of β can lead to optimal solutions in expected polynomial time. We answer this question negatively and point out that there are graphs where choosing $\beta > 1$ leads to an exponential optimization time regardless of the choice of α .

THEOREM 3. Choosing $\beta = 1 + \varepsilon$, $\varepsilon > 0$ a constant, the time until MMAS* has found a min-cut for H_n is $\exp(c_n^\varepsilon)$ with probability $1 - 2^{-\Omega(n^\varepsilon)}$, for some constant $c > 0$.

PROOF. We bound the probability that the heavy edge e will be chosen in any given choice of the algorithm from below if, for all edges e' , $\tau(e) \geq \tau(e')$. Note that the condition on the pheromone values will hold as long as e was chosen in each previous iteration. In the worst case no edge creates a cycle. Then, edge e will be chosen with probability at least

$$\begin{aligned} &\frac{\tau(e)^\alpha \cdot (n-2)^\beta}{\tau(e)^\alpha \cdot (n-2)^\beta + \sum_{e' \in E_n \setminus \{e\}} \tau(e')^\alpha w(e')^\beta} \\ &\geq \frac{\tau(e)^\alpha \cdot (n-2)^\beta}{\tau(e)^\alpha \cdot (n-2)^\beta + 2 \cdot \binom{n}{2} \tau(e)^\alpha} \\ &= \frac{(n-2)^\beta}{(n-2)^\beta + (n^2 - n)}, \end{aligned}$$

which, if $\beta > 2$, is lower bounded by a constant > 0 , and is otherwise lower bounded by $\frac{1}{4}n^{\beta-2}$ (we omit the easy algebraic transformations). Hence, if $\beta \leq 2$, the probability of *not* choosing the heavy edge in any given iteration is at most

$$\begin{aligned} \left(1 - \frac{1}{4n^{2-\beta}}\right)^{n-2} &= \left[\left(1 - \frac{1}{4n^{2-\beta}}\right)^{4n^{2-\beta}}\right]^{\frac{n-2}{4n^{2-\beta}}} \\ &\leq e^{-\frac{n-2}{n^{2-\beta}}} \\ &= e^{-\Omega(n^\varepsilon)}. \end{aligned}$$

For $\beta \leq 2$, using the union bound, the probability that the minimum cut has not been found within e^{n^ε} steps is at least $1 - e^{n^\varepsilon} \cdot e^{-\Omega(n^\varepsilon)} = 1 - e^{-\Omega(n^\varepsilon)}$. The case of $\beta \geq 2$ is similar. \square

4.3 Lower Bounds for $\alpha = 1$ and $\beta = 0$

After having studied the impact of heuristic information we analyze the impact of the pheromone value in greater detail. By setting $\beta = 0$, we remove all explicit biases towards choosing edges with larger weights. Hence the algorithm only relies on pheromone values to choose edges. In

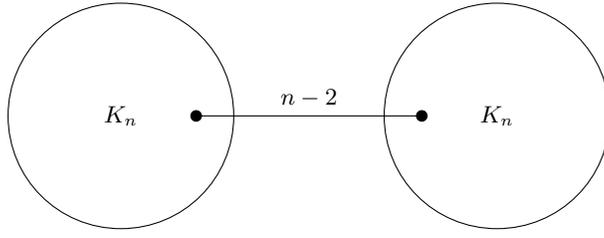


Figure 2: The Graph (H_n, w_n) .

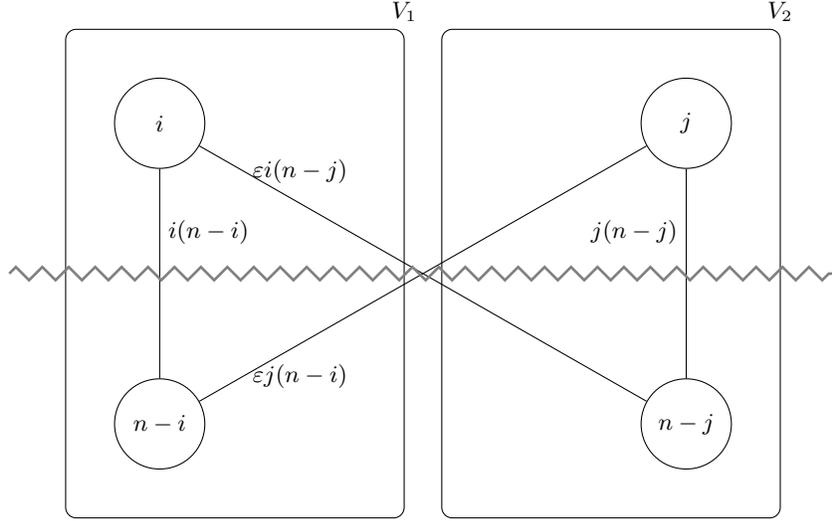


Figure 3: A Cut of the Graph (G_n, w_n) .

the following we will prove that there exist instance classes of the minimum cut problem where the algorithm requires exponential time with high probability (i.e. $1 - o(1)$).

DEFINITION 2. For all n , let G_n be the complete graph on the vertices $V = V_1 \cup V_2$ where $V_k := \{1, \dots, n\} \times \{k\}$. For any $\varepsilon, 0 < \varepsilon < 1/n$, we assign weights to the edges of G_n such that for all $(a, i), (b, j) \in V$,

$$w_n((a, i), (b, j)) := \begin{cases} \varepsilon & \text{if } i \neq j, \text{ and} \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

The graph G_n is depicted with a cut between P and $V \setminus P$ in Figure 3, where $i = |P \cap V_1|$ and $j = |P \cap V_2|$. The edges connecting nodes in V_1 with nodes in V_2 have weights ε , whereas edges connecting nodes internally within V_1 or V_2 have weights 1.

PROPOSITION 1. The minimum cut of G_n corresponds to the partition V_1 and V_2 , and has cost $\varepsilon n^2 < n$. Except for the minimum cut, no other cut has less cost than the cuts containing a single vertex v and $V \setminus \{v\}$.

PROOF. Let the partition be given by $P \subset V_1 \cup V_2$, and define $i := |P \cap V_1|$ and $j := |P \cap V_2|$. W.l.o.g., we can assume that $i \leq j$ and $i \leq n/2$. The cost of this cut is $(n-i)(i + \varepsilon j) + (n-j)(j + \varepsilon i)$ (see Figure 3).

For the partition $\{V_1, V_2\}$, we have $i = 0$ and $j = n$ and the cut has cost $\varepsilon n^2 := c^*$. For partitions containing only

a single vertex, we have $i = 0$ and $j = 1$, and the cut has cost $n(1 + \varepsilon) - 1 := c_l > c^*$. In any other partition, it is necessary that $i + j \geq 2$. In the case where $i = 0$ and $2 \leq j \leq n - 1$, the cut has cost at least $2\varepsilon n + 2(n - 2) > c_l$. Finally, if $1 \leq i \leq j \leq n - 1$, the cut has cost at least $(n - i)i + (n - j)j \geq 2(n - 1) > c_l$. \square

Firstly, we consider pheromone values such that the h/ℓ ratio is not too large. In the following theorem we prove that in such cases the MMAS* requires exponential time to find the minimum cut of G_n with overwhelming probability.

THEOREM 4. Let $\alpha = 1, \beta = 0$ and $c_n = h/\ell < n^{2-\delta}$ for any $\delta > 0$. Then the probability that MMAS* has found a minimum cut of G_n within $\exp(n^\delta/6)$ steps is at most $e^{-\Omega(n^\delta)}$.

PROOF. From Proposition 1 the minimum cut of G_n corresponds to the partition V_1 and V_2 . Hence, no ε -edges have to be chosen, when constructing the optimal solution. Let p_i be the probability of selecting an ε -edge in a construction step i , and $m := n(2n - 1)$ the total number of edges. For each construction step, this probability is always bounded

from below by

$$\begin{aligned} p_i &\geq \frac{(n^2 - i)\ell}{(2n - 2 - i)h + m\ell} \\ &\geq \frac{\ell n/2}{2h + 2n\ell} \\ &\geq \frac{1}{4(c_n/n + 1)} \\ &> \frac{1}{5n^{1-\delta}}. \end{aligned}$$

Thus, the probability of not choosing an ε -edge during each construction phase is at most

$$\left(1 - \frac{1}{5n^{1-\delta}}\right)^{2n-2} \leq e^{-n^\delta/5}.$$

Finally, taking the union bound we get a probability of at most $e^{-\Omega(n^\delta)}$ to find the minimum cut in $\exp(n^\delta/6)$ steps. \square

Hence the h/ℓ ratio should be at least n^2 . For the remainder of this paper, we choose h and ℓ such that $h = (m - 2n + 2)(\log n)\ell$ holds, which was shown to be successful for finding minimum spanning trees in graphs [15]. In this case, the probability of taking a rewarded edge (if applicable) is always at least $1 - 1/\log n$.

Let $m := n(2n - 1) \leq 2n^2$ be the total number of edges. Note that the ratio $\ell/h = 1/(m - 2n + 2) \log n$ is bounded by

$$\frac{1}{2n^2 \log n} \leq \frac{\ell}{h} \leq \frac{1}{n^2 \log n}.$$

We now present the main result of this section, which states that in the worst case the algorithm fails to find the minimum cut in polynomial time with high probability.

THEOREM 5. *For any constant $\delta, 0 < \delta < 1$, the probability that MMAS* has not found a minimum cut of G_n within e^{cn} steps is $1 - O(n^{-(1-\delta)})$, for some constant $c > 0$.*

The proof idea is first to show that with high probability, a partition v and $V \setminus \{v\}$ is constructed. Furthermore, with high probability, this partition will be constructed by choosing a large number of the ε -edges. Once such a partition is found, only the minimum cut will be accepted. Finally, this requires every selected ε -edge to be deselected in a single iteration, an event which happens with overwhelmingly low probability.

PROPOSITION 2. *The solution created in the first iteration of MMAS* is constructed by selecting at least $n/5$ ε -edges with probability $1 - e^{-\Omega(n)}$.*

PROOF. We only consider construction steps where at least one of the contracted components contains only a single node. In the initial construction step, there are $2n$ single-node components (i.e. all the nodes). A single-node component can be contracted by any of the $n - 1$ 1-edges and any of the n ε -edges. The probability that a single-node component is contracted by an ε -edge is therefore $n/(n + n - 1) \geq 1/2$, while the probability that it is contracted differently is less than $1/2$. Each of the $2n - 2$ contractions reduce the number of single-node components by at most 2, so the total number of contractions of single-node components is at least $(2n - 2)/2 \geq n/2$. Hence, the expected number of contractions of ε -edges is at least $n/4$, and by Chernoff bounds, the

probability that the number of contractions of ε -edges is less than $n/5$ is $e^{-\Omega(n)}$. \square

The probability of constructing a partition v and $V \setminus \{v\}$ for some node v depends on the number of leaf nodes in the current solution. We now show that the ‘‘contraction graph’’ is likely to have many leaf nodes. The initial construction of a solution in Algorithm 2 can be expressed equivalently in the following way. Obtain first a spanning tree T_n on the graph by contracting edges randomly as long as no cycles are introduced. Then remove a randomly chosen edge to obtain two trees $T_{n,1}$ and $T_{n,2}$. Clearly, the total number of leaf nodes in $T_{n,1}$ and $T_{n,2}$ is at least as large as the number of leaf nodes in the original tree T_n , assuming that single nodes are counted as leaf nodes. The probability that T_n contains less than βn leaf nodes for some constant $\beta > 0$ can be bounded from above using a result due to Gerke et al. [7], which we here express in a slightly less general form.

THEOREM 6 ([7]). *Let Y be the number of leaf nodes of T_n . Then there exists $\beta > 0$ such that*

$$\text{Prob}[Y \leq \beta n] < e^{-\beta n}.$$

We would like to show that the number of contracted ε -edges and the number of leaf nodes in the contraction graph do not decrease too rapidly. This follows from the next Proposition.

PROPOSITION 3. *Let H be the set of edges with pheromone level h after the first iteration of MMAS*, and c and δ any constants where $c > 0$ and $0 < \delta < 1$. The probability that, within n^δ iterations, more than cn edges in H have been given pheromone value ℓ at least once is $O(n^{-(1-\delta)})$.*

PROOF. In each construction step, selecting an edge with pheromone level ℓ can only reduce the number of h -edges that can be chosen later by 1. Hence, in construction step i , there are at least $2n - 2 - i$ edges with pheromone level h that can be chosen without introducing a cycle. Let p_i be the probability of selecting an edge with pheromone level ℓ in construction step i . For each construction step i , this probability is bounded from above by

$$p_i \leq \frac{\ell m}{(2n - 2 - i)h} \leq \frac{2}{\log n(2n - 2 - i)}.$$

Hence, the expected number of ℓ -edges chosen during one construction phase is

$$\sum_{i=1}^{2n-3} 1 \cdot p_i + 0 \cdot (1 - p_i) \leq \frac{2}{\log n} \sum_{i=1}^{2n-3} \frac{1}{2n - 2 - i} \leq k$$

for some constant k . The expected number of ℓ -edges chosen during n^δ iterations is therefore no more than kn^δ . By Markov’s inequality, the probability that more than cn ℓ -edges have been selected is therefore no more than $k/cn^{1-\delta}$. \square

PROPOSITION 4. *The probability of constructing a cut between partitions v and $V \setminus \{v\}$ in any iteration where the construction graph contains at least cn leaf nodes for any constant $c > 0$ is $\Omega(1)$.*

PROOF. Let v be any leaf node, and assume that the current partition is $A \cup \{v\}$ and B . The partition v and $A \cup B$, which is a local optimum, can now be constructed by deselecting the edge connecting node v with component A ,

and selecting one of the ℓ -edges between A and B . These ℓ -edges are at least $2n - 2$.

Consider the event that this happens for a particular leaf node v . To do this, we first estimate the probability that in all steps except step i , we re-select an h -edge except for the edge connecting node v to A . Then we estimate the probability that in step i , we select one of the $2n - 2$ ℓ -edges connecting partition A with partition B .

The probability of not selecting an h edge, assuming that only h -edges have been chosen until step $j - 1$ is no more than

$$\frac{(m - 2n + 2)\ell}{(2n - 2 - (j - 1))h} \leq \frac{1}{(2n - 1 - j) \log n}.$$

The probability that in all steps except step i , an h -edge is selected is therefore at least

$$\begin{aligned} 1 - \sum_{\substack{j=1 \\ j \neq i}}^{2n-2} \frac{1}{(2n - 1 - j) \log n} &\geq 1 - \frac{\ln(2n - 2) + 1}{\log n} + \frac{1}{\log n} \\ &\geq 1 - \frac{\ln(2n)}{\log n} = \Omega(1). \end{aligned}$$

The probability that in step i one of the $2n - 2$ ℓ -edges that connects A and B gets selected is at least

$$\begin{aligned} \frac{(2n - 2)\ell}{(2n - 2 - i)h + (m - 2n + 2)\ell} &\geq \frac{(2n - 2)\ell}{2(2n - 2 - i)h} \\ &\geq \frac{(n - 1)}{2(2n - 2 - i)n^2 \log n} \\ &\geq \frac{1}{3(2n - 2 - i)n \log n}. \end{aligned}$$

Hence, the probability that the partition v and $A \cup B$ will be produced is at least

$$\Omega(1) \cdot \sum_{i=1}^{2n-2} \frac{1}{3(2n - 2 - i)n \log n} = \Omega(1/n). \quad (2)$$

There are at least cn leaf nodes, hence the probability that any local optimum will be produced is $\Omega(1)$. \square

PROPOSITION 5. *If the current solution of MMAS* is a partition containing a single node, and this solution was constructed by contracting at least cn ε -edges, for any constant $c > 0$, then the probability of accepting a different solution in the next iteration is $e^{-\Omega(n)}$.*

PROOF. By Proposition 1, no solutions constructed by contracting at least one ε -edge have fitness strictly better than the current solution. Hence, in order to accept a new solution, it is necessary not to contract any ε -edge during $2n - 2$ construction steps. In any step of the constructing process, let i be the number of contracted edges with pheromone level h , and j be the number of contracted edges with pheromone level ℓ . Out of a total of $m \leq 2n^2$ edges, the total amount of pheromone on the edges that can still be contracted is no more than $(2n - 2 - i)h + (m - 2n + 2 - j)\ell \leq 2n(h + n\ell)$. Assuming that no ε -edge has been contracted so far, the probability of contracting an ε -edge in the next construction step is therefore at least

$$\frac{cnh}{2n(h + n\ell)} \geq \frac{c}{2 + \frac{2}{n \log n}} \geq c'$$

for some constant c' . Since the above probability holds for any i and j , the probability of never contracting an ε -edge during $2n - 2$ construction steps is no more than $(1 - c')^{2n-2} = e^{-\Omega(n)}$. \square

PROOF OF THEOREM 5. We consider two phases. Phase 1 lasts the first n^δ iterations, and Phase 2 lasts until the minimum cut has been found. For some constant $c > 0$, a failure occurs in Phase 1 if one of the following three events occur:

1. the number of leaf nodes in the construction graph is less than cn , or
2. the number of ε -edges in the construction graph is less than cn , or
3. a partition containing a single node v and $V \setminus \{v\}$ is not found before the end of the phase.

From Theorem 6, the number of leaf nodes in the initial iteration is at least βn with probability $1 - e^{-\Omega(n)}$. Replacing an h -node with an ℓ -node can only reduce the number of leaf nodes by 2. By Proposition 3, the probability that $(\beta/4)n$ h -edges, and hence $(\beta/2)n$ leaves have been lost within n^δ steps is $O(n^{-(1-\delta)})$. Multiplying, we get a failure probability for event 1 of $O(n^{-(1-\delta)})$. From Proposition 2, the number of ε -edges in the initial iteration is at least $n/5$ with probability $1 - e^{-\Omega(n)}$. In order to reduce the number of ε -edges, it is necessary not to choose an h -edge. Again, by Proposition 3, the probability that more than $cn/10$ h -edges, and hence $cn/10$ ε -edges have been lost within n^δ steps is $O(n^{-(1-\delta)})$. Multiplying, we get a failure probability for event 2 of $O(n^{-(1-\delta)})$. From Proposition 4, assuming event 2 occurred without a failure, the probability of finding a local optimum in any iteration is at least c' for some constant c' . Hence, the probability that a local optimum has not been found within Phase 1, and hence the failure probability of event 3 is $(1 - c')^{n^\delta} = e^{-\Omega(n^\delta)}$. Multiplying, the probability of any failure during Phase 1 is bounded by $1 - O(n^{-(1-\delta)})$.

If no failure event occurs during Phase 1, then by Proposition 5, the probability of finding the minimum cut in any iteration of Phase 2 is no more than $e^{-\Omega(n)}$. By union bound, the probability that the minimum cut has not been found within e^{cn} steps is at least $1 - e^{cn} \cdot e^{-\Omega(n)} = 1 - e^{-\Omega(n)}$ for sufficiently small c . \square

5. CONCLUSIONS

One of the most important challenges in recent research is to improve the theoretical foundations of general purpose algorithms, such as evolutionary algorithms and ant colony optimization. Especially, for ACO algorithms the theoretical understanding of how and why they work is rather weak.

With this paper, we have contributed to the theoretical understanding of this kind of algorithm by investigating the minimum cut problem. We have shown that a simple ACO algorithm solves this problem in expected polynomial time for a restricted parameter setting. On the other hand, we have pointed out that for other parameter settings there exist instances for which our ACO algorithm needs exponential time to achieve an optimal solution, yielding ACO inappropriate to find minimum cuts.

Acknowledgement

We thank Günter Rudolph for an initial discussion on the topic of this paper. Per Kristian Lehre was supported by

the EPSRC under grant no. EP/D052785/1. Timo Kötzing and Frank Neumann were supported by the Deutsche Forschungsgemeinschaft (DFG) under grant no. NE 1182/5-1.

Pietro Simone Oliveto was supported by the EPSRC PhD+ grant no. EP/P502322/1.

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(3):88–92, 2008.
- [2] B. Doerr, F. Neumann, D. Sudholt, and C. Witt. On the runtime analysis of the 1-ANT ACO algorithm. In *GECCO '07: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 33–40, New York, 2007. ACM Press.
- [3] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344:243–278, 2005.
- [4] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
- [5] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.
- [6] A. Duarte, Á. Sánchez, F. Fernández, and R. Cabido. A low-level hybridization between memetic algorithm and VNS for the max-cut problem. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 999–1006, 2005.
- [7] S. Gerke, D. Schlatter, A. Steger, and A. Taraz. The random planar graph process. *Random Structures and Algorithms*, 32(2):236–261, 2008.
- [8] W. J. Gutjahr. First steps to the runtime complexity analysis of Ant Colony Optimization. *Computers and Operations Research*, 35(9):2711–2727, 2008.
- [9] 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.
- [10] C. Horoba and D. Sudholt. Running time analysis of ACO systems for shortest path problems. In T. Stützle, M. Birattari, and H. H. Hoos, editors, *SLS*, volume 5752 of *Lecture Notes in Computer Science*, pages 76–91. Springer, 2009.
- [11] T. Jansen and I. Wegener. Evolutionary algorithms - how to cope with plateaus of constant fitness and when to reject strings of the same fitness. *IEEE Transactions on Evolutionary Computation*, 5(6):589–599, 2001.
- [12] D. R. Karger and C. Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, 1996.
- [13] F. Neumann, J. Reichel, and M. Skutella. Computing minimum cuts by randomized search heuristics. In C. Ryan and M. Keijzer, editors, *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 779–786. ACM, 2008. Journal version appears in *Algorithmica*.
- [14] F. Neumann, D. Sudholt, and C. Witt. Analysis of different MMAS ACO algorithms on unimodal functions and plateaus. *Swarm Intelligence*, 3(1):35–68, 2009.
- [15] F. Neumann and C. Witt. Ant colony optimization and the minimum spanning tree problem. In V. Maniezzo, R. Battiti, and J.-P. Watson, editors, *LION*, volume 5313 of *Lecture Notes in Computer Science*, pages 153–166. Springer, 2007.
- [16] F. Neumann and C. Witt. Runtime analysis of a simple ant colony optimization algorithm. *Algorithmica*, 54(2):243–255, 2009.
- [17] P. S. Oliveto, J. He, and X. Yao. Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, 4(3):281–293, 2007.
- [18] J. Puchinger, G. R. Raidl, and G. Koller. Solving a real-world glass cutting problem. In *Proc. of EvoCOP '04*, pages 165–176. Springer, 2004.
- [19] V. Vazirani. *Approximation Algorithms*. Springer, 2001.
- [20] I. Wegener. Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. In R. Sarker, X. Yao, and M. Mohammadian, editors, *Evolutionary Optimization*, pages 349–369. Kluwer, Dordrecht, Netherlands, 2002.
- [21] Y. Zhou. Runtime analysis of an ant colony optimization algorithm for TSP instances. *IEEE Transactions on Evolutionary Computation*, 13(5):1083–1092, 2009.