UNIVERSITY OF DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes
and Systems by means of Computational Intelligence Methods

On the Influence of Pheromone Updates in ACO Algorithms

Benjamin Doerr, Frank Neumann,
Dirk Sudholt and Carsten Witt

No. CI-223/07

Technical Report ISSN 1433-3325 January 2007

Secretary of the SFB 531 · University of Dortmund · Dept. of Computer Science/LS 2
44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the University of Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

On the Influence of Pheromone Updates in ACO Algorithms

Benjamin Doerr
Algorithms and Complexity Group
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Frank Neumann
Algorithms and Complexity Group
Max-Planck-Institut für Informatik
Saarbrücken, Germany
fne@mpi-sb.mpg.de

Dirk Sudholt*
Fachbereich Informatik, LS 2
Universität Dortmund
Dortmund, Germany
dirk.sudholt@cs.uni-dortmund.de

Carsten Witt*
Fachbereich Informatik, LS 2
Universität Dortmund
Dortmund, Germany
carsten.witt@cs.uni-dortmund.de

Abstract

The runtime analysis of randomized search heuristics is a growing field where, in the last two decades, many rigorous results have been obtained. These results, however, apply particularly to classical search heuristics such as Evolutionary Algorithms (EAs) and Simulated Annealing. First runtime analyses of modern search heuristics have been conducted only recently w. r. t. a simple Ant Colony Optimization (ACO) algorithm called 1-ANT. In particular, the influence of the evaporation factor in the pheromone update mechanism and the robustness of this parameter w. r. t. the runtime have been determined for the example function OneMax.

This paper puts forward the runtime analysis of the 1-ANT on example functions. With respect to EAs, such analyses have been essential to develop methods for the analysis on more complicated problems. The proof techniques required for the 1-ANT, unfortunately, differ significantly from those for EAs, which means that a new reservoir of methods has to be built up. Again, the influence of the evaporation factor is analyzed rigorously, and it is proved that its choice can be very crucial to allow efficient runtimes. Moreover, the analyses provide insight into the working principles of ACO algorithms and, in terms of their robustness, describe essential differences to other randomized search heuristics.

1 Introduction

The analysis of randomized search heuristics is a growing research area where many results have been obtained in recent years. This class of heuristics contains not only well-known approaches such as Randomized Local Search, Simulated Annealing, and Evolutionary Algorithms (EAs) but also less-known and more modern instances such as Estimation-of-Distribution Algorithms, Particle Swarm Optimization, and Ant Colony Optimization (ACO). Such heuristics are often applied to problems

*This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as a part of the Collaborative Research Center "Computational Intelligence" (SFB 531)

whose structure is not known or if there are not enough resources such as time, money, or knowledge to obtain good specific algorithms. It is widely acknowledged that a solid theoretical foundation for such heuristics is needed. An obvious and accepted theoretical approach stemming from theoretical computer science is to analyze the (expected) runtime of randomized search heuristics by adapting the probabilistic methods available for the analysis of randomized algorithms, see, e. g., Feller (1968, 1971) and Motwani and Raghavan (1995).

The first steps to a runtime analysis of randomized search heuristics were made for a very simple EA called (1+1) EA. Initially, the (1+1) EA was investigated for the optimization of example functions such as OneMax, LeadingOnes, BinVal, trap functions etc. (Droste, Jansen and Wegener, 2002), all of which indubitably may be regarded as toy problems. Due to the simple structure of these problems, however, it was possible to develop methods for the analysis of EAs. This approach can be considered very successful since nowadays runtime analyses of EAs can be carried out w. r. t. well-known combinatorial optimization problems such as maximum matchings (Giel and Wegener, 2003), minimum spanning trees (Neumann and Wegener, 2004), partition problems (Witt, 2005), Eulerian cycle problems (Doerr, Hebbinghaus and Neumann, 2006), and graph coloring problems (Sudholt, 2005).

The theoretical runtime analysis for the modern and very popular randomized search heuristic ACO (e. g., Dorigo and Stützle, 2004), however, lags far behind the results for the classical EAs. Until 2006, only convergence results (e. g., Gutjahr, 2003) and results on the dynamics of models of ACO (e. g., Merkle and Middendorf, 2002) were known. In a survey on theoretical studies of ACO by Dorigo and Blum (2005), researchers were encouraged to follow the approach taken for the analysis of EAs by starting a rigorous runtime analysis of simple ACO algorithms on OneMax. Soon after this appeal, first steps towards such analyses appeared in a report by Gutjahr (2006a), and, independently, the first theorems on the runtime of a simple ACO algorithm were published in a work by Neumann and Witt (2006b). In that paper, a simple ACO algorithm called 1-ANT is defined based on the model of Gutjahr (2003) and the runtime w. r. t. the fitness function OneMax is bounded from above and below. It is shown that the so-called evaporation factor ρ, the probably most important parameter in ACO algorithms, has a crucial impact on the runtime. More precisely, it is proved that there exists a threshold value for ρ below which no efficient optimization is possible. A lesson learnt from the results is that the 1-ANT may not be robust w. r. t. the choice of ρ.

A closer look at the paper by Neumann and Witt (2006b) reveals that the mathematical methods employed for the analysis of the 1-ANT differ heavily from those for the analysis of EAs. Even more conspicuously, it seems that the mathematical tools are tailored for the symmetric function OneMax. It is by no means clear whether a comprehensive runtime analysis of the 1-ANT can be conducted on more complicated problems. A recent analysis of the 1-ANT on the combinatorial minimum spanning tree problem by the same authors (Neumann and Witt, 2006a) basically considers a special case of the 1-ANT with two pheromone values and fails to deliver statements on the choice of ρ.

The aim of this paper is to put forward the analysis of the 1-ANT on example problems in a similar fashion to Neumann and Witt (2006b). As Gutjahr (2006b) has observed, such analyses are an important and emergent issue in the community of ACO. We choose the non-symmetric functions LeadingOnes and BinVal investigated by Droste, Jansen and Wegener (2002) and analyze the runtime of the 1-ANT on these functions w. r. t. n, the dimensionality of the search space, and the evaporation factor ρ. It turns out that a similar phase transition behavior can be observed as by Neumann and Witt (2006b). If ρ is asymptotically smaller than a threshold, no efficient optimization is possible; however, for values a little above the threshold, polynomial runtimes are very likely. Hence, our investigations again suggest that the 1-ANT is not robust w. r. t. the choice of ρ. The proofs contribute new methods for the runtime analysis of ACO algorithms and may serve as a basis for further theoretical studies.

The outline of the paper is as follows. In Section 2, we provide the necessary definitions and recapitulate the previous results for OneMax. Section 3 is a technical part, where general properties of the pheromone update mechanism of the 1-ANT are summarized and proved. The following Sections 4 and 5 deal with the main results of the paper, namely lower and upper bounds on the runtime of the 1-ANT on LeadingOnes, respectively. A generalization of the results to the function BinVal is discussed in Section 6. The paper is finished with some conclusions.

2 The Algorithm

ACO algorithms construct solutions by random walks on so-called constructions graphs. This random walk is influenced by values on the edges called pheromone values. In addition, the walk may be influenced by heuristic information about the problem. We use the setting of Neumann and Witt (2006b) where no heuristic information is used. As in Neumann and Witt (2006b), our main aim is to consider the effect of the pheromone update in a simple ACO algorithm called 1-ANT (see Algorithm 2) and to analyze its effect on the runtime for growing sizes of the optimization problem.

Let $C = (V, E)$ be the construction graph with a designated start vertex s and pheromone values τ on the edges. Starting at s, an ant traverses the construction graph depending on the pheromone value using Algorithm 1. Assuming that the ant is at vertex v, the ant moves to a successor w of v, where w is chosen proportionally to the pheromone values of all non-visited successors of v. The process is iterated until a situation is reached where all successors of the current vertex v have been visited.

Algorithm 1 (Construct(C, τ))

```
1.) v:=s, mark v as visited.
2.) While there is a successor of v in C that has not been visited:
    a.) Let $N_v$ be the set of non-visited successors of v and $T := \sum_{(v,w)|w \in N_v} \tau_{(v,w)}$.
    b.) Choose one successor w of v where the probability of selection of any fixed
        $u \in N_v$ is $\tau_{(v,u)}/T$.
    c.) Mark w as visited, set $v := w$ and go to 2.).
3.) Return the solution x and the path $P(x)$ constructed by this procedure.
```

In the initialization step of the 1-ANT, each edge gets a pheromone value of $1/|E|$ such that the pheromone values sum up to 1. After that, an initial solution x^* is produced by a random walk on the construction graph and the pheromone values are updated with respect to this walk. In each iteration, a new solution x is constructed and the pheromone values are updated if this solution is not inferior to the currently best solution x^*. We formulate our algorithm for maximization problems although it can be easily adapted to minimization.

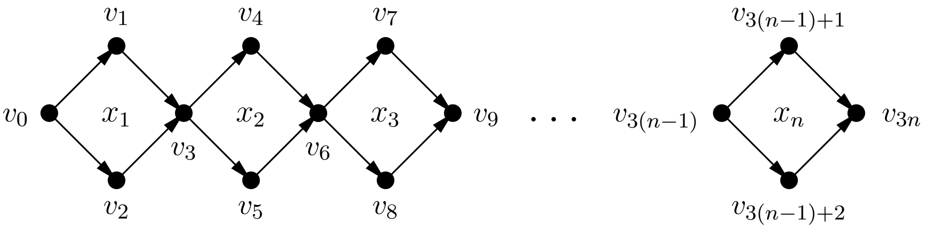

Figure 1: Construction graph for pseudo-Boolean optimization

Algorithm 2 (1-ANT)

```
1.) Set $\tau_{(u,v)} = 1/|E|$ for all $(u,v) \in E$.
2.) Compute $x$ (and $P(x)$) using Construct($C, \tau$).
3.) Update($\tau, P(x)$) and set $x^* := x$.
4.) Compute $x$ (and $P(x)$) using Construct($C, \tau$).
5.) If $f(x) \geq f(x^*)$, Update($\tau, P(x)$) and set $x^* := x$.
6.) Go to 4.).
```

For theoretical investigations, it is common to have no termination condition in such an algorithm. One is interested in the random optimization time which equals the number of constructed solutions until the algorithm has produced an optimal search point. Often one tries to bound the expected value of this time.

Considering the optimization for pseudo-Boolean fitness functions $f\colon \{0,1\}^n \to \mathbb{R}$ with $n \geq 3$, we use the construction graph $C_{\text{bool}} = (V, E)$ (see Figure 1) with $s = v_0$. In the literature, this graph is also known as *Chain* (Gutjahr, 2006a). Optimizing bit strings of length n, the graph has $3n + 1$ vertices and $4n$ edges. The decision whether a bit x_i, $1 \leq i \leq n$, is set to 1 is made at node $v_{3(i-1)}$. In case that the edge $(v_{3(i-1)}, v_{3(i-1)+1})$ is chosen, x_i is set to 1 in the constructed solution. Otherwise $x_i = 0$ holds. After this decision has been made, there is only one single edge which can be traversed in the next step. In case that $(v_{3(i-1)}, v_{3(i-1)+1})$ has been chosen, the next edge is $(v_{3(i-1)+1}, v_{3i})$, and otherwise the edge $(v_{3(i-1)+2}, v_{3i})$ will be traversed. Hence, these edges have no influence on the constructed solution and we can assume

$$\tau_{(v_{3(i-1)}, v_{3(i-1)+1})} = \tau_{(v_{3(i-1)+1}, v_{3i})}$$

and

$$\tau_{(v_{3(i-1)}, v_{3(i-1)+2})} = \tau_{(v_{3(i-1)+2}, v_{3i})}$$

for $1 \leq i \leq n$. We call the edges $(v_{3(i-1)}, v_{3(i-1)+1})$ and $(v_{3(i-1)+1}, v_{3i})$ 1-*edges* and the other edges 0-*edges*.

The pheromone values are chosen such that at each time

$$\sum_{(u,v) \in E} \tau_{(u,v)} = 1$$

holds. In addition, it seems to be useful to have bounds on the pheromone values (e. g., Dorigo and Blum, 2005) to ensure that each search point has a positive probability of being chosen in the next step. We restrict each $\tau_{(u,v)}$ to the interval $\left[\frac{1}{2n^2}, \frac{n-1}{2n^2}\right]$ and ensure $\sum_{(u,\cdot) \in E} \tau_{(u,\cdot)} = \frac{1}{2n}$ for $u = v_{3i}$, $0 \leq i \leq n-1$, and $\sum_{(\cdot,v)} \tau_{(\cdot,v)} = \frac{1}{2n}$ for $v = v_{3i}$, $1 \leq i \leq n$. This can be achieved by normalizing the pheromone values after an update and replacing the current value by $\frac{1}{2n^2}$ if $\tau_{(u,v)} < \frac{1}{2n^2}$ and by $\frac{n-1}{2n^2}$ if $\tau_{(u,v)} > \frac{n-1}{2n^2}$ holds. Depending on whether edge (u, v) is contained in the path $P(x)$ of the accepted solution x, the pheromone values are updated to τ' in the procedure Update$(\tau, P(x))$ as follows:

$$\tau'_{(u,v)} = \min\left\{\frac{(1-\rho)\cdot\tau_{(u,v)} + \rho}{1-\rho+2n\rho}, \frac{n-1}{2n^2}\right\} \quad \text{if } \ (u,v) \in P(x)$$

and

$$\tau'_{(u,v)} = \max\left\{\frac{(1-\rho)\cdot\tau_{(u,v)}}{1-\rho+2n\rho}, \frac{1}{2n^2}\right\} \quad \text{if } \ (u,v) \notin P(x).$$

Let $p_i = \text{Prob}(x_i = 1)$, $1 \leq i \leq n$, be the probability of setting the bit x_i to one in the next constructed solution. A consequence of the described setting is that $p_i \in [1/n, 1 - 1/n]$ holds due to the upper and lower bounds on the pheromone values.

In the paper by Neumann and Witt (2006b), it has been shown that the introduced 1-ANT behaves for $\rho \geq (n-2)/(3n-2)$ as the well-known evolutionary algorithm called (1+1) EA. In this case, the pheromone values always attain their upper or lower bounds after the first update has occurred. For the function OneMax with

$$\text{OneMax}(x) = \sum_{i=1}^{n} x_i$$

the influence of ρ has been analyzed in greater detail. Neumann and Witt have shown that there is a phase transition from exponential to polynomial runtime as ρ grows. In particular they have given an exponential lower bound for the case $\rho = O(n^{1-\varepsilon})$ and a polynomial upper bound for $\rho = \Omega(n^{1+\varepsilon})$ where $\varepsilon > 0$ is in each case a positive constant. The main argument for the lower bound is that the value of the currently best solution and the expected value of the one constructed in the next iteration may differ in the function value that leads to an exponential optimization time. In contrast to this, the polynomial upper bound relies on the observation that the function value of the last accepted solution determines the expected value of the next constructed solution almost exactly.

We consider the function LeadingOnes (proposed by Rudolph, 1997) with

$$\text{LeadingOnes}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j,$$

whose function value equals the number of leading ones in the considered bit string x. A non-optimal solution may always be improved by appending a single one to the leading ones. LeadingOnes differs from OneMax in the essential way that the assignment of the bits after the leading ones do not contribute to the function value. This implies that bits at the beginning of the bit string have a stronger influence on the function value than bits at the end. Because of this, the methods developed by Neumann and Witt (2006b) cannot be used for analyzing the 1-ANT on LeadingOnes as these methods make particular use of the fact that all bits contribute equally to the function value. We will develop new methods to deal with the circumstance that different bits may have different priorities for the optimization process. A well-known linear function that relies on the different priorities is BinVal (introduced by Droste, Jansen and Wegener, 2002) defined as

$$\text{BinVal}(x) = \sum_{i=1}^{n} 2^{n-i} x_i,$$

which interprets a bit string as the binary representation of an integer. After having analyzed the 1-ANT on LeadingOnes, we will show how to adapt the developed methods for analyzing the 1-ANT on BinVal.

3 On the Pheromone Update Mechanism

To analyze the 1-ANT for pseudo-Boolean optimization, it is necessary to understand the interrelation of pheromone values and success probabilities at single bits. As defined in Section 2, for each bit, there is a pair of 1-edges and a pair of 0-edges. The pheromone values on the edges in a pair are always the same. Therefore, we speak of only a single 1-edge and a single 0-edge for each bit when considering the pheromone values for a bit. We already know that the probability of setting a bit to 1 is proportional to the pheromone value on the corresponding 1-edge by a factor of $2n$. An analogous statement holds for the probability of setting the bit to 0 and the 0-edge. Finally, throughout the paper it is crucial to note that the bits are processed independently by the 1-ANT.

Consider an arbitrary but fixed bit x_i. If this bit is set to 1 in the next constructed solution, we speak of a success and a failure otherwise. Obviously, if a success occurs, the success probability in the next step is increased, too. However, the amount of increase depends on the previous pheromone value on the 1-edge (or, equivalently, the previous success probability). The larger the pheromone value, the lower the amount of increase will be.

In the following, we gather some formulas describing how the success probabilities interrelate and how they increase. Moreover, we estimate the number of successes sufficient to reach the upper bound $1 - 1/n$ on the success probability. All following estimations in this section assume that this bound has not been broken yet.

Definition 1 *Let p be the current success probability of a certain bit. Let $I^{(t)}(p)$ be the cumulative increase of its success probability after $t > 0$ successes and no failures at the bit.*

We prove some properties such as monotonicity of the increases $I^{(t)}(p)$ and of the actual success probabilities.

Lemma 1 *If $p \le q$ then $I^{(t)}(p) \ge I^{(t)}(q)$ for any $t > 0$. However, $p + I^{(t)}(p) \le q + I^{(t)}(q)$. Finally, $I^{(1)}(p) = \frac{2n\rho}{1-\rho+2n\rho} \cdot (1-p)$.*

Proof: It is sufficient to show the claims for $t = 1$, and we omit the index 1 in $I^{(1)}$ for readability. By the update mechanism of the 1-ANT and the proportionality of success probabilities and pheromone updates,

$$I(p) = 2n \cdot \frac{(1-\rho)(p/(2n)) + \rho}{1-\rho+2n\rho} - p = \frac{2n\rho}{1-\rho+2n\rho} \cdot (1-p).$$

This proves the first and the third claim. For the second claim, we assume $p \le q$ and consider the difference $q + I(q) - (p + I(p))$. By the expression just derived, the difference equals

$$(q-p) + (p-q) \cdot \frac{2n\rho}{1-\rho+2n\rho} \ = \ (q-p) \cdot \frac{1-\rho}{1-\rho+2n\rho} \ \ge \ 0.$$

□

By the preceding lemma, we justify in our forthcoming analyses the places where actual success probabilities are replaced with lower bounds on these probabilities. We define some worst-case lower bounds.

Definition 2 *Let $p^{(0)} := 1/n$ and $p^{(t)} := p^{(0)} + I^{(t)}(p^{(0)})$.*

After t successes and no failures at a bit, the success probability is at least $p^{(t)}$. We are interested in concrete lower bounds on $p^{(t)}$.

Lemma 2 $p^{(t)} \geq 1 - \left(\frac{1}{1+2n\rho}\right)^t$.

Proof: We consider the difference $p^{(t)} - p^{(t-1)}$ for $t \geq 1$. Recalling Lemma 1, we can write $p^{(t)} - p^{(t-1)} = I^{(1)}(p^{(t-1)}) = R \cdot (1 - p^{(t-1)})$, where $R := (2n\rho)/(1-\rho+2n\rho)$. This implies the recursion $1 - p^{(t)} = (1-R)(1-p^{(t-1)})$. Hence,

$$1 - p^{(t)} = (1-R)^t \cdot (1 - p^{(0)}) \;\leq\; (1-R)^t.$$

By definition,

$$1 - R \;=\; \frac{(1-\rho+2n\rho) - 2n\rho}{1-\rho+2n\rho} \;=\; \frac{1-\rho}{1-\rho+2n\rho}.$$

The last fraction is at most $1/(1+2n\rho)$. Hence, $1 - p^{(t)} \leq 1/(1+2n\rho)^t$, which proves the lemma. □

We are interested in the number of successes necessary to increase the success probability of the bit to a certain value r or above, where $1/n < r < 1 - 1/n$.

Definition 3 *Let $N(r) := \min\{t \mid p^{(t)} \geq r\}$. Moreover, let $N^* := \max\{t \mid p^{(t)} < 1 - 1/n\}$.*

Lemma 3 $N(r) \leq \frac{-\log(1-r)}{\log(1+2n\rho)}$.

Proof: For $p^{(t)} \geq r$, which is equivalent to $1 - p^{(t)} \leq 1 - r$, it is by Lemma 2 sufficient that

$$\left(\frac{1}{1+2n\rho}\right)^t \;\leq\; 1 - r.$$

Taking logarithms and observing that the base is smaller than 1, we obtain that the choice

$$t(r) = \left\lceil \frac{-\log(1-r)}{\log(1+2n\rho)} \right\rceil$$

implies that $p^{(t(r))} \geq r$. □

The quantity $N^* + 1$ denotes how many of the $p^{(t)}$ are less than the common upper bound $1 - 1/n$.

Lemma 4 $N^* \leq \frac{\log(n)}{\log(1+2n\rho)} \leq \frac{\log n}{\log n - \log(1/\rho)}$. *If $\rho \leq 1/n$ then $N^* \leq \frac{2\log n}{n\rho}$.*

Proof: The first inequality follows by setting $r := 1 - 1/n$ and applying Lemma 3. For the second inequality, we use the trivial estimation $\log(1+2n\rho) \geq \log(n\rho) = \log(n) - \log(1/\rho)$. If $\rho \leq 1/n$ then we use $\log(1+2n\rho) \geq n\rho/2$ since $e^x \leq 1 + 4x$ for $x \leq 2$. □

4 Lower Bound on LeadingOnes

In this section, we show that the 1-ANT is very inefficient on LeadingOnes if $\rho = o(1/(n \log n))$, i. e., if ρ is asymptotically smaller than $1/(n \log n)$. The following theorem shows that then even polynomially many multistarts fail within polynomial time with overwhelming probability.

One of the main reasons for the failure is that with the small evaporation factor, the success probabilities at single bits can reach large enough values only slowly. In consequence, the 1-ANT is faster in finding good solutions than in storing this knowledge in the pheromone values. This leads to the claimed time bound already for (re-)detecting an acceptable solution.

Theorem 1 *With probability* $1 - 2^{-\Omega(\min\{1/(n\rho),n\})}$*, the runtime of the 1-ANT on* LeadingOnes *is* $2^{\Omega(\min\{1/(n\rho),n\})}$.

Note that the stated lower bound is only meaningful if $\rho = O(1/(n \log n))$. However, if we choose, e. g., $\rho = 1/(n \log^2 n)$, the bound is already superpolynomially large.

Proof: Let $k = 1/(8n\rho)$. Consider the state of the 1-ANT at the earliest time when one of the following two conditions is fulfilled.

(i) The fitness f_c of the current solution is at least $n/2$.

(ii) 1-ANT has performed k accepted steps.

We first convince ourselves that in this situation, all success probabilities never left the interval $[1/4, 3/4]$, then, that with high probability we are not done yet, and finally, that the next accepted step takes the time claimed in the theorem.

Consider the success probability of a certain bit. From Lemma 1, we see that in each accepted step, it increases by at most $2n\rho/(1 - \rho + 2n\rho) \leq 2n\rho$. A simple induction together with the monotonicity statement of Lemma 1 shows that in at most k accepted steps, independent of the particular accepted solutions, the success probability can increase by at most $k \cdot 2n\rho$. Hence the success probability p after k such steps is bounded by $p \leq 1/2 + 2nk\rho \leq 3/4$. By symmetry, the same holds for the failure probability $1 - p$. Hence $p \in [1/4, 3/4]$.

Now let us regard the last (accepted) step before the system reached the state fixed above. At the start of this step, we have all success probabilities in $[1/4, 3/4]$ and our current solution has fitness $f_0 < n/2$. Hence the probability that the 1-ANT finds the optimal solution in this single step is bounded by $(3/4)^{(n/2)}$—recall that we know already that this step will be accepted, hence the first f_0 bits of this solution are one with probability one. Nevertheless, we see that with probability $1 - 2^{-\Omega(n)}$, we have not found the optimum yet.

Finally, let us estimate the time to obtain an accepted step from the state fixed above. We first estimate the current fitness f_c. If we did not perform k accepted steps, then clearly $f_c \geq n/2$. Hence let us assume that we actually did perform k accepted steps. Conditional on the fact that a step was accepted, the probability that this lead to a fitness increase is at least $1/4$ since the probability of a success at the leftmost zero-bit is at least $1/4$. Hence $E(f_c) \geq k/4$, and the usual Chernoff bounds imply $\mathrm{Prob}(f_c \geq k/8) = 1 - 2^{-\Omega(k)} = 1 - 2^{-\Omega(1/(n\rho))}$. Combining the two cases, we have

$$\mathrm{Prob}\big(f_c \geq \min\{n/2, k/8\}\big) \;=\; 1 - 2^{-\Omega(1/(n\rho))}.$$

If $f_c \geq \min\{n/2, k/8\}$, then the probability that the next step is accepted is at most $(3/4)^{\min\{n/2,\lfloor k/8 \rfloor\}}$, i. e., at most $2^{-\Omega(\min\{n,1/(n\rho)\})}$. Consequently, there is some small constant $c > 0$ such that a phase consisting of $2^{c \cdot \min\{n,1/(n\rho)\}}$ steps does not produce another accepted step with probability $1 - 2^{-\Omega(\min\{n,1/(n\rho)\})}$. □

5 Upper Bound on LeadingOnes

In contrast to the situation from the last section, large values of ρ allow the 1-ANT to rediscover the leadings ones of previous solutions efficiently. To prove the following theorem, we make heavy use of the observations from Section 3.

Recall the notations $p^{(t)}$ and N^* from Definitions 2 resp. 3. First, we need a lower bound on the product of the fraction $p^{(t)}/(2-p^{(t)})$ for all $p^{(t)}$ below $1-1/n$, which will be used to lower bound the probability of creating an accepted solution.

Lemma 5

$$\prod_{t=1}^{N^*} \frac{p^{(t)}}{2-p^{(t)}} \;=\; \Omega\Big(2^{-9/(n\rho)}\Big).$$

A proof of this lemma is given at the end of this section.

Theorem 2 *The expected runtime of the 1-ANT on* LeadingOnes *is bounded above by* $O(n^2 \cdot 2^{9/(n\rho)})$.

Note that the bound is polynomial for $\rho = \Omega(1/(n \log n))$ and only $O(n^2)$ for $\rho = \Omega(1/n)$. For $\rho = o(1/(n \log n))$, it is superpolynomially large.

Proof: We show that the probability of increasing the so far maximum LeadingOnes-value is always bounded below by $\Omega(2^{-9/(n\rho)}/n)$ provided the optimum has not been reached. Multiplying the expected time for an improvement by the maximum number of improvements, n, will yield the theorem.

Suppose the currently best LeadingOnes-value equals $k < n$. For an improvement, it is necessary and sufficient to set the first $k+1$ bits in a newly constructed solution to 1. Since the $(k+1)$-st bit was set to 0 in the last accepted solution, its success probability, i. e., the probability of being set to 1, was decreased in the last pheromone update. Therefore, we estimate the success probability from below by $1/n$ for this bit. If we can prove that the first k bits are all set to 1 with probability at least $\Omega(2^{-9/(n\rho)})$, we obtain the theorem.

The success probabilities for the first k bits may differ significantly. However, it is crucial that these probabilities are non-decreasing in all following pheromone updates and that at least one increase has happened due to the last accepted solution. There may be a block of $\ell < k$ leading bits whose success probabilities have reached the upper bound $1-1/n$. All these bits will be set to 1 in the next constructed solution with probability at least $(1-1/n)^\ell = \Omega(1)$. We concentrate on the window of size $k-\ell$ from the $(\ell+1)$-st bit to the k-th bit. Obviously, for $\ell+1 \le i \le k-1$, it holds that the success probability of the i-th bit has been increased at least as often as the one of the $(i+1)$-st bit.

If a new solution could only increase the best LeadingOnes-value by at most 1, we could estimate the probabilities in the window by bounding the success probability of the i-th bit from below by $p^{(k-i+1)}$ for $\ell+1 \le i \le k$. Hence, the probability of having only successes in the window would be at least $\prod_{t=1}^{N^*} p^{(t)}$, which can be bounded below by $\Omega(2^{-9/(n\rho)})$ using the techniques from Lemma 5.

However, the assumption that the best LeadingOnes-value increases by at most 1 is unrealistic since we are faced with additional one-bits gained in accepted solutions, so-called free-riders (see Droste, Jansen and Wegener, 2002). We observe that the rightmost $n-k-1$ bits (following the leftmost zero at position $k+1$) do not contribute to the LeadingOnes-value, implying that the corresponding success probabilities are governed by a random walk on the pheromone values. If a new solution increases the best LeadingOnes-value from k to $k+g$, $g \ge 1$, we obtain a block of g bits that will form the right end of the window w. r. t. the next improvement. All of these g bits have to be successful then. The success probability for each bit in the block can only be bounded below by $p^{(1)}$ after the improving step and, generally, only by $p^{(t)}$ after another $t-1$ improvements. Therefore, the probability of all bits in the block being successful is only bounded from below by $\bigl(p^{(t)}\bigr)^g$. This suggests to show that large values of g are unlikely.

Due to the symmetry of the pheromone update process for the possible free-rider bits, we can show that each of these bits is independently set to 1 with probability 1/2; a formal derivation is deferred to the end of this proof. Hence, the probability of having exactly $g-1$ free-riders in an improving step is at most 2^{-g}, which holds independently for each improvement. Under the condition of $g-1$ free-riders, we use the proposed bound of $\bigl(p^{(t)}\bigr)^g$ on the success probability for the gained block of g bits after t updates of the block. Hence, using the law of total probability and $p^{(t)} < 1$, the unconditional success probability for a block is at least

$$\sum_{g=1}^{\infty} \left(p^{(t)}\right)^g \cdot 2^{-g} \;=\; \frac{1}{1-p^{(t)}/2} - 1 \;=\; \frac{p^{(t)}}{2-p^{(t)}}$$

after t updates of the block. (Here we pessimistically assume that the number of free-riders is unbounded.) Since we have at most N^* blocks with success probabilities smaller than $1-1/n$, we arrive at the lower bound

$$\prod_{t=1}^{N^*} \frac{p^{(t)}}{2-p^{(t)}}$$

on the probability of obtaining only successes in the window. Note that this bound is independent of the window size. By Lemma 5, the product is at least $\Omega(2^{-9/(n\rho)})$, which will prove the theorem.

We still have to prove the claim on the free-rider probabilities. Consider a bit that does never contribute to the LeadingOnes-value and let τ_t denote the pheromone value on the 1-edge of this bit after t (not necessarily positive) pheromone updates. Note that τ_t itself is a random variable and that large τ_t favor increases of the τ-value during the next update and vice versa. The conditional success probability for the bit, given a pheromone value of τ_t, equals $2n\tau_t$. However, we can prove that τ_t is a martingale according to

$$E(\tau_{t+1}) \;=\; 2n\tau_t \cdot \frac{(1-\rho)\tau_t + \rho}{1-\rho+2n\rho} + (1-2n\tau_t)\cdot\frac{(1-\rho)\tau_t}{1-\rho+2n\rho} \;=\; \tau_t.$$

Due to the initialization of the 1-ANT, this means that $E(\tau_t) = \tau_0 = 1/(4n)$. Hence, for any t, the considered bit leads to a success with probability

$$\int_{1/(2n^2)}^{(n-1)/(2n^2)} 2np \cdot \mathrm{Prob}(\tau_t = p)\,\mathrm{d}p \;=\; 2n \cdot E(\tau_t) \;=\; \frac{1}{2}.$$

This proves the claim and, therefore, the theorem. □

Finally, we provide the proof of the technical estimation of Lemma 5.

Proof of Lemma 5: We first consider the simple case $\rho > 1/(2n)$. By Lemma 2,

$$p^{(t)} \;\geq\; 1 - \left(\frac{1}{1+2n\rho}\right)^t \;\geq\; 1-2^{-t}.$$

Hence, $p^{(1)}/(2-p^{(1)}) \geq 1/3$ and the remaining factors are bounded by

$$\prod_{t=2}^{N^*} \frac{1-2^{-t}}{1+2^{-t}} \;=\; \prod_{t=2}^{N^*}\left(1 - \frac{2^{-t+1}}{1+2^{-t}}\right) \;\geq\; \prod_{t=2}^{N^*}\left(1-2^{-t+1}\right).$$

Using $1-x \geq e^{-2x}$ for $x \leq 1/2$, we obtain

$$\prod_{t=2}^{N^*} \left(1-2^{-t+1}\right) \;\geq\; \prod_{t=2}^{N^*} e^{-2^{-t+2}} \;=\; e^{-\sum_{t=2}^{N^*} 2^{-t+2}} \;\geq\; e^{-2}.$$

Since $1/3 \cdot e^{-2} = \Omega\left(2^{-\frac{9}{n\rho}}\right)$ if $\rho \geq 1/(2n)$, this proves the claim for $\rho > 1/(2n)$.

For $\rho \leq 1/(2n)$, we split the product into three terms that are handled separately. Let $\alpha = \lfloor 1/(5n\rho) \rfloor$ and $\beta = \lceil (\ln 4)/(n\rho) \rceil$, then

$$\prod_{t=1}^{N^*} \frac{p^{(t)}}{2-p^{(t)}} \;=\; \prod_{t=1}^{\alpha} \frac{p^{(t)}}{2-p^{(t)}} \cdot \prod_{t=\alpha+1}^{\beta-1} \frac{p^{(t)}}{2-p^{(t)}} \cdot \prod_{t=\beta}^{N^*} \frac{p^{(t)}}{2-p^{(t)}}.$$

Observe that products may be empty, in which case they are trivially bounded by 1. For the first product, we exploit that the $p^{(t)}$ grow almost linearly as long as $p^{(t)} \leq 1/2$.

By Lemma 1, $I^{(1)}(p) = \frac{2n\rho}{1-\rho+2n\rho} \cdot (1-p) \leq 2n\rho$. Moreover, $I^{(1)}(p)$ is bounded below by $\frac{n\rho}{1-\rho+2n\rho} \geq n\rho/2$ if $p \leq 1/2$ and $\rho \leq 1/(2n)$. W.l.o.g., $n \geq 10$. Since $p^{(0)} = 1/n$ and α updates can increase p by at most $\alpha \cdot 2n\rho \leq 2/5$ (cf. the proof of Theorem 1), we have $p^{(t)} \leq 1/2$ for all $t \leq \alpha$, implying that $tn\rho/2$ is a lower bound for these $p^{(t)}$. Thus,

$$\begin{aligned}\prod_{t=1}^{\alpha} \frac{p^{(t)}}{2-p^{(t)}} \;&\geq\; \prod_{t=1}^{\alpha} \frac{p^{(t)}}{2} \;\geq\; \prod_{t=1}^{\alpha} \left(t \cdot \frac{n\rho}{4}\right) \;=\; \left(\frac{n\rho}{4}\right)^{\alpha} \alpha! \;\geq\; \left(\frac{n\rho}{4}\right)^{\alpha} \left(\frac{\alpha}{e}\right)^{\alpha} \\ &\geq\; (20 \cdot e)^{-\frac{1}{5n\rho}}.\end{aligned}$$

For all $p^{(t)}$ considered in the second product, we have $p^{(t)} \geq (\alpha+1) \cdot n\rho/2 \geq 1/10$ and thus $p^{(t)}/(2-p^{(t)}) \geq 1/19$. Hence,

$$\prod_{t=\alpha+1}^{\beta-1} \frac{p^{(t)}}{2-p^{(t)}} \geq \left(\frac{1}{19}\right)^{\beta-1-\alpha} \;=\; \Omega\left(19^{-\frac{5\ln 4-1}{5n\rho}}\right).$$

For the remaining factors, we exploit that the $p^{(t)}$ converge to 1 quite fast. By Lemma 2 along with $\rho \leq 1/(2n)$,

$$1-p^{(t)} \;\leq\; \left(\frac{1}{1+2n\rho}\right)^{t} \;=\; \left(1-\frac{2n\rho}{1+2n\rho}\right)^{t} \;\leq\; (1-n\rho)^{t}.$$

We obtain

$$\frac{p^{(t)}}{2-p^{(t)}} \;\geq\; \frac{1-(1-n\rho)^t}{1+(1-n\rho)^t} \;=\; 1-\frac{2(1-n\rho)^t}{1+(1-n\rho)^t} \;\geq\; 1-2(1-n\rho)^t \;\geq\; 1-2e^{-tn\rho}.$$

Thus, we have

$$\begin{aligned}\prod_{t=\beta}^{N^*} \frac{p^{(t)}}{2-p^{(t)}} \;&\geq\; \prod_{t=\beta}^{N^*} \left(1-2e^{-tn\rho}\right) \;\geq\; \prod_{t=\beta}^{\infty} \left(1-2e^{-n\rho\cdot\beta\cdot\lfloor t/\beta \rfloor}\right) \\ &\geq\; \prod_{t=\beta}^{\infty} \left(1-2\cdot 4^{-\lfloor t/\beta \rfloor}\right).\end{aligned}$$

Grouping sets of β equal factors each and performing an index transformation with $j = t\beta$, we arrive at

$$\prod_{t=\beta}^{N^*} \frac{p^{(t)}}{2-p^{(t)}} \;\geq\; \prod_{j=1}^{\infty} \left(1-2\cdot 4^{-j}\right)^{\beta}.$$

Using $1 - x \geq e^{-2x}$ for $x \leq 1/2$, we obtain

$$\prod_{j=1}^{\infty} e^{-4\cdot 4^{-j}\cdot\beta} \;=\; e^{-4\beta\sum_{j=1}^{\infty}4^{-j}} = e^{-\frac{4\beta}{3}} \;=\; \Omega\Big(4^{-\frac{4}{3n\rho}}\Big).$$

The claim follows since $(20\cdot e)^{-\frac{1}{5}}\cdot 19^{-\frac{5\ln 4-1}{5}}\cdot 4^{-\frac{4}{3}} > 2^{-9}$. □

The Theorems 1 and 2 show that there is a phase transition in the behavior of the 1-ANT on LeadingOnes when ρ crosses the threshold $1/(n\log n)$. Below the threshold, no efficient optimization is possible since the effect of pheromone updates is more or less irrelevant. This is similar to the behavior observed on OneMax (Neumann and Witt, 2006b), where the threshold has been identified at $1/n$. This shows that the 1-ANT is not robust w. r. t. the choice of the parameter ρ on two well-known and simple example functions. Other randomized search heuristics, e. g., the (1+1) EA, are not so sensitive to their parameter settings. Decreasing the mutation probability in the (1+1) EA from $1/n$ to, e. g., $1/n^2$ would not have such desastrous effects for simple functions.

6 Generalization to BinVal

The example function BinVal (see Section 2) is a linear function, although, in some respect an extreme example. The coefficient 2^{n-i} of the i-th bit outweights the sum of all smaller coefficients. This leads to the following relation to LeadingOnes: If LeadingOnes$(x') >$ LeadingOnes(x) for $x, x' \in \{0,1\}^n$ then also BinVal$(x') >$ BinVal(x). This allows us to treat the LeadingOnes-value of the current solution as a potential function while BinVal is optimized. It is sufficient to increase the potential at most n times to reach the optimal solution (albeit the number of different BinVal-values is 2^n). Similarly, with probability $1-2^{-\Omega(n)}$, it is necessary to increase the potential altogether by at least $n/2$ to reach the optimum since the initial LeadingOnes-value does not exceed $n/2$ with this probability.

When the (1+1) EA optimizes BinVal, the described approach allows us to immediately take over the upper bound $O(n^2)$ for the expected optimization time on LeadingOnes. This is not the best bound possible since $O(n\log n)$ can be shown by a direct approach. Such a direct approach seems difficult for the 1-ANT on BinVal. Therefore, we rather try to transfer our results from LeadingOnes to BinVal using the potential function.

With respect to the lower bound, we inspect the proof of Theorem 1. Instead of considering the real BinVal, we take the LeadingOnes-value of a bit string as a pseudo-fitness. The arguments on the pheromone values are still valid, and moreover, it is still necessary to create a solution with pseudo-fitness at least n to optimize BinVal, implying that the optimum is not found with probability $1-2^{-\Omega(n)}$ before the crucial point of time in the proof is reached. The Chernoff-bound arguments on the (pseudo-)fitness carry over, too. Moreover, for an accepted step, it is afterwards still necessary to create a search point with pseudo-fitness at least f_c. We have shown that $\rho = o(1/(n\log n))$ leads to superpolynomial runtimes also on BinVal.

Theorem 3 *With probability* $1-2^{-\Omega(\min\{n,1/(n\rho)\})}$*, the runtime of the 1-ANT on* BinVal *is* $2^{\Omega(\min\{n,1/(n\rho)\})}$.

It is slightly more difficult to adapt the proof of the upper bound on LeadingOnes. The only but essential difference is that the BinVal is influenced by the configuration of the bits after the leftmost zero. Hence, the probability of a bit being a "free-rider" does not necessarily equal 1/2; in fact we conjecture larger probabilities. Pessimistically assuming that each bit is a success with probability $1 - 1/n$, we estimate the probability of $g - 1$ free-riders, i. e., a block length of g, by $(1/n)(1 - 1/n)^{g-1}$. This decreases the estimation of the unconditional success probability for a block after t updates to

$$\sum_{g=1}^{\infty} \left(p^{(t)}\right)^g \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{g-1} \;=\; \frac{p^{(t)}}{n - (n-1) \cdot p^{(t)}} \;=:\; \tilde{p}^{(t)}.$$

Using the techniques from the proof of Lemma 5, it can be shown that $\prod_{t=1}^{N^*} \tilde{p}^{(t)} = 2^{-O((\log^2 n)/(n\rho))}$. We obtain:

Theorem 4 *The expected runtime of the 1-ANT on* BinVal *is bounded above by* $O(n^2 \cdot 2^{O((\log^2 n)/(n\rho))})$.

Note that this upper bound is polynomial only if $\rho = \Omega((\log n)/n)$. Hence, it is open whether the phase transition from superpolynomial to polynomial runtime of the 1-ANT on BinVal occurs at $\rho = \Theta(1/(n \log n))$ as in the case of LeadingOnes or at larger values, e. g., $\rho = \Theta(1/n)$ as in the case of OneMax, or even larger. In any case, our analyses suggest that the efficiency of the 1-ANT on pseudo-Boolean optimization problems is not robust w. r. t. the pheromone update mechanism and the choice of ρ.

7 Conclusions

We have investigated the pheromone update mechanism in a simple ACO algorithm. Our investigations show some general properties for the update scheme used in the 1-ANT. Based on these investigations, we have shown that there is a phase transition for $\rho = 1/(n \log n)$ from exponential to polynomial for the function LeadingOnes. Afterwards we have shown how the results obtained for LeadingOnes can be transferred to BinVal. There are several open questions. First of all, it would be desirable to determine the behavior of the 1-ANT on BinVal for the remaining values of ρ. Another open problem is to analyze the 1-ANT on a classical combinatorial optimization problem for update parameters that do not enforce the pheromone values to touch their upper or lower bounds.

References

Doerr, B., Hebbinghaus, N., and Neumann, F. (2006). Speeding up evolutionary algorithms by restricted mutation operators. In *Proc. of PPSN IX*, vol. 4193 of *LNCS*, 978–987.

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

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization.* MIT Press.

Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, **276**, 51–81.

Feller, W. (1968). *An Introduction to Probability Theory and Its Applications*, vol. 1. Wiley, 3rd ed.

Feller, W. (1971). *An Introduction to Probability Theory and Its Applications*, vol. 2. Wiley, 2nd ed.

Giel, O. and Wegener, I. (2003). Evolutionary algorithms and the maximum matching problem. In *Proc. of STACS '03*, vol. 2607 of *LNCS*, 415–426.

Gutjahr, W. J. (2003). A generalized convergence result for the graph-based ant system metaheuristic. *Probability in the Engineering and Informational Sciences*, **17**, 545–569.

Gutjahr, W. J. (2006a). First steps to the runtime complexity analysis of Ant Colony Optimization. Tech. Rep. 2006-01, Department of Statistics and Decision Support Systems, University of Vienna, Austria.

Gutjahr, W. J. (2006b). Mathematical runtime analysis of ACO algorithms: Survey on an emerging issue. Tech. Rep. 2006-08, Department of Statistics and Decision Support Systems, University of Vienna, Austria.

Merkle, D. and Middendorf, M. (2002). Modelling the dynamics of Ant Colony Optimization algorithms. *Evolutionary Computation*, **10**(3), 235–262.

Motwani, R. and Raghavan, P. (1995). *Randomized Algorithms.* Cambridge University Press.

Neumann, F. and Wegener, I. (2004). Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. In *Proc. of GECCO '04*, vol. 3102 of *LNCS*, 713–724.

Neumann, F. and Witt, C. (2006a). Ant Colony Optimization and the minimum spanning tree problem. In *Electronic Colloquium on Computational Complexity (ECCC).* Report No. 143.

Neumann, F. and Witt, C. (2006b). Runtime analysis of a simple Ant Colony Optimization algorithm. In *Proc. of ISAAC '06*, vol. 4288 of *LNCS*, 618–627. Springer.

Rudolph, G. (1997). *Convergence Properties of Evolutionary Algorithms.* Verlag Dr. Kovač.

Sudholt, D. (2005). Crossover is provably essential for the Ising model on trees. In *Proc. of GECCO '05*, 1161–1167. ACM Press.

Witt, C. (2005). Worst-case and average-case approximations by simple randomized search heuristics. In *Proc. of STACS '05*, vol. 3404 of *LNCS*, 44–56.