Rigorous Analyses for the Combination of Ant Colony Optimization and Local Search

Frank Neumann[1], Dirk Sudholt[2*], and Carsten Witt[2*]

[1] Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany,
fne@mpi-inf.mpg.de
[2] Fakultät für Informatik, LS 2, Technische Universität Dortmund, 44221 Dortmund,
Germany,
{sudholt, cw01}@ls2.cs.uni-dortmund.de

Abstract. Ant colony optimization (ACO) is a metaheuristic that produces good results for a wide range of combinatorial optimization problems. Often such successful applications use a combination of ACO and local search procedures that improve the solutions constructed by the ants. In this paper, we study this combination from a theoretical point of view and point out situations where introducing local search into an ACO algorithm enhances the optimization process significantly. On the other hand, we illustrate the drawback that such a combination might have by showing that this may prevent an ACO algorithm from obtaining optimal solutions.

1 Introduction

Ant colony optimization (ACO) is a metaheuristic that has been applied successfully to various combinatorial optimization problems. Often ACO is combined with local search methods [3, 6, 7]. Experimental investigations show that the combination of ACO with a local search procedure improves the performance significantly. On the other hand, there are examples where local search cannot help to improve the search process or even mislead the search process [1]. Therefore, it is interesting to figure out how the incorporation of local search into ACO algorithms can significantly influence the optimization process.

The goal of this paper is to investigate this effect from a theoretical point of view, using rigorous runtime analyses. The analysis of ACO algorithms with respect to their runtime behavior is a relatively new research direction. Recently, initial results with respect to the runtime behavior of variants of the MAX-MIN Ant System (MMAS) [10] have been obtained [2, 5, 8, 9].

Our aim is to point out situations where the effect of local search becomes visible in a way that can be tackled by rigorous arguments. Therefore we present functions where MMAS variants with and without local search show a strongly different runtime behavior. On one function, MMAS with local search outperforms MMAS without local search, while on a different function the effect is
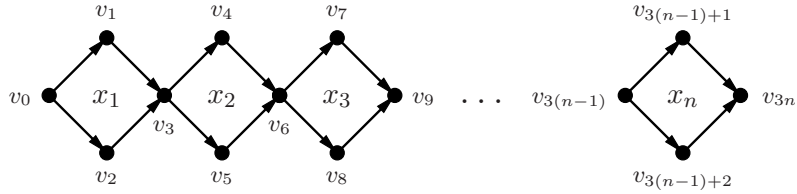
Fig. 1. Construction graph for pseudo-Boolean optimization

reversed. The differences shown in this paper are so drastic that the question of whether to use local search or not decides between polynomial and exponential runtimes.

The outline of this paper is as follows. In Section 2, we define MMAS variants with and without local search. Section 3 discusses different effects that a combination of ACO and local search can have. In Section 4, we present a rigorous analysis showing the benefits of such a combination. Contrarily, in Section 5 we investigate a different function and prove the opposite effect. We finish with some conclusions.

2 Algorithms

We consider the runtime behavior of two ACO algorithms for the optimization of pseudo-Boolean functions. Solutions for a given problem are constructed by a random walk on a so-called construction graph C according to pheromone values τ on the edges.

Algorithm 1 (Construct(C, τ))
1.) $v:=s$, mark v as visited.
2.) While there is an unvisited successor of v in C:
   a.) Let N_v be the set of unvisited successors of v and $T := \sum_{(v,w)|w \in N_v} \tau_{(v,w)}$.
   b.) Choose $w \in N_v$ with probability $\tau_{(v,w)}/T$.
   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.

We examine the construction graph given in Figure 1 known as *Chain* [4]. Constructing bit strings of length n, 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 the edge $(v_{3(i-1)}, v_{3(i-1)+1})$ (called 1-edge) is chosen, x_i is set to 1 in the constructed solution. Otherwise the corresponding 0-edge is taken, and $x_i = 0$ holds. After this decision has been made, the only available edge leads to the decision node for the next bit.

We ensure $\sum_{(u,\cdot) \in E} \tau_{(u,\cdot)} = 1$ for all decision nodes $u = v_{3i}$, $0 \leq i \leq n - 1$. Let $p_i = \text{Prob}(x_i = 1)$ be the probability of setting the bit x_i to 1 in the next constructed solution. Due to our setting, we have $p_i = \tau_{(3(i-1),3(i-1)+1)}$ and $1 - p_i = \tau_{(3(i-1),3(i-1)+2)}$, i.e., the pheromone values correspond directly to the probabilities for choosing the bits in the constructed solution. In addition,

following the MAX-MIN ant system by Stützle and Hoos [10], we restrict each $\tau_{(u,v)}$ to the interval $[1/n, 1 - 1/n]$ such that every solution always has a positive probability of being chosen.

Depending on whether edge (u, v) is contained in the path $P(x)$ of the constructed solution x, the pheromone values are updated to τ' as follows:

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

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

The following algorithm, which we call MMAS*, has been defined by Gutjahr and Sebastiani [5] under the original name MMAS$_{\text{bs}}$. Here, in each generation the best solution obtained during the run of the algorithm, called best-so-far solution, is rewarded. Another property of the model is that the best-so-far solution may not switch to another one that has the same fitness.

Algorithm 2 (MMAS*)
1.) Set $\tau_{(u,v)} = 1/2$ for all edges (u, v).
2.) Compute a solution x^* using $Construct(C, \tau)$.
3.) Update the pheromone values with respect to x^*.
4.) Compute a solution x using $Construct(C, \tau)$.
5.) If $f(x) > f(x^*)$, set $x^* := x$.
6.) Update the pheromone values with respect to x^*.
7.) Go to 4.).

We enhance the MMAS* with local search. Throughout this work, local search is a procedure that repeatedly replaces the current solution by a Hamming neighbor with strictly larger fitness until a local optimum is found. We do not specify a pivoting rule, hence we implicitly deal with a class of algorithms.

Algorithm 3 (MMAS-LS*)
1.) Set $\tau_{(u,v)} = 1/2$ for all edges (u, v).
2.) Compute a solution x using $Construct(C, \tau)$.
3.) Set $x^* := \text{LocalSearch}(x)$.
4.) Update the pheromone values with respect to x^*.
5.) Compute a solution x using $Construct(C, \tau)$.
6.) Set $z := \text{LocalSearch}(x)$.
7.) If $f(z) > f(x^*)$, set $x^* := z$.
8.) Update the pheromone values with respect to x^*.
9.) Go to 5.).

The fitness functions considered in the following only have a linear number of fitness values, hence the number of iterations in one local search call is bounded by $O(n)$. Depending on the pivoting rule, the number of fitness evaluations needed to find a better Hamming neighbor may vary; however, it is trivially bounded by n. Hence, the number of function evaluations is at most by a factor $O(n^2)$ larger than the number of generations.

We consider as performance measure the number of generations, also referred to as optimization time. This yields an advantage for MMAS-LS* w. r. t. fitness evaluations. However, the upcoming performance gaps are between polynomial and exponential values, and an advantage of order n^2 is negligible.

3   The Effect of Combining ACO and Local Search

The effect of using local search with ACO algorithms is manifold. Firstly, local search can help to find good solutions more quickly as it increases the "greedi-ness" within the algorithm. Moreover, the pivoting rule used in local search may guide the algorithm towards certain regions of the search space. For example, first ascent pays more attention to the first bits in the bit string, which may induce a search bias. However, we will not deal with this effect in our study. In particular, our functions are designed such that the pivoting rule is not essential.

There is, however, another effect that we want to investigate more closely. The pheromone values induce a sampling distribution over the search space. On a typical fitness landscape, once the best-so-far solution has reached a certain quality, sampling new solutions with a high variance becomes inefficient and the current best-so-far solution x^* is maintained for some time. Previous studies on MMAS variants [5, 8] have shown that then the pheromones quickly reach the upper and lower bounds corresponding to x^*. This means that the algorithm turns to sampling close to x^*. In other words, MMAS variants typically reach a situation where the "center of gravity" of the sampling distribution follows the current best-so-far solution and the variance of the sampling distribution is low.

When introducing local search into an MMAS algorithm, this may not be true. Local search is able to find local optima that are far away from the current best-so-far solution. In this case the "center of gravity" of the sampling distribution is far away from the best-so-far solution.

Assume there is a path of Hamming neighbors with increasing fitness leading to a local optimum. Assume further that all points close to the path have lower fitness. Then for MMAS* it is likely that the sampling distribution closely follows the path. The path of increasing fitness need not be straight. In fact, it can make large bends through the search space until a local optimum is reached. On the other hand, MMAS-LS*, when starting with the same setting, will reach the local optimum within a single iteration of local search. Then the local optimum becomes the new best-so-far solution x^* while the sampling distribution is still concentrated around the starting point. In the following generations, as long as the best-so-far solution is not exchanged, the pheromone values on all bits synchronously move towards their respective bounds in x^*. This implies for the sampling distribution that the "center of gravity" takes a (sort of) direct route towards the local optimum, irrespective of the bent path taken by local search. An illustration is given in Figure 2.

Consequences are that different parts of the search space are sampled by MMAS* and MMAS-LS*, respectively. Moreover, with MMAS* the variance in the solution construction is always quite low as the sampling distribution is
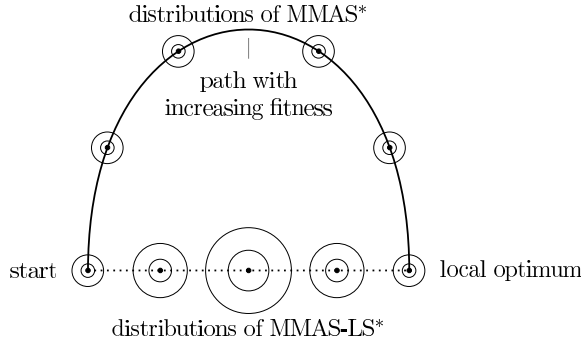
Fig. 2. A sketch of the search space showing the behavior of MMAS* and MMAS-LS*. The dots and circles indicate the sampling distributions of MMAS* and MMAS-LS*, resp., at different points of time. While the distribution of MMAS* tends to follow the fitness-increasing path from left to right, the distribution of MMAS-LS* takes a direct route towards the local optimum.

always concentrated on certain points on the path. But when the best-so-far solution with local search suddenly moves a long distance, the variance in the solution construction may be very high as the bits differing between the starting point and x^* may have pheromones close to $1/2$. This strongly resembles a uniform crossover operation well-known in evolutionary computation.

Our aim in the following is to create functions where MMAS* and MMAS-LS* have a different runtime behavior. Moreover, we want the performance difference to be drastic in order to show how deep the impact of local search can possibly be. To this end, we exploit that the sampling distributions can follow different routes through the search space. More precise, for one function we place a target region with many global optima on the straight line between starting point and local optimum and turn the local optimum into a trap that is hard to overcome. In such a setting, we expect MMAS-LS* to drastically outperform MMAS*. These ideas are made precise in Section 4. On the other hand, if the region of global optima is made a region of traps and the global optimum is very close to the local optimum, MMAS* has a clear advantage over MMAS-LS*. Another function following this idea is defined and analyzed in Section 5.

4   Benefits of Combining ACO and Local Search

We now formally define a function where local search is beneficial according to the ideas from Section 3. It is named SP-Target (short path with target). The path with increasing fitness is given by the set $SP = \{1^i 0^{n-i} \mid 0 \leq i \leq n\}$. The path ends with the local optimum 1^n. A large target area containing all global optima is specified by $OPT = \{x \mid |x|_1 \geq (3/4) \cdot n \wedge H(x, SP) \geq n/(\gamma \log n)\}$, where $H(x, SP)$ denotes the Hamming distance of x to the closest search point of SP and $\gamma \geq 1$ is a constant to be chosen later. For all remaining search points,

the function SP-Target gives hints to reach 0^n, the start of the path SP. We denote by $|x|_0$ the number of zeros in x and by $|x|_1$ the number of ones in x.

$$\text{SP-Target}(x) := \begin{cases} |x|_0 & x \notin (\text{SP} \cup \text{OPT}) \\ n + i & x = 1^i 0^{n-i} \in \text{SP} \\ 3n & x \in \text{OPT}. \end{cases}$$

The function SP-Target is sketched in Figure 3. Note that we have actually defined a class of functions dependent on γ. All following results will hold for arbitrary constant $\gamma \geq 1$ unless stated otherwise.
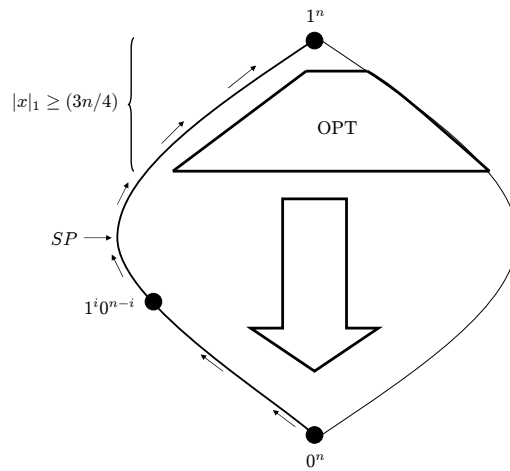

Fig. 3. Illustration of Boolean hypercube and the function SP-Target. Arrows indicate gradients of increasing fitness.

The following theorem shows that MMAS* without local search is not successful. We restrict ourselves to polynomially large $1/\rho$ here and also in the remainder of the paper as otherwise the ACO component would be too close to random search.

Theorem 1. *Choosing $\rho = 1/poly(n)$, the optimization time of MMAS* on SP-Target is $2^{\Omega(n^{2/9})}$ with probability at least $1 - 2^{\Omega(n^{2/9})}$,*

To prove the preceding theorem, we have to take into account situations where the pheromone values of MMAS* have not yet reached their bounds and the construction procedure samples with high variance. This is the case in particular after initialization. The following lemma will be used to check the probability of finding the optimum in the early steps of MMAS* on SP-Target.

Lemma 1. *If the best-so-far solution of MMAS* has never had more than $2n/3$ 1-bits, the probability of creating a solution with at least $3n/4$ 1-bits is $2^{-\Omega(n)}$ in each generation.*

Proof. The proof is an application of Chernoff bounds w. r. t. the number of ones in the solutions created by MMAS*. Let the potential $P_t := p_1 + \cdots + p_n$ at time t denote the current sum of the probabilities of sampling ones over all bits, which, by definition of the construction procedure, equals the expected number of ones in the next constructed solution. If we know that $P_t \leq 2n/3$, the probability of creating a solution with at least $3n/4$ 1-bits is $2^{-\Omega(n)}$. We now show: if all best-so-far solutions up to time t have at most $2n/3$ ones, then $P_i \leq 2n/3$ for $0 \leq i \leq t$. This will prove the lemma.

For the last claim, we denote by k the number of ones in the best-so-far solution according to which pheromones are updated. Due to the pheromone update mechanism, the new potential P_{i+1} is obtained from P_i and k according to $P_{i+1} = (1-\rho)P_i + k\rho$. Hence, if $P_i \leq 2n/3$ and $k \leq 2n/3$ then also $P_{i+1} \leq 2n/3$. The claim follows by induction since $P_0 = n/2 \leq 2n/3$. $\qquad\square$

We can now prove Theorem 1.

Proof of Theorem 1. We distinguish two different phases in the run according to the best-so-far solution x^*. Phase 1 holds as long as $x^* \notin \mathrm{SP}$ and $|x^*|_1 \leq 2n/3$, and Phase 2 applies as long as $x^* \in \mathrm{SP}$. Our aim is to show that a typical run passes through the two phases in their order with a failure probability of only $2^{-\Omega(n^{2/9})}$. The probability of finishing the second phase will be bounded by $2^{-\Omega(n^{2/9})}$ for each step of the phase, which will imply the theorem.

Consider the first (and best-so-far) solution x^* created by MMAS*. By Chernoff bounds, $n/3 \leq |x^*|_1 \leq 2n/3$ with probability $1 - 2^{-\Omega(n)}$. There is only a single solution in SP for each value of $|x^*|_1$. By the symmetry of the construction procedure, we conclude $\mathrm{Prob}(x^* \in \mathrm{SP} \mid |x^*|_1 = k) = 1/\binom{n}{k}$. The last expression is $2^{-\Omega(n)}$ for $n/3 \leq k \leq 2n/3$. Hence, with probability $1 - 2^{-\Omega(n)}$, there is a non-empty Phase 1. By Lemma 1, the probability that a specific generation in Phase 1 creates an optimum is $2^{-\Omega(n)}$. Otherwise, the behavior is as for MMAS* on the function $|x|_0$. Using $\rho = 1/poly(n)$ and the analyses for the symmetric function $|x|_1$ from [8] and [5], the expected time until the first phase is finished is polynomial. By the law of total probability and the union bound, the total failure probability in Phase 1 is bounded by the product of its expected length and the failure probability in a single generation. Therefore, the total failure probability for the first phase is still of order $2^{-\Omega(n)}$.

In Phase 2 we have $x^* \in \mathrm{SP}$. The goal is now to show that a solution from SP with high probability can only be created if the sampling distribution is sufficiently concentrated around solutions in SP. This in turn makes creating solutions of high Hamming distance from SP, including OPT, very unlikely.

We make this idea precise and consider a point $1^i 0^{n-i} \in \mathrm{SP}$. This search point consists of a prefix of i ones and a suffix of $n - i$ zeros. For a newly constructed solution x we define $P(i) := p_1 + \cdots + p_i$ as the expected number of ones in the

prefix and $S(i) := (1 - p_{i+1}) + \cdots + (1 - p_n)$ as the expected number of zeros in the suffix. The number of ones in the prefix plus the number of zeros in the suffix yields the number of bits equaling in $1^i 0^{n-i}$ and x, i.e., $n - H(1^i 0^{n-i}, x)$. We call $P(i)$ ($S(i)$) *insufficient* iff $P(i) \leq i - i^{2/3}$ ($S(i) \leq (n-i) - (n-i)^{2/3}$) holds. We now show that with insufficiencies it is very unlikely to create $1^i 0^{n-i}$. As this holds for all i, we conclude that if SP is reached after a certain number of generations, the pheromones do not have insufficiencies, with high probability.

Let $s(i)$ denote the probability of constructing the solution $1^i 0^{n-i}$. We distinguish three cases and apply Chernoff bounds to prove the following implications:

Case 1: $i < n^{2/3}$. Then insufficient $S(i)$ implies $s(i) = 2^{-\Omega(n^{1/3})}$.

Case 2: $i > n - n^{2/3}$. Then insufficient $P(i)$ implies $s(i) = 2^{-\Omega(n^{1/3})}$.

Case 3: $n^{2/3} \leq i \leq n - n^{2/3}$. Then insufficient $P(i)$ and insufficient $S(i)$ each imply $s(i) = 2^{-\Omega(n^{2/9})}$.

We assume that the described insufficiencies do not occur whenever a best-so-far solution $x^* = 1^i 0^{n-i}$ in Phase 2 is accepted. The failure probability is $2^{-\Omega(n^{2/9})}$ for each new best-so-far solution x^*. Generations in between two exchanges of x^* cannot create insufficiencies as $P(i)$ and $S(i)$ can only increase as long as x^* is maintained. Hence, we do not have insufficiencies in Phase 2 for at least $2^{\Omega(n^{2/9})}$ generations with probability at least $1 - 2^{-\Omega(n^{2/9})}$.

Being in Phase 2 without insufficiencies, we show depending on the three cases for the current $x^* = 1^i 0^{n-i}$ that creating an optimal solution has probability $2^{-\Omega(n^{2/9})}$. In the first case, the expected number of zeros in the suffix of x is at least $(n-i) - (n-i)^{2/3}$. By Chernoff bounds, the random number of zeros is at least $(n-i) - 2(n-i)^{2/3}$ with probability at least $1 - 2^{-\Omega(n^{1/3})}$. Along with $i < n^{2/3}$, it follows that then the solution has Hamming distance at most $3n^{2/3}$ from SP. By the definition of SP-Target, this is not enough to reach OPT. The second case is treated analogously. In the third case, the probability of obtaining less than $i - 2i^{2/3}$ ones in the prefix or less than $(n-i) - 2(n-i)^{2/3}$ zeros in the suffix is altogether bounded by $2^{-\Omega(n^{2/9})}$. Then the solution has Hamming distance at most $4n^{2/3}$ from SP, which is also not enough to reach the optimum. This finishes the analysis of the second phase, and, therefore, proves the theorem. $\qquad\square$

The following theorem proves the benefits of local search.

Theorem 2. *Choosing $1/poly(n) \leq \rho \leq 1/16$, the optimization time of MMAS-LS* on SP-Target is $O(1/\rho)$ with probability $1 - 2^{-\Omega(n)}$. If $\gamma \geq 1$ is chosen large enough but constant, the expected optimization time is also $O(1/\rho)$.*

Proof. The first solution x^* is either 1^n or a global optimum. In the first case all pheromone values increase simultaneously and uniformly from their initial value $1/2$ towards their upper bound $1 - 1/n$. We divide a run into two phases. The first phase ends when either all pheromones become larger than $27/32$ or when a global optimum has been found. The second phase ends when a global optimum has been found, hence it is empty if the first phase ended with an optimum.

We first bound the length of the first phase by the first point of time t^* where all pheromone values exceed $27/32$. Since the pheromone values are at least $\min\{1 - 1/n, 1 - (1/2)(1 - \rho)^t\}$ after t steps, solving the equation

$$1 - \left(\frac{1}{2}\right)(1 - \rho)^t \ = \ 27/32 \ \Leftrightarrow \ (1 - \rho)^t \ = \ 5/16$$

yields the upper bound

$$t^* \leq \left\lceil \frac{\ln(5/16)}{\ln(1 - \rho)} \right\rceil \leq \frac{\ln(16/5)}{\rho} + 1 = O(1/\rho).$$

The assumption $\rho \leq 1/16$ implies that at the last step in the first phase the pheromone value at any bit is within the interval $[25/32, 27/32]$, pessimistically assuming that a global optimum has not been found before. The new search point x then created fulfills the following two properties with probability $1 - O(2^{-n/2400})$:

1. $\frac{3n}{4} \leq |x|_1 \leq \frac{7n}{8}$,
2. $H(x, \text{SP}) \geq n/(\gamma \log n)$.

Using Chernoff bounds with $\delta := 1/25$, the failure probability for the first event is at most $2e^{-(25n/32)(\delta^2/3)} = 2e^{-n/2400}$. To bound the failure probability of the second event, given the first event, we exploit that all pheromone values are equal. Therefore, if we know that $|x|_1 = k$ then x is uniform over all search points with k ones. Since the number of search points with k ones is monotone decreasing for $3n/4 \leq k \leq 7n/8$, we only consider search points with $k = 7n/8$ ones as a worst case. The number of such search points is $\binom{n}{n/8}$, and the number of search points of Hamming distance at most $m := n/(\gamma \log n)$ from SP is at most $m \cdot \binom{n}{m}$. Altogether, the probability of $H(x, \text{SP}) \leq m$ given that $3n/4 \leq |x|_1 \leq 7n/8$ is bounded from above by

$$\frac{m\binom{n}{m}}{\binom{n}{n/8}} \ \leq \ \frac{m\left(\frac{en}{m}\right)^m}{\left(\frac{n}{n/8}\right)^{n/8}} \ \leq \ m \cdot 2^{o(n)} \cdot 8^{-n/8}.$$

The last expression is even $O(2^{-n/8})$. Altogether, the sum of the failure probabilities is $O(2^{-n/2400})$ as suggested, and the first statement follows.

For the second statement we estimate the time in the second phase, provided that the first phase has been unsuccessful. Using [8] and $\rho = 1/poly(n)$, the time to reach the pheromone bound is $O((\log n)/\rho) = poly(n)$, or an optimum is created anyway. With all pheromones at the upper bound, the solution construction process equals a standard mutation of 1^n, i.e., flipping each bit in 1^n independently with probability $1/n$. Flipping the first m bits results in a global optimum as $0^m 1^{n-m}$ has Hamming distance at least m to $1^i 0^{n-i}$ for any i. The probability of creating $0^m 1^{n-m}$ in a standard mutation is at least

$$\left(\frac{1}{n}\right)^{n/(\gamma \log n)} \left(1 - \frac{1}{n}\right)^{n - n/(\gamma \log n)} \geq e^{-1} \cdot 2^{-n/\gamma}.$$

This means that the expected time in the second phase is $O(poly(n)2^{n/\gamma})$. Using that the first phase is unsuccessful only with probability $O(2^{-n/2400})$ and applying the law of total probability, the expected optimization time altogether is $O(1/\rho) + O(2^{-n/2400}) \cdot O(poly(n)2^{n/\gamma})$. The latter is $O(1/\rho)$ for $\gamma > 2400$, which proves the second statement. $\square$

5 Drawbacks of Combining ACO and Local Search

Similarly to the function SP-Target, we design another function SP-Trap (short path with trap) where local search is detrimental, using ideas from Section 3. We take over the path with increasing fitness, $\text{SP} = \{1^i 0^{n-i} \mid 0 \leq i \leq n\}$, but in contrast to SP-Target, the former region of global optima now becomes a trap, $\text{TRAP} = \{x \mid |x|_1 \geq (3/4) \cdot n \wedge H(x, \text{SP}) \geq n/\log n\}$. The unique global optimum is placed within distance 2 from the local optimum: $\text{OPT} = \{0^2 1^{n-2}\}$. This ensures that local search climbing the path SP cannot reach the global optimum. All remaining search points give hints to reach the start of the path.

$$
\text{SP-Trap}(x) := \begin{cases} |x|_0 & x \notin (\text{SP} \cup \text{TRAP} \cup \text{OPT}) \\ n+i & x = 1^i 0^{n-i} \in \text{SP} \\ 3n & x \in \text{TRAP} \\ 4n & x \in \text{OPT}. \end{cases}
$$

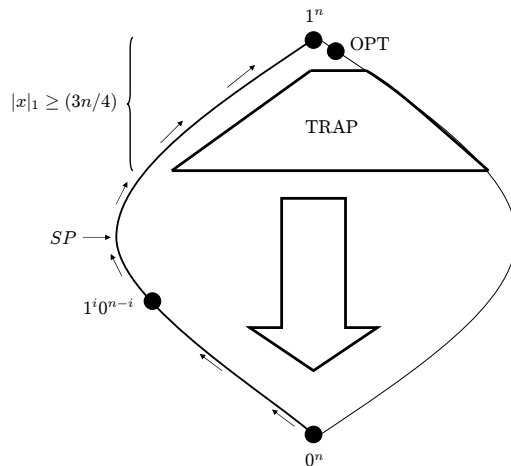The function SP-Trap is sketched in Figure 4.

Fig. 4. Illustration of Boolean hypercube and the function SP-Trap. Arrows indicate gradients of increasing fitness.

In the remainder of this section, we prove that MMAS* is efficient on SP-Trap while MMAS-LS* fails dramatically. Tuning the definition of SP-Trap, we could

also extend the following theorem by a polynomial bound on the expected optimization time. We refrain from such modifications to illustrate the main effects.

Theorem 3. *Choosing $\rho = 1/poly(n)$, the optimization time of MMAS* on SP-Trap is $O((n \log n)/\rho + n^3)$ with probability $1 - 2^{-\Omega(n^{2/9})}$.*

Proof. By the argumentation from Theorem 1, the probability that a solution in TRAP is produced is at most $2^{-\Omega(n^{2/9})}$. This still holds when taking the union bound over $O(n^3)$ generations.

Under the assumption that TRAP is never reached until the global optimum is found, MMAS* behaves equally on SP-Trap and a modified function where $x \in$ TRAP receives fitness $|x|_0$. We apply fitness-level arguments from [5, 8] to estimate the expected optimization time on the latter, easier function. The number of fitness levels is $O(n)$. On every fitness level, the number of generations until either all pheromones are frozen or the current best-so-far solution has improved is bounded by $O((\log n)/\rho)$ with probability 1 [8]. We pessimistically assume that an improvement can only happen once all pheromones have been frozen. Then the optimization time is bounded by $O((n \log n)/\rho)$ plus the sum of waiting times for improvements on all fitness levels. Showing that the latter quantity is bounded by $O(n^3)$ with probability $1 - 2^{-\Omega(n)}$ completes the proof.

After freezing, the solution construction process equals a standard mutation of the best-so-far solution x^*. The probability for an improvement from $x^* = 1^n$ is at least $1/(en^2)$. For all other $x^* \notin$ TRAP, there is always a better Hamming neighbor, hence the probability for an improvement is at least $1/(en)$. Together, the expected waiting times for improvements on all fitness levels sum up to $en^2 + O(n) \cdot en = O(n^2)$. By Markov's inequality the probability of waiting more than cn^2 steps is at most $1/2$ for a suitable constant $c > 0$. Hence, the probability that more than n independent phases of length cn^2 are needed is bounded by $2^{-\Omega(n)}$. Therefore, the bound $O(n^3)$ holds with probability $1 - 2^{-\Omega(n)}$. □

Theorem 4. *Choosing $1/poly(n) \le \rho \le 1/16$, the optimization time of MMAS-LS* on SP-Trap is $2^{\Omega(n)}$ with probability $1 - 2^{-\Omega(n)}$.*

Proof. We follow the lines of proof for Theorem 2. As long as OPT $= 0^2 1^{n-2}$ is not created, the behavior of MMAS-LS* on SP-Trap and SP-Target is identical. Reconsider the first phase described in the proof of Theorem 2 (with the former OPT replaced by TRAP) and denote by $P := p_1 + \cdots + p_n$ the sum of probabilities of sampling ones over all bits. Throughout the phase, $P \le 27n/32$, hence the probability of sampling at least $n - 2$ ones, which is necessary to reach OPT, is $2^{-\Omega(n)}$ according to Chernoff bounds.

With probability $1 - 2^{-\Omega(n)}$, the first best-so-far solution 1^n is replaced by some $x^{**} \in$ TRAP where $|x^{**}|_1 \le 7n/8$ when the first phase is ended. Due to strict selection, x^{**} then can only be replaced if OPT is created. The latter has probability $2^{-\Omega(n)}$ for the following reasons: the P-value is at most $27n/32 \le 7n/8$ when x^{**} is accepted. Hence, following the argumentation from the proof of Lemma 1, the P-value will not exceed $7n/8$ unless x^{**} is replaced. With a P-value of at most $7n/8$, creating OPT has probability $2^{-\Omega(n)}$. □

6 Conclusions

We have investigated the combination of ACO and local search from a theoretical point of view and pointed out how this combination can influence the search process. In particular, we have rigorously shown that the combination of both methods can outperform ACO algorithms not using local search procedures. Furthermore, we have proven that the combination of ACO and local search may mislead the search process. Our results are a further step in the runtime analysis of ACO and its hybridizations.

Acknowledgement

References

1. P. Balaprakash, M. Birattari, T. Stützle, and M. Dorigo. Incremental local search in ant colony optimization: Why it fails for the quadratic assignment problem. In *Proc. of ANTS Workshop '06*, pages 156–166, 2006.
2. B. Doerr, F. Neumann, D. Sudholt, and C. Witt. On the runtime analysis of the 1-ANT ACO algorithm. In *Proc. of GECCO '07*, pages 33–40. ACM, 2007.
3. M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.
4. W. J. Gutjahr. First steps to the runtime complexity analysis of ant colony optimization. *Computers and Operations Research*, 35(9):2711–2727, 2008.
5. W. J. Gutjahr and G. Sebastiani. Runtime analysis of ant colony optimization with best-so-far reinforcement. *Methodology and Computing in Applied Probability*, 2008. (to appear).
6. H. H. Hoos and T. Stützle. *Stochastic Local Search: Foundations & Applications*. Elsevier/Morgan Kaufmann, 2004.
7. J. Levine and F. Ducatelle. Ant colony optimisation and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society*, 2004.
8. F. Neumann, D. Sudholt, and C. Witt. Comparing variants of MMAS ACO algorithms on pseudo-boolean functions. In *Proc. of SLS '07*, volume 4638 of *LNCS*, pages 61–75. Springer, 2007.
9. F. Neumann and C. Witt. Runtime analysis of a simple ant colony optimization algorithm. In *Proc. of ISAAC '06*, volume 4288 of *LNCS*, pages 618–627. Springer, 2006.
10. T. Stützle and H. H. Hoos. MAX-MIN ant system. *Journal of Future Generation Computer Systems*, 16:889–914, 2000.