UNIVERSITÄT DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

SONDERFORSCHUNGSBEREICH 531

Design und Management komplexer technischer Prozesse und Systeme mit Methoden der Computational Intelligence

Local Search in Memetic Algorithms: the Impact of the Local Search Frequency

Dirk Sudholt

Nr. CI-208/06

Interner Bericht ISSN 1433-3325 June 2006

Sekretariat des SFB 531 · Universität Dortmund · Fachbereich Informatik/XI
44221 Dortmund · Germany

Diese Arbeit ist im Sonderforschungsbereich 531, „Computational Intelligence“, der Universität Dortmund entstanden und wurde auf seine Veranlassung unter Verwendung der ihm von der Deutschen Forschungsgemeinschaft zur Verfügung gestellten Mittel gedruckt.

Local Search in Memetic Algorithms: the Impact of the Local Search Frequency

Dirk Sudholt*

FB Informatik, Univ. Dortmund, 44221 Dortmund, Germany
Dirk.Sudholt@udo.edu

Abstract

Memetic algorithms are popular randomized search heuristics combining evolutionary algorithms and local search. Their efficiency has been demonstrated in countless applications covering a wide area of practical problems. However, theory of memetic algorithms is still in its infancy and there is a strong need for a rigorous theoretical foundation to better understand these heuristics. Here, we attack one of the fundamental issues in the design of memetic algorithms from a theoretical perspective, namely the choice of the frequency with which local search is applied. Since no guidelines are known for the choice of this parameter, we care about its impact on memetic algorithm performance. We present worst-case problems where the choice of the local search frequency has an enormous impact on the performance of a simple memetic algorithm. A rigorous theoretical analysis shows that on these problems, with overwhelming probability, even a small factor of 2 decides about polynomial versus exponential optimization times.

1 Introduction

Solving optimization problems is a fundamental task in computer science. Theoretical computer science has developed powerful techniques to design problem-specific algorithms and to provide guarantees on the worst-case runtime and the quality of solutions. Nevertheless, these algorithms can be quite complicated and difficult to implement. Moreover, practitioners often have to deal with problems where they have only limited insight into the structure of the problem, thus making it impossible to design specific algorithms.

The advantage of randomized search heuristics like randomized local search, tabu search, simulated annealing, and evolutionary algorithms is that they are easy to design and easy to implement. Despite the lack of performance guarantees, they often yield good results in short time and they can be applied in scenarios where the optimization problem at hand is only known as a black box.

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

Therefore, practitioners often apply randomized search heuristics like, e. g., evolutionary algorithms to find good solutions during a random search process. Often the performance of evolutionary algorithms can be enhanced if (problem-specific) local search techniques are integrated into the search process. These hybrid algorithms are known as memetic algorithms. Using problem-specific local search can provide a better guidance for the random search process while preserving the low costs of implementation. That way, the advantages of problem-specific algorithms and simple randomized search heuristics can be combined. It is therefore not surprising that practitioners have applied memetic algorithms to a wide range of applications, see Moscato [8] for a survey or Hart, Krasnogor, and Smith [4] for various applications.

However, from a theoretical point of view this situation is unsatisfactory because these algorithms are presently not considered in the theory of algorithms. Despite a broad activity in the area of memetic algorithms, theory on memetic algorithms is hanging behind and rigorous theoretical results are rare.

We present a brief survey of theoretical approaches concerning memetic algorithms. Hart [3] empirically investigates the role of the local search frequency and the local search depth, i. e., the maximal number of iterations in one local search call, on three artificial test functions. Goldberg and Voessner [2] present a macro-level result on the design of global-local search hybrids explaining how to balance global and local search. Lourenço, Martin, and Stützle [6] empirically analyze the runtime distribution of memetic algorithms on problems from combinatorial optimization. Merz [7] adapts the parameterization of memetic algorithms to the given problem by using problem-specific knowledge gained from empirical analysis of the problem structure. This approach is extended by Watson, Howe, and Whitley [12] who additionally consider the dynamic behavior of the algorithm. Krasnogor [5] presents a simple PLS-completeness proof for a class of memetic algorithms and establishes a connection between local search complexity theory (see Papadimitriou, Schäffer, and Yannakakis [9]) and Kolmogorov complexity theory. Finally, Sudholt [11] compares a simple memetic algorithm with two well-known randomized search heuristics and proves rigorously for an artificial function that the local search depth has a large impact on the behavior of the algorithm.

Although memetic algorithms are easy to implement, they are not designed to support an analysis. In fact, the rigorous analysis of these algorithms can be quite challenging. To the best of our knowledge, [11] and this work are the first rigorous analyses of memetic algorithms in terms of computational complexity.

In the design of memetic algorithms it is essential to find a proper balance between evolutionary (global) search and local search. If the effect of local search is too weak, we fall back to standard evolutionary algorithms. If the effect of local search is too strong, the algorithm suffers from the disadvantages of pure local search as it may get stuck in local optima of bad quality. Moreover, the algorithms is likely to rediscover the same local optimum over and over again, wasting computational effort. Lastly, when dealing with population-based algorithms, too much local search quickly leads to a loss of diversity within the population.

A common design strategy is to apply local search with a fixed frequency, say every τ generations for some $\tau \in \mathbb{N}$. At present, there are no guidelines available for the choice of this parameter. Hence, before trying to establish design guidelines, an interesting question is what impact the local search frequency has on the performance of the algorithm. We will define a simple memetic algorithm and ask whether the algorithm is robust to the choice of the local search frequency and what can happen in the worst case (w. r. t. the problem instance) if we choose a wrong parameterization.

We will prove that in the worst case even small changes to the local search frequency can totally change the algorithm's behavior and decide about polynomial versus exponential optimization times, with overwhelming probability.

The investigation of worst-case problems leads us to a class of artificially constructed functions that are far away from real-world problems. The long-term goal is a powerful theory with implications for practical problems. However, in order to obtain results on practical problems, the first step is to develop a rigorous theoretical foundation and the methodology to analyze memetic algorithms. Thus, besides the main results, this work is interesting from a methodological point of view. By presenting a rigorous theoretical analysis, we show that memetic algorithms can be analyzed rigorously and we present the methodology to do so. Using insights gained by the analysis, we can hope to extend the analysis of memetic algorithms to a wider range of problems, including practical problems, in the near future.

The paper is structured as follows. In Section 2 we define a simple memetic algorithm, the (1+1) Memetic Algorithm. Then in Section 3 we define so-called race functions where local search effects compete with global search effects. Section 4 proves rigorously that the choice of the local search frequency has a large impact on the (1+1) MA on race functions and that even a factor of 2 makes an enormous difference. Finally, we finish with some conclusions in Section 5.

2 Definitions

The (1+1) Memetic Algorithm ((1+1) MA) is a simple memetic algorithm with population size 1 that has already been investigated in [11]. It employs the following local search procedure that stops after a predefined maximal number of $\delta = \mathit{poly}(n)$ iterations. The algorithm is defined for maximization of pseudo-boolean functions $f\colon \{0,1\}^n \to \mathbb{R}$ including problems from combinatorial optimization. $\mathrm{H}(x, y)$ denotes the Hamming distance between x and y.

Procedure 1 (Local Search(y) with depth δ)**.**

$t := 1$.
While $t \leq \delta$ and $\exists z\colon (H(z, y) = 1$ and $f(z) > f(y))$ {
$y := z$.
$t := t + 1$.
}
Return y.

If there is more than one Hamming neighbor with larger fitness, z may be chosen arbitrarily among them as this choice is immaterial to the results presented hereinafter.

Algorithm 1 ((1+1) Memetic Algorithm ((1+1) MA))**.**

1. ***Initialization***
   gen := 0. *Choose* x *uniformly at random.* $x :=$ *Local Search*(x).
2. ***Mutation***
   $y := x$. *Flip every bit in* y *independently with probability* $1/n$.
3. ***Local Search***
   If gen mod $\tau = 0$ *then* $y :=$ *Local Search*(y).
4. ***Selection***
   If $f(y) \geq f(x)$ *then* $x := y$.
5. ***Loop***
   gen := gen +1. *Continue at line 2.*

We do not specify a termination condition as we are only interested in the number of f-evaluations until a global optimum is found. Note that, in the worst case, an iteration of local search requires n f-evaluations.

Definition 1. *An event* E *occurs with overwhelming probability (w. o. p.) if* $Prob(E) = 1 - 2^{-\Omega(n^{\varepsilon})}$ *for a constant* $\varepsilon > 0$*,* n *the search space dimension.*

We say that an algorithm $\mathcal{A}$ *is* efficient *on a function* f *iff* $\mathcal{A}$ *finds a global optimum on* f *in a polynomial number of* f*-evaluations w. o. p.*

We say that an algorithm $\mathcal{A}$ fails *on a function* f *iff* $\mathcal{A}$ *does not find a global optimum in an exponential number of* f*-evaluations w. o. p.*

When constructing the race functions, we will make use of so-called long K-paths. A long K-path is a sequence of Hamming neighbors where all points are different. The following definition is taken from [1].

Definition 2. *Let* $K, N \in \mathbb{N}$ *with* $(N-1)/K \in \mathbb{N}$*. The long* K*-path of dimension* N *is a sequence of bit strings from* $\{0,1\}^N$ *defined recursively as follows. The long* K*-path of dimension 1 is defined as* $P_1^K := (0,1)$*. Let* $P_{N-K}^K = (v_1, \ldots, v_\ell)$ *be the long* K*-path of dimension* $N-K$*. Then the long* K*-path of dimension* N *is defined by prepending* K *bits to the search points from* $\{v_1, \ldots, v_\ell\}$*: let* $S_0 := (0^K v_1, 0^K v_2, \ldots, 0^K v_\ell)$*,* $S_1 := (1^K v_\ell, 1^K v_{\ell-1}, \ldots, 1^K v_1)$*, and* $B := (0^{K-1}1 v_\ell, 0^{K-2}1^2 v_\ell, \ldots, 01^{K-1} v_\ell)$*. The search points in* S_0 *and* S_1 *differ in the* K *leading bits and the search points in* B *represent a bridge between them. The long* K*-path of dimension* N *is constructed by concatenating* S_0, B*, and* S_1*.*

If $N := K^2+1$, the length of the path is $\Omega(2^K)$. Moreover, for all $0 < i < K$ the following statement holds. Let x be a point on the long path. If x has at least i successors on the path, then the ith successor has Hamming distance i of x and all other successors of x on the path have Hamming distances different from i (a proof is given in [1]). This implies that all successors on the path except the K next ones have Hamming distance at least K to x.

The index of some search point z on a long K-path will be denoted by $i(z)$. If z is not on the path, $i(z)$ is defined as $i(z) := -1$.

3 Race Functions: Where Local Search and Global Search Compete

Now we will define the aforementioned race functions where local search effects compete with global search effects. The idea behind the construction is quite intuitive. We will identify two non-overlapping blocks of the bit string of length N, referred to as x' and x'' if x is the current bit string. These partial bit strings span subspaces of the original search space. Then, subfunctions are defined on these two subspaces such that the value of the original superior function is the (weighted) sum of the subfunctions' values for an important part of the search space.

The two subfunctions are defined as follows. The function on the left block x' is based on a coherent subpath of a long K-path of adjustable length. The fitness is increasing on the path, thus it can be optimized efficiently by local search. The function on the right block x'' consists of a much shorter subpath, but only every third search point on the path has positive fitness. Hence, this subfunction contains a sequence of isolated peaks with increasing fitness and mutation can help to jump to the next peak by mutations flipping three specific bits.

To conclude, the function on the left block (or shortly, the left path) can be optimized efficiently by local search and the right path can only be optimized by mutations. The (1+1) MA on the superior function now optimizes the two subfunctions in parallel. If the local search frequency is high, we expect the algorithm to optimize the left path prior to the right path. Contrarily, if the local search frequency is low, then we expect the right path to be optimized prior to the left one.

By defining special fitness values for cases where some path end is reached we obtain a function where it makes a large difference which path is optimized first. For example, we can define the end of the left path of being globally optimal. However, if the right path is optimized prior to the left one, the function turns into a so-called deceptive function giving hints to move away from all global optima and to get stuck in a local optimum. That way, the (1+1) MA typically optimizes this function efficiently if the local search frequency is high and it gets stuck in a local optima if the local search frequency is low. Another function can be defined analogously where it is globally optimal to reach the end of the right path.

We now present our main theorem that will be proved in Section 4 according to the ideas described above.

Theorem 1 (Main Theorem). *Let $\delta, \tau \in \mathbb{N}$ be defined such that $\delta = poly(n)$, $\delta \geq 22$, $\delta/\tau \geq 2/n$, $\tau = \omega(n^{2/3})$, and $\tau = O(n^3)$ hold. There exist functions* $\text{Race}_{\ell,\text{r}}^{\text{left}}, \text{Race}_{\ell,\text{r}}^{\text{right}} \colon \{0,1\}^n \to \mathbb{R}$ *such that*

- *the (1+1) MA with local search frequency $1/\tau$ is efficient on $\text{Race}_{\ell,\text{r}}^{\text{left}}$ while the (1+1) MA with local search frequency $1/(2\tau)$ fails on $\text{Race}_{\ell,\text{r}}^{\text{left}}$ and*
- *the (1+1) MA with local search frequency $1/\tau$ fails on $\text{Race}_{\ell,\text{r}}^{\text{right}}$ while the (1+1) MA with local search frequency $1/(2\tau)$ is efficient on $\text{Race}_{\ell,\text{r}}^{\text{right}}$.*

Definition 3. *Let $n = 4N$ and $N = K^2 + 1$ with $K/3 \in \mathbb{N}$. Let P_i be the ith point on the long K-path of dimension N.*

The (1+1) MA initializes uniformly at random. However, we want the optimization of the two paths to start with specific starting points. Therefore, we use a construction that is explained in detail in Section 3 of [11] (here, we use a slight transformation of the search space which is immaterial to the algorithm). In a nutshell, we append additional $2N$ bits denoted by x''' to the $2N$ bits used by the two blocks x' and x''. The following subfunction ZZO guides the algorithm to reach $x''' = 0^{2N}$ (i. e. a concatenation of $2N$ zeros) and then to reach specific starting points for x' and x'', namely $x'x'' = 0^N P_{n^5-1}$ ($n^5 - 1$ is the multiple of 3 closest to n^5 due to the choice of K and n). Afterwards, x''' is turned into $x''' = 1^{2N}$. Once all these bits are ones, the optimization of the two paths begins.

Definition 4. *Let $x = x'x''x'''$ with $x', x'' \in \{0,1\}^N$ and $x''' \in \{0,1\}^{2N}$. We define $\text{ZZO}\colon \{0,1\}^n \to \mathbb{R}$ as*

$$\text{ZZO}(x) := \begin{cases} -H\big(x''', 0^{2N}\big) - 4N & \text{if } x'x'' \neq 0^N P_{n^5-1}, x''' \neq 0^{2N}, \\ -H\big(x'x'', 0^N P_{n^5-1}\big) - 2N & \text{if } x'x'' \neq 0^N P_{n^5-1}, x''' = 0^{2N}, \\ -H\big(x''', 1^{2N}\big) & \text{if } x'x'' = 0^N P_{n^5-1}. \end{cases}$$

Definition 5. *Call a search point $x = x'x''x'''$* well-formed *iff $i(x') \geq 0$, $i(x'') \geq 0$, $i(x'')/3 \in \mathbb{N}$, and $x''' = 1^{2N}$. Given $\ell, r \in \mathbb{N}$ we define*

$$\text{Race}^{\text{left}}_{\ell,\text{r}}(x) := \begin{cases} \text{ZZO}(x) & \text{if } H\big(x''', 1^{2N}\big) \geq 3, \\ i(x') \cdot n + i(x'') & \text{if } x \text{ well-formed}, i(x') < \ell, i(x'') < r, \\ 2^N + H(x', P_\ell) & \text{if } x \text{ well-formed}, i(x') < \ell, i(x'') = r, \\ 2^{2N} & \text{if } x \text{ well-formed}, i(x') = \ell, \\ -\infty & \text{otherwise.} \end{cases}$$

$$\text{Race}^{\text{right}}_{\ell,\text{r}}(x) := \begin{cases} \text{ZZO}(x) & \text{if } H\big(x''', 1^{2N}\big) \geq 3, \\ i(x') \cdot n + i(x'') & \text{if } x \text{ well-formed}, i(x') < \ell, i(x'') < r, \\ 2^N + H(x'', P_r) & \text{if } x \text{ well-formed}, i(x') = \ell, i(x'') < r, \\ 2^{2N} & \text{if } x \text{ well-formed}, i(x'') = r, \\ -\infty & \text{otherwise.} \end{cases}$$

In a typical run, after random initialization the function ZZO is optimized guiding the search towards the well-formed search point $x = x'x''x'''$ with $i(x') = 0$, $i(x'') = P_{n^5-1}$, and $x''' = 1^{2N}$. There is a gap between the ZZO-dependent search points and all well-formed search points since all search points with one or two zero-bits in the x'''-part have fitness $-\infty$. However, this gap can easily be jumped over by mutation in expected time $O(n^3)$. Moreover, the probability that at least $K = \Theta(n^{1/2})$ bits flip in this jump is exponentially small. Thus, it is very likely that we reach search points close to the desired starting points in polynomial time. For a proof of a result similar to the following corollary, we refer the reader to [11].

Corollary 1. *With overwhelming probability, the (1+1) MA on either* $\mathrm{Race}^{\mathrm{left}}_{\ell,\mathrm{r}}$ *or* $\mathrm{Race}^{\mathrm{right}}_{\ell,\mathrm{r}}$ *reaches some well-formed search point* x^* *with* $i(x^{*\prime}) < K$ *and* $|i(x^{*\prime\prime}) - (n^5 - 1)| < K$ *within the first* n^4 *generations.*

4 Analyzing the Impact of the Local Search Frequency

To prove our main theorem, we will investigate the progress of the algorithm on the two paths. The progress will be estimated by separating the effects of different operations and proving bounds for the cumulated progress for single types of operations.

For the rest of the section, we consider the (1+1) MA on $\mathrm{Race}^{\mathrm{left}}_{\ell,\mathrm{r}}$ or $\mathrm{Race}^{\mathrm{right}}_{\ell,\mathrm{r}}$ after some well-formed search point has been reached. In a generation with local search, the mutation only affects the algorithm if the outcome of local search is accepted in the selection step. Thus, we only have to take into account those mutations where the outcome of the following local search call is accepted.

Lemma 1. *Let* $x = x'x''x'''$ *be the current population,* x *well-formed, let* $y = y'y''y'''$ *be an offspring created by mutation, and let* $z = z'z''z'''$ *be the result of local search applied to* y*. Then* z *is accepted in the selection step only if* y *has Hamming distance at most 1 to a well-formed search point.*

Proof. Let $w = w'w''1^{2N}$ be a well-formed search point with minimal Hamming distance to y. We distinguish three cases according to $\mathrm{H}(y''', x''')$, i. e., the number of zero-bits in y'''.

- If $\mathrm{H}(y''', x''') \geq 2$, the function to be optimized during the local search process is ZZO since the fitness of all search points with one or two zero-bits in the x'''-part is $-\infty$ and the fitness is $\mathrm{ZZO}(\cdot) > -\infty$ in case of three or more zero-bits. However, due to the gap between ZZO-dependent search points and well-formed search points, local search cannot reach a well-formed search point. Hence, the offspring z is rejected in the selection step.
- In case $y''' = 1^{2N}$ and $\mathrm{H}(w'w'', y'y'') \geq 2$ we have fitness $-\infty$ for y and all Hamming neighbors of y. Hence, local search stops immediately in this case.
- Lastly, if $\mathrm{H}(y''', x''') = 1$ and $\mathrm{H}(w'w'', y'y'') \geq 1$ we have fitness $-\infty$ and the fitness cannot be increased by flipping single bits in $y'y''$. The Hamming neighbor obtained by flipping the unique zero-bit in y''' has fitness $-\infty$ and so do all Hamming neighbors with a larger number of zero-bits in the x''' part. Thus, local search stops immediately, here.

□

An important observation is that mutations followed by local search are in some sense more powerful than mutations without local search. It is possible that mutation yields a non-well-formed search point with Hamming distance 1 to a well-formed one. Then local search will reach the well-formed search point within its first iteration and the outcome of local search may be accepted by the algorithm (note that Lemma 1 provides a necessary condition, not a sufficient one). Hence, mutations followed by local search are more likely to yield an accepted search point than mutations without local search and the first iteration of local search plays a crucial role, here. As a consequence, we may in some situations regard the first iteration of local search as being part of the mutation instead of local search.

Definition 6. *An* extended mutation *is either a mutation reaching a well-formed search point or a mutation followed by one iteration of local search in case the mutant is not well-formed.*

Using these insights, we now formally define the intuitive notion of progress. In a generation without local search, the progress by one mutation on, say, the left path is defined as $i(y')-i(x')$ if y is accepted and 0 otherwise. In a generation with local search let x be the current search point, y be the individual obtained by an extended mutation, and z be the result of local search applied to y. Then the progress by one extended mutation is defined as $i(y') - i(x')$ if z is accepted and 0 otherwise and the progress by local search is $i(z') - i(y')$ if z is accepted and 0 otherwise. The progress on the right path is defined analogously.

In the following lemmas, we will prove lower and upper bounds on the cumulated progress for specific operations, namely mutations in generations without local search, extended mutations, and the remaining iterations of local search after extended mutations.

In all proofs we consider a typical run of the algorithm. Events preventing a run from being typical are called errors and the total error probability is bounded by the sum of single error probabilities. If there is only a polynomial number of exponentially small single error probabilities, a run is typical with overwhelming probability.

First, we consider the progress by mutations in generations without local search. We will bound the progress by a Chernoff-Hoeffding-type bound due to McDiarmid, see Scheideler [10] (Theorem 3.44). By adding a hypothesis on the variance, we obtain slightly simplified terms.

Lemma 2. *Let $X_1, \dots, X_m$ be independent random variables and $S_m := X_1 + \cdots + X_m$. If $X_i \le E(X_i) + b$ for all $1 \le i \le n$ and $V(S_m) = O(bd)$ for some $d \ge 0$,*

$$Prob(S_m \ge E(S_m) + d) = 2^{-\Omega(d/b)}.$$

Proof. Let $\nu = \mathrm{V}(S_m)$ and $\delta := b \cdot d/\nu$. Due to Scheideler [10] (Theorem 3.44)

$$\mathrm{Prob}(S_m \ge \mathrm{E}(S_m) + d) \le e^{-\frac{d^2}{2\nu(1+\delta/3)}}.$$

Since $\nu = O(bd)$, the e-term is $2^{-\Omega(d/b)}$. □

Lemma 3. *Let $\Delta_{\text{mut}}^{\text{left}}$ ($\Delta_{\text{mut}}^{\text{right}}$) be the progress on the left (right) path in $T = \Omega(n^4), T = poly(n)$ mutations. Then with probability $1 - 2^{-\Omega(n^{1/2})}$ for $\varepsilon > 0$*

$$(1-\varepsilon) \cdot \frac{T}{en} < \Delta_{\text{mut}}^{\text{left}} < (1+\varepsilon) \cdot \frac{T}{en}$$

and

$$(1-\varepsilon) \cdot \frac{T}{en^3} < \Delta_{\text{mut}}^{\text{right}} < (1+\varepsilon) \cdot \frac{T}{en^3}.$$

Proof. In a typical mutation, less than K bits flip simultaneously as the complementary event has error probability $2^{-\Omega(K \log K)}$. The probability that this happens at least once in T steps is still of order $2^{-\Omega(K \log K)}$.

Let $x = x'x''1^{2N}$ be the current search point, x well-formed. Let $y = y'y''y'''$ be a mutant of x. Apart from the special situations when P_ℓ or P_r is found on the left or right path, resp., the following holds. The mutant y is accepted if and only if y is well-formed and either $i(y') > i(x')$ or $i(y') = i(x')$ and $i(y'') \geq i(x'')$.

We first prove an upper bound on $\Delta_{\text{mut}}^{\text{left}}$. Let x be the current well-formed search point. For $1 \leq j \leq K$, let b_j be bit differing between $P_{i(x')+j-1}$ and $P_{i(x')+j}$. Let $B = \{b_1, \dots, b_K\}$. A mutation step creating y from x is called *relevant* iff all bits outside of B do not flip, $i(y'') \geq 0$ and $y''' = 1^{2N}$. The probability of $i(y'') \geq 0$ is $(1 + o(1)) \cdot \left(1 - \frac{1}{n}\right)^N$ since the probability of $y'' = x''$ is $\left(1 - \frac{1}{n}\right)^N$ and the probabilities to reach some other search point on the right path decrease exponentially with the Hamming distance. Hence, the probability of a relevant step is

$$\begin{aligned}
(1+o(1)) \cdot \left(1 - \frac{1}{n}\right)^{N-K+N+2N} &= (1+o(1)) \cdot \left(1 - \frac{1}{n}\right)^{n-K} \\
&\leq (1+o(1)) \cdot \left(1 - \frac{1}{n}\right)^{n} \cdot \left(1 - \frac{1}{n}\right)^{-n^{1/2}} \\
&= (1+o(1)) \cdot e^{-1}
\end{aligned}$$

By Chernoff bounds, the probability to have more than $(1 + c) \cdot T/e$ relevant mutation steps in T mutations is $2^{-\Omega(T)} = 2^{-\Omega(n^4)}$ for some constant $c > 0$ that will be chosen later.

Non-relevant steps cannot lead to a progress by mutation if less than K bits flip in one step. We estimate the progress in relevant steps as follows. The mutation operator decides independently for each bit whether it flips or not. Suppose the mutation operator makes these decisions sequentially and w. l. o. g. the bits in B are processed in order $b_1, \dots, b_K$. Then the progress in one relevant step is bounded by the number of flipping bits before the first non-flipping bit occurs. Thus, the random variable Z describing the progress in this step is due to a geometric distribution with parameter $p := (1 - 1/n)$ with the exception that Z is bounded above by K. The variance of a geometric distributed variable with parameter $p = (1 - 1/n)$ is $(1 - p)/p^2 = (1 + o(1)) \cdot 1/n$. Since the upper bound K can only decrease the variance, $\mathrm{V}(Z) = (1 + o(1)) \cdot 1/n$.

Let $Z_1, \dots, Z_m$ be random variables describing the progress in m relevant mutation steps. Due to the definition of a relevant step, $Z_i \leq K$ implying

$Z_i \leq \mathrm{E}(Z_i)+K$. Let $S := Z_1+\cdots+Z_m$, then $\mathrm{E}(S) \leq m(1-p)/p = (1+o(1))\cdot m/n$ and $\mathrm{V}(S) = \mathrm{V}(Z_1) + \cdots + \mathrm{V}(Z_m) = (1 + o(1)) \cdot m/n$. Lemma 2 yields

$$\mathrm{Prob}(S \geq \mathrm{E}(S) + cm/n) = 2^{-\Omega((cm/n)/K)} = 2^{-\Omega(m\cdot n^{-3/2})}.$$

Setting $m = (1 + c) \cdot T/e$, this bound is $2^{-\Omega(T\cdot n^{-3/2})} = 2^{-\Omega(n^{5/2})}$. Taking into account the error probability $2^{-\Omega(K \log K)} = 2^{-\Omega(n^{1/2} \log n)}$ for not flipping K or more bits at once, we have shown: the probability to have progress at least $(1 + c + o(1)) \cdot (1 + c) \cdot T/(en)$ is $2^{-\Omega(n^{1/2})}$. Choosing c such that $(1 + c + o(1)) \cdot (1 + c) \leq (1 + \varepsilon)$ completes the proof of the upper bound.

The progress on the right path can be bounded from above in the same fashion. The random variable describing the progress in one step is estimated by a geometric distribution with parameter $p := (1 - 1/n^3)$ implying $\mathrm{E}(S) \leq (1 + o(1)) \cdot m/n^3$ and $\mathrm{V}(S) = (1 + o(1)) \cdot m/n^3$. Lemma 2 yields

$$\mathrm{Prob}\big(S \geq (1 + c + o(1)) \cdot m/n^3\big) = 2^{-\Omega(m\cdot n^{-7/2})}.$$

Setting $m = (1 + c) \cdot T/e$, the claim follows since $T = \Omega(n^4)$.

Now we prove a lower bound on $\Delta^{\mathrm{left}}_{\mathrm{mut}}$. A relevant mutation step increasing the position on the path by at least 1 is called a *progressing relevant step*. The probability of a progressing relevant step is at least $1/n \cdot (1 - 1/n)^{n-1} \geq 1/(en)$ since it suffices to reach the next successor on the path with a 1-bit-mutation. The expected number of progressing relevant steps in T mutation steps is $T/(en)$ and by Chernoff bounds, the probability to have less than $(1 - c)T/(en)$ progressing relevant steps is $2^{-\Omega(n)}$.

The lower bound on $\Delta^{\mathrm{right}}_{\mathrm{mut}}$ can be obtained in the same manner, except that changes on the left path can dominate changes on the right path due to the larger weight in the definition of the functions. A relevant mutation step where the index on the right path is decreased due to a progress on the left path is called a *regressing relevant step*. The probability of a regressing relevant step is $(1 + o(1)) \cdot 1/n^4$. Let $Z_1, \ldots, Z_m$ describe the regress in m mutations, then for $S = Z_1 + \cdots + Z_m$ both $\mathrm{E}(S) = (1 + o(1)) \cdot m/n^4$ and $\mathrm{V}(S) = (1 + o(1)) \cdot m/n^4$ holds. Lemma 2 yields

$$\mathrm{Prob}\big(S \geq (1 + o(1)) \cdot m/n^4 + c \cdot m/n^3\big) = 2^{-\Omega(m\cdot n^{-7/2})}.$$

Thus, the probability to have a total regress of $(1 + o(1)) \cdot cm/n^3$ is $2^{-\Omega(n^{1/2})}$ if $m = (1 + c) \cdot T/e$. Setting progress against regress, the net progress on the right path is $(1 - c - (1 + o(1))c) \cdot T/(en^3)$ with probability $2^{-\Omega(n^{1/2})}$. Choosing c such that $(1 - c - (1 + o(1))c) \geq (1 - \varepsilon)$ completes the proof.

□

Lemma 4. *Let* $\Delta^{\mathrm{right}}_{\mathrm{enh}}$ *be the progress on the right path in* $T = O(n^4)$ *extended mutations of parents whose index on the right path is greater than* 0*. Let* $\delta \geq 6$*, then with probability* $1 - 2^{-\Omega(n^{1/4})}$

$$-\frac{4T^{3/4}}{n^{3/2}} - n^{1/2} < \Delta^{\mathrm{right}}_{\mathrm{enh}} < \frac{4T^{3/4}}{n^{3/2}} + n^{1/2}.$$

Proof. Like in the proof of Lemma 3, we only consider mutations flipping less than K bits as the complementary event is considered an error. Consider the case that we have Hamming distance at least K to both the start and the end of a path. Then an extended mutation creating the offspring where the index increases by some $i < K$ has the same probability as an extended mutation creating the offspring where the index decreases by i. The upper bound on the progress will turn out to be ignorant of the fact that the algorithm may get closer than K to the end of the path. This implies that for the created offspring, a progress of $0 < i \leq K$ is at least as probable as a regress of i (i. e. a progress of $-i$). Taking into account the effects of selection, this property also holds for the selected individual. As a consequence, an upper bound on the progress on either path also represents an upper bound for the regress on that path and it suffices to prove an upper bound on the progress.

We distinguish two cases according to T. Let $T < n^{5/2}/9$. The probability to have progress $3i$ for some $1 \leq i < K/3$ in one extended mutation can be computed as follows. With probability $(1/n)^{3i} \cdot (1-1/n)^{n-3i}$, $P^* := P_{i(x'')+3i}$ is reached directly by mutation. Moreover, there are $3i$ Hamming neighbors of P^* with a Hamming distance of $3i-1$ to x and $n-3i$ Hamming neighbors of P^* with a Hamming distance of $3i+1$ to x. Thus, the probability to have progress $3i$ on the right path is

$$\left(\frac{1}{n}\right)^{3i}\left(1-\frac{1}{n}\right)^{n-3i} + 3i\left(\frac{1}{n}\right)^{3i-1}\left(1-\frac{1}{n}\right)^{n-3i+1} + (n-3i)\left(\frac{1}{n}\right)^{3i+1}\left(1-\frac{1}{n}\right)^{n-3i-1}$$
$$\leq \left(1-\frac{1}{n}\right)^{n-K}\left(\frac{1}{n}\right)^{3i}\left[1+3i\cdot n+\frac{n-3i}{n}\right]$$
$$= \frac{1}{e}\cdot(1+o(1))\cdot 3i\cdot n^{-3i+1}$$

Now imagine a sequence of binary random variables of infinite length where each variable takes value 1 with probability $p = 3n^{-2}$. The probability to obtain a block of $i \geq 1$ consecutive variables with value 1 is

$$\begin{aligned}
&\text{Prob(block of } i \text{ consecutive ones)}\\
&= 3^i \cdot n^{-2i} \cdot (1-p)\\
&\geq 3i \cdot n^{-3i+1} \cdot (1-p)\\
&\geq \frac{1}{e}\cdot(1+o(1))\cdot 3i\cdot n^{-3i+1}\\
&\geq \text{Prob(progress } 3i \text{ on the right path)}
\end{aligned}$$

since $(1-p) \geq 1/e \cdot (1+o(1))$ if n large enough.

Hence, the random process describing the number of ones among the first T random variables stochastically dominates the random process describing the progress on the right path divided by 3. We can now apply Chernoff bounds on the former process since all random variables are independent.

$$\begin{aligned}\text{Prob}\left(\Delta_{\text{enh}}^{\text{right}} \geq n^{1/2}\right) &\leq \frac{e^{(n^{1/2}/(pT)-1)pT}}{(n^{1/2}/(pT))^{n^{1/2}}} \\ &\leq e^{n^{1/2}} \cdot (n^{5/2}/(3T))^{-n^{1/2}} \\ &\leq e^{n^{1/2}} \cdot 3^{-n^{1/2}} = 2^{-\Omega(n^{1/2})}\end{aligned}$$

where the last inequality follows from $T < n^{5/2}/9$.

Now let $T \geq n^{5/2}/9$. The probability to have progress of at least 6 or at most -6 in one extended mutation is $O(n^{-5})$. By Chernoff bounds, it is easy to show that the probability to have more than $n^{1/2}$ of those larger steps within T steps is $2^{-\Omega(n^{1/2})}$. Even if we assume that all these steps yield a progress of K, the total progress by larger steps is less than n.

We now concentrate on steps yielding a progress of ± 3. The expected number of these steps is at most $3Tn^{-2}$ and the probability to have less than $2Tn^{-2}$ or more than $4Tn^{-2}$ of these steps is $2^{-\Omega(n^{1/2})}$ by Chernoff bounds. Let s be the number of ± 3-steps and let $X_1, \dots, X_s \in \{-1, +1\}$ be random variables such that $3X_i$ describes the progress by the ith ± 3-step. Then

$$\left|\text{E}\left(\Delta_{\text{enh}}^{\text{right}} \mid X_1, \dots, X_i\right) - \text{E}\left(\Delta_{\text{enh}}^{\text{right}} \mid X_1, \dots, X_{i-1}\right)\right| \leq 1$$

and we can apply the method of bounded martingale differences (Theorem 3.67 in [10]). Let $X := X_1 + \cdots + X_s$, then

$$\text{Prob}\left(X \geq \text{E}(X) + s^{3/4}\right) \leq e^{-s^{3/2}/(2s)} \leq e^{-s^{1/2}/2}.$$

Since $s \geq 2Tn^{-2} = \Omega(n^{1/2})$, the e-term is $2^{-\Omega(n^{1/4})}$.

Now we bound $\text{E}(X)$. Let x be the current well-formed search point, y the offspring obtained by an extended mutation and z be the search point obtained by applying local search to y. In case another well-formed search point is reached with z, it is very likely that due to improvements by local search $i(z') > i(x')$ holds and z is accepted regardless of $i(z'')$. Hence, as long as we do not reach the end points with indices 0 and r on the right path, a progress of 3 has the same probability as a progress of -3. Thus, the progress of ± 3-steps can be modelled by a symmetric random walk. The only exception is an extended mutation yielding a large regress on the left path such that $i(z') = i(x')$. In such a case, the sign of $i(z'') - i(x'')$ determines whether z is accepted or not. However, since $\delta \geq 6$, the probability for such a pathological step is $O(n^{-5})$. Even if we estimate the progress in such a step by the trivial bound K, the contribution of all these steps in $T = O(n^4)$ extended mutations to the expected progress is $o(1)$.

Since in non-pathological steps progresses of 3 and -3 have the same probability, the expected progress by these steps is 0. Thus, we have shown $\text{E}(X) = o(1)$ and $\Delta_{\text{enh}}^{\text{right}} \leq (4Tn^{-2})^{3/4} + n + o(1) \leq 4T^{3/4}n^{-3/2}$ with probability $2^{-\Omega(n^{1/4})}$. □

Lemma 5. *Let $\Delta_{\text{enh}}^{\text{left}}$ be the progress on the left path in $T \geq n^{1/2}, T = poly(n)$ extended mutations. Then with probability $1 - 2^{-\Omega(n^{1/2})}$ for $\varepsilon > 0$*

$$-(1+\varepsilon) \cdot \frac{T}{e} < \Delta_{\text{enh}}^{\text{left}} < (1+\varepsilon) \cdot \frac{T}{e}.$$

Lemma 6. *Let $\Delta_{\text{ls}}^{\text{left}}$ be the progress on the left path in $T = poly(n)$ calls of local search. Then with probability $1 - 2^{-\Omega(T)}$ for $\varepsilon > 0$*

$$(\delta - 1) \cdot (1-\varepsilon) \cdot \frac{2T}{e} \leq \Delta_{\text{ls}}^{\text{left}} \leq \delta \cdot (1+\varepsilon) \cdot \frac{2T}{e}.$$

Proof. The probability that an extended mutation leads to an offspring that is accepted after local search is at least $(1-1/n)^n + (1-1/n)^{n-1} = 2/e - o(1)$ since a sufficient condition is to flip at most one bit in the mutation step. Afterwards, as long as the end of the left path is not found, local search leads to a progress of $\delta - 1$ or δ. By Chernoff bounds, the probability to have less than $(1-\varepsilon) \cdot 2/e \cdot T$ of these events in T steps where local search is called is $2^{-\Omega(T)}$. Thus, the probability that local search leads to a progress less than $(\delta - 1) \cdot (1-\varepsilon) \cdot 2T/e$ is $2^{-\Omega(T)}$.

The upper bound can be proved similarly. It is easy to show that the probability that an extended mutation leads to an offspring accepted after local search is bounded above by $(1 + o(1)) \cdot 2/e$. By Chernoff bounds, the probability to have more than $(1+\varepsilon) \cdot 2/e \cdot T$ of these events in T generations with local search is $2^{-\Omega(T)}$. Thus, the probability that local search leads to a progress larger than $\delta \cdot (1+\varepsilon) \cdot 2T/e$ is $2^{-\Omega(T)}$. □

Finally, we are able to prove our main theorem.

Proof of the Main Theorem. Let $\ell = \frac{1-\varepsilon}{e} \cdot \left(n^3 + \frac{(2\delta-4)n^4}{\tau}\right)$ be the length of the left path and $r = n^5 - 1 + \frac{1+\varepsilon}{e} \cdot n + \frac{4n^{3/2}}{\tau^{3/4}} + n^{1/2} + 2K$ be the length of the right path for a small enough constant $\varepsilon > 0$. W. l. o. g. $r/3 \in \mathbb{N}_0$.

We investigate typical runs of the (1+1) MA with local search frequency $1/\tau$ and $1/(2\tau)$ on $\text{Race}_{\ell,\text{r}}^{\text{left}}$ and $\text{Race}_{\ell,\text{r}}^{\text{right}}$. The following statements hold w. o. p. By Corollary 1, the (1+1) MA reaches some well-formed search point x_{first} with $i(x'_{\text{first}}) < K$ and $|i(x''_{\text{first}}) - (n^5 - 1)| < K$ within the first n^4 steps.

We consider a period of n^4 generations of the (1+1) MA with local search frequency $1/\tau$ after x_{first} has been reached. Let Δ^{left} be the total progress on the left path and Δ^{right} be the total progress on the right path in n^4 generations. Then we apply Lemmas 3, 5, and 6 w. r. t. $n^4 - n^4/\tau$ mutations, n^4/τ extended mutations or n^4/τ local search calls, respectively. We obtain

$$\begin{aligned} i(x'_{\text{first}}) + \Delta^{\text{left}} &\geq \frac{1-\varepsilon}{e} \cdot \left(n^3 - \frac{n^3}{\tau}\right) - \frac{1+\varepsilon}{e} \cdot \frac{n^4}{\tau} + (\delta - 1) \cdot (1-\varepsilon) \cdot \frac{2n^4}{e\tau} \\ &= \frac{1-\varepsilon}{e} \cdot \left(n^3 - \left(2 + \frac{1+\varepsilon}{1-\varepsilon} + \frac{1}{n}\right) \frac{n^4}{\tau} + \frac{2\delta n^4}{\tau}\right) \\ &\geq \frac{1-\varepsilon}{e} \cdot \left(n^3 + \frac{(2\delta-4)n^4}{\tau}\right) = \ell \end{aligned}$$

if $\varepsilon < 1/3$ and n is large enough. Thus, the end of the left path is reached within the considered period.

Moreover, we show that the end of the right path is not reached within this period. First, we consider the very last search point with index $i(x''_{\text{first}}) + \Delta^{\text{right}}$ on the right path and show that the probability of $i(x''_{\text{first}}) + \Delta^{\text{right}} > r - K$ is exponentially small, i. e., the last considered search point is by at least K path points away from the end of the right path with overwhelming probability. This is done by applying Lemmas 3 and 4 where Lemma 4 can be applied since $i(x''_{\text{first}}) \geq n^5 - 1 - K$ and n^4 steps can only decrease the index by $n^4 \cdot K$ implying that the index on the right path cannot become 0.

We obtain

$$i(x''_{\text{first}}) + \Delta^{\text{right}} \leq n^5 - 1 + K + \frac{1+\varepsilon}{e} \cdot n + \frac{4n^{3/2}}{\tau^{3/4}} + n^{1/2} = r - K.$$

Observe that the probability to reach the end of a path cannot increase with decreasing number of generations. Hence, this bound also holds for all other search points reached within the period and the error probability increases by a factor of n^4.

Together, the (1+1) MA with local search frequency reaches the end of the left path within $O(n^4)$ generations and $O(n^4 + n \cdot \delta/\tau) = \mathit{poly}(n)$ function evaluations. This implies that on $\text{Race}^{\text{left}}_{\ell,\text{r}}$, a global optimum is found and the (1+1) MA is efficient. On $\text{Race}^{\text{right}}_{\ell,\text{r}}$, however, since $i(x''_{\text{first}}) + \Delta^{\text{right}} \leq r - K$ the Hamming distance to the end of the right path is at least K. As all search points closer to P_r now have worse fitness, the only way to reach a global optimum is a direct jump flipping at least K bits. The probability for such an event is at most $n^{-K} = 2^{-\Omega(n^{1/2} \log n)}$, thus the (1+1) MA fails on $\text{Race}^{\text{right}}_{\ell,\text{r}}$.

The argumentation for the (1+1) MA with local search frequency $1/(2\tau)$ is symmetric. We now consider a period of $\sqrt{2}n^4$ generations of the (1+1) MA with local search frequency $1/(2\tau)$ after x_{first} has been reached and define Δ^{left} and Δ^{right} according to this new period. Then we apply Lemmas 3, 5, and 6 w. r. t. $\sqrt{2}n^4 - \sqrt{2}n^4/(2\tau)$ mutations, $\sqrt{2}n^4/(2\tau)$ extended mutations or $\sqrt{2}n^4/(2\tau)$ local search calls, respectively.

We will show that $i(x''_{\text{first}}) + \Delta^{\text{right}} \geq r$ and $i(x'_{\text{first}}) + \Delta^{\text{left}} \leq \ell - K$. Repeating the line of thought from above, the (1+1) MA is efficient on $\text{Race}^{\text{right}}_{\ell,\text{r}}$ and it gets trapped on $\text{Race}^{\text{left}}_{\ell,\text{r}}$.

First, we prove $i(x''_{\text{first}}) + \Delta^{\text{right}} \geq r$ using Lemmas 3 and 4. We have

$$\begin{aligned} i(x''_{\text{first}}) + \Delta^{\text{right}} &\geq n^5 - 1 - K + (1-\varepsilon) \cdot \frac{\sqrt{2}n^4 - \sqrt{2}n^4/(2\tau)}{en^3} - \frac{2^{13/8} n^{3/2}}{\tau^{3/4}} - n^{1/2} \\ &= n^5 - 1 - K + \frac{1-\varepsilon}{e} \cdot \sqrt{2}n - \frac{2^{13/8} n^{3/2}}{\tau^{3/4}} - o(n). \end{aligned}$$

By hypothesis $\tau = \omega(n^{2/3})$ implying $2^{29/8} \cdot \frac{n^{3/2}}{\tau^{3/4}} + n^{1/2} + 3K = o(n)$. Adding the left hand side and another term $-o(n)$ yields

$$i(x''_{\text{first}}) + \Delta^{\text{right}} \geq n^5 - 1 + \frac{1-\varepsilon}{e} \cdot \sqrt{2}n + \frac{4n^{3/2}}{\tau^{3/4}} + n^{1/2} + 2K - o(n).$$

If $\varepsilon > 0$ is small enough such that $(1-\varepsilon)\cdot\sqrt{2} > (1+\varepsilon)$ and n is sufficiently large,

$$i(x''_{\text{first}}) + \Delta^{\text{right}} \geq n^5 - 1 + \frac{1+\varepsilon}{e}\cdot n + \frac{4n^{3/2}}{\tau^{3/4}} + n^{1/2} + 2K = r.$$

Finally, we show $i(x'_{\text{first}}) + \Delta^{\text{left}} \leq \ell - K$ using Lemmas 3, 5, and 6.

$$\begin{aligned}
i(x'_{\text{first}}) + \Delta^{\text{left}} &\leq K + \frac{1+\varepsilon}{e}\cdot\sqrt{2}\cdot\left(\frac{n^4 - n^4/(2\tau)}{n} + \frac{n^4}{2\tau} + \delta\cdot\frac{n^4}{\tau}\right)\\
&< K + \frac{1+\varepsilon}{e}\cdot\sqrt{2}\cdot\left(n^3 + \frac{(\delta+1)n^4}{\tau}\right)\\
&= \frac{1+\varepsilon}{e}\cdot\sqrt{2}\cdot\left(n^3 + \frac{(\delta+1)n^4}{\tau}\right) - K + o(n^3)
\end{aligned}$$

where the last equality follows from $2K = o(n^3)$. If ε is small enough such that $(1+\varepsilon)\cdot\sqrt{2} < (1-\varepsilon)\cdot 3/2$ and n is large enough,

$$\begin{aligned}
i(x'_{\text{first}}) + \Delta^{\text{left}} &\leq \frac{1-\varepsilon}{e}\cdot\left(\frac{3}{2}\cdot n^3 + \frac{3}{2}\cdot\frac{(\delta+1)n^4}{\tau}\right) - K\\
&= \frac{1-\varepsilon}{e}\cdot\left(n^3 + \frac{\tau}{2n}\cdot\frac{n^4}{\tau} + \frac{3}{2}\cdot\frac{(\delta+1)n^4}{\tau}\right) - K
\end{aligned}$$

By hypothesis, $\delta/\tau \geq 2/n$ and $\delta \geq 22$ implying $(\delta - 11)/\tau \geq 1/n$. Plugging this into the above equality yields

$$\begin{aligned}
i(x'_{\text{first}}) + \Delta^{\text{left}} &\leq \frac{1-\varepsilon}{e}\cdot\left(n^3 + \frac{\delta-11}{2}\cdot\frac{n^4}{\tau} + \frac{3}{2}\cdot\frac{(\delta+1)n^4}{\tau}\right) - K\\
&= \frac{1-\varepsilon}{e}\cdot\left(n^3 + \frac{(2\delta-4)\cdot n^4}{\tau}\right) - K = \ell - K.
\end{aligned}$$

□

5 Conclusions

We presented a rigorous theoretical analysis of a simple memetic algorithm, the (1+1) MA, thus showing that these randomized search heuristics can be analyzed in terms of computational complexity. On worst-case instances we have shown that the choice of the local search frequency has an enormous impact on the performance of the (1+1) MA: with overwhelming probability, even altering the parameterization by a factor of 2 turns a polynomial runtime behavior into an exponential one and vice versa.

Furthermore, we have gained insights into the behavior of memetic algorithms and into the interplay of mutation and local search. Although more work has to be done to obtain rigorous analyses of more complex memetic algorithms on real practical problems, this work is a valuable contribution to this promising research area.

References

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

[2] D. E. Goldberg and S. Voessner. Optimizing global-local search hybrids. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, pages 220–228, 1999.

[3] W. E. Hart. *Adaptive Global Optimization with Local Search*. PhD thesis, University of California, San Diego, CA, 1994.

[4] W. E. Hart, N. Krasnogor, and J. E. Smith, editors. *Recent Advances in Memetic Algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing*. Springer, 2004.

[5] N. Krasnogor. *Studies on the Theory and Design Space of Memetic Algorithms*. PhD thesis, University of the West of England, Bristol, 2002.

[6] H. R. Lourenço, O. Martin, and T. Stützle. Iterated local search. In *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 321–353. Kluwer Academic Publishers, Norwell, MA, 2002.

[7] P. Merz. Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12(3):303–326, 2004.

[8] P. Moscato. Memetic algorithms: a short introduction. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 219–234. McGraw-Hill, 1999.

[9] C. H. Papadimitriou, A. A. Schäffer, and M. Yannakakis. On the complexity of local search. In *Proceedings of the 22nd ACM Symposium on Theory of Computing (STOC)*, pages 438–445, 1990.

[10] C. Scheideler. *Probabilistic Methods for Coordination Problems*. HNI-Verlagsschriftenreihe 78, University of Paderborn, 2000. Habilitation Thesis, available at `http://www14.in.tum.de/personen/scheideler/index.html.en`.

[11] D. Sudholt. On the analysis of the (1+1) memetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, 2006. To appear. Technical report available at `http://sfbci.cs.uni-dortmund.de`.

[12] J.-P. Watson, A. E. Howe, and L. D. Whitley. An analysis of iterated local search for job-shop scheduling. In *Fifth Metaheuristics International Conference (MIC 2003)*, 2003.