Electronic Colloquium on Computational Complexity, Report No. 76 (2004)

# Searching Randomly for Maximum Matchings

Oliver Giel* and Ingo Wegener*

FB Informatik, LS 2, Univ. Dortmund, 44221 Dortmund, Germany
{oliver.giel, ingo.wegener}@cs.uni-dortmund.de

**Abstract.** Many real-world optimization problems in, e. g., engineering or biology have the property that not much is known about the function to be optimized. This excludes the application of problem-specific algorithms. Simple randomized search heuristics are then used with surprisingly good results. In order to understand the working principles behind such heuristics, they are analyzed on combinatorial optimization problems whose structure is well-studied. The idea is to investigate when it is possible to "simulate randomly" clever optimization techniques and when this random search fails. The main purpose is to develop methods for the analysis of general randomized search heuristics. The maximum matching problem is well suited for this approach since long augmenting paths do not allow local improvements and since our results on randomized local search and simple evolutionary algorithms can be compared with published results on the Metropolis algorithm and simulated annealing.

## 1 Introduction

Jerrum and Sorkin (1998) analyze the Metropolis algorithm (which is simulated annealing at a fixed temperature) for the graph bisection problem and they motivate their paper in the following way: "Our main contribution is not, then, to provide a particularly effective algorithm for the minimum bisection problem on random instances, but to analyse the performance of a popular heuristic applied to a reasonably realistic problem in combinatorial optimization." Here, we investigate the performance of simple randomized search heuristics on the maximum matching problem for different graph classes. The choice of the maximum matching problem will be motivated later. First, we want to motivate the analysis of search heuristics which certainly will not outperform clever algorithms designed for the solution of the considered problem.

We investigate randomized local search (RLS) and a simple but fundamental evolutionary algorithm ((1+1) EA) and compare our results with results of Sasaki and Hajek (1988) and Jerrum and Sorkin (1998) for the Metropolis algorithm (MA) and simulated annealing (SA). Jerrum (1992) describes MA as "just one level of sophistication beyond randomized local search." The same holds for

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

ISSN 1433-8092

the (1+1) EA. SA is a little more sophisticated than MA since it adaptively controls its temperature with a cooling schedule.

Following Jerrum and Sorkin (1998) we also are convinced that these search heuristics do not outperform the best known algorithms on the famous and well-investigated problems of combinatorial optimization. Why do we nevertheless analyze these heuristics? Are there applications for these simple heuristics? We claim that there are many of them in real-world applications. In large projects, one has to solve algorithmic problems which are well structured, but efficient algorithmic solutions of such problems are not published. The development of such algorithms is possible, but there are not enough resources (time, money, or experts) to design such an algorithm. Then people are satisfied with randomized heuristics. More important is the following scenario, also called black-box scenario. There exists a function $f\colon S \to \mathbb{R}$ to be optimized, but no compact description of $f$ is available. The only possibility to gain information on $f$ is to measure $f(x)$ by an experiment or the computer simulation of such an experiment. This is a typical scenario in, e. g., engineering and sciences. In engineering, one has to optimize some kind of machine where there are $m$ free parameters, the $i$th one taking values in $A_i$. Then $S = A_1 \times \cdots \times A_m$ and nobody can predict the "quality" $f(a_1, \dots, a_m)$ of the machine corresponding to the choice of the parameter combination $(a_1, \dots, a_m)$. Simple randomized search heuristics are quite successful in such situations. We add the description of a particular problem from molecular biology. Enantioselective catalysts are used to separate molecules with good chemical or pharmaceutical properties from their chiral counterparts. Reetz (2001) has applied a simple evolutionary algorithm for the optimization of such catalysts (the $f$-values are determined by experiments) and has produced the best-known results in this area revealing also new biological insights which then have been used to further improve the results.

Hence, simple randomized search heuristics find applications and it would be useful to understand why they are efficient for some problems and not for others. Moreover, it would be interesting to understand which heuristic is appropriate for certain types of problems. However, it is impossible to analyze an algorithm on problem instances where the considered function $f$ is not known. Our hope is that we can learn something on these questions when we analyze the heuristics on problems with well-investigated structures.

The maximum matching problem is well suited for our purposes. Improvements of non-maximum matchings are possible by exchanging the roles of matching and non-matching edges on augmenting paths (Hopcroft and Karp (1973)). Algorithms like the blossom algorithm that search for augmenting paths are clever and their correctness proofs are non-trivial. The best known algorithms are complicated (Micali and Vazirani (1980), Vazirani (1994)). Motwani (1994) and Bast *et al.* (2004) have proved that, with high probability, for each matching of a randomized graph in the $G(n, p)$-model, $p \geq c/(n-1)$ and $c \geq \ln n$ in the first paper and $c \geq 35.1$ in the second paper, there exists an augmenting path of length $O(\log n)$ only. Short augmenting paths support the optimization process of randomized search heuristics. Most results of our paper consider situations where long augmenting paths are possible.

In Section 2, we introduce the fundamental search heuristics which are discussed later on. In particular, we present the search principles that will be compared. In Section 3, we present the results of other papers and motivate our results discussed in the later sections. In Section 4, it is proved that RLS and the (1+1) EA are polynomial-time randomized approximation schemes for all graphs, i. e., $(1+\varepsilon)$-optimal solutions can be obtained in expected polynomial time, and also in polynomial time with overwhelming probability. Because of this result we are later only interested in the time for exact optimization.

In Section 5 and Section 6, we investigate the behavior of the search heuristics on simple graphs, namely paths and trees. The case of RLS on paths is simple, but for the (1+1) EA on paths one already has to control the effect of non-local steps. Trees can be handled by RLS in expected polynomial time. In Section 7, a class of graphs which is particularly difficult for search heuristics (see also Sasaki and Hajek (1988)) is introduced. In Section 8, it is proved that RLS and the (1+1) EA need an exponential expected time to produce maximum matchings on these graphs, and, in Section 9, it is shown for a subclass of these graphs that the optimization time is exponential even with overwhelming probability. In Section 10, we discuss how our results can be generalized to other search heuristics. We finish the paper with some conclusions.

## 2 Randomized Search Heuristics and the Maximum Matching Problem

We work with the following model of the maximum matching problem. For graphs with $n$ vertices and $m$ edges, the search space $S$ equals $\{0,1\}^m$ and the search point $s$ is interpreted as the characteristic vector of the set of chosen edges. The function $f\colon S \to \mathbb{Z}$ is defined by $f(s) := s_1 + \cdots + s_m$ if $s$ describes a matching, i. e., no two chosen edges share a node. There are two possibilities to handle non-matchings. The first one is to start with the empty set and to forbid to accept non-matchings. The second one is to define a penalty for non-matchings which directs the search towards matchings. The penalty $p(v)$ of a vertex $v$ with degree $d(v)$ with respect to the chosen edges equals $r \cdot \max\{0, d(v) - 1\}$ where $r \geq m + 1$. Then $f(s)$ equals the number of chosen edges $s_1 + \cdots + s_m$ minus the sum of all $p(v)$. The scaling factor $r \geq m+1$ ensures that the fitness of non-matchings is strictly worse than the fitness of any matching. An individual-based randomized search heuristic on $\{0,1\}^m$ looks as follows.

*Initialization:* Choose $s \in \{0,1\}^m$ according to some probability distribution.

*Search:* Let $q(s)$ be a probability distribution on $\{0,1\}^m$. Let $s'$ be chosen according to $q(s)$.

*Selection:* Based on $f(s)$ and $f(s')$ decide whether $s$ or $s'$ is chosen as search point $s$ for the next step of the infinite loop consisting of search and selection.

We are interested in the time $T$ until an optimal search point is created. Time is measured as the number of calls of search and selection. We discuss three search distributions $q(s)$:

- choose a Hamming neighbor of $s$ uniformly at random (RLS1),
- with probability $1/2$ choose a Hamming neighbor of $s$ uniformly at random, and otherwise, choose a search point in Hamming distance 2 uniformly at random (RLS2),
- for each bit position $i$, let $s'_i := 1 - s_i$ with probability $1/m$ and $s'_i = s_i$, otherwise; the bit positions are handled independently (GS).

RLS1 is the typical operator of randomized local search. For the matching problem and all maximal but non-maximum matchings, RLS1 produces only smaller matchings or non-matchings. The other local search operator RLS2 may shorten or lengthen augmenting paths by two edges. The global search operator GS is the typical so-called mutation operator from evolutionary computation.

We discuss two selection procedures:

- choose $s := s'$ if and only if $f(s') \geq f(s)$ (elitist strategy),
- choose $s := s'$, if $f(s') \geq f(s)$, and otherwise choose $s := s'$ with probability $p(s, s') \in (0, 1)$ depending on $f(s) - f(s')$.

Randomized local search (RLS) uses a local search operator and the elitist selection strategy; in our case this makes sense only for RLS2. The (1+1) EA combines the elitist selection strategy with the global search operator GS. Considering long time intervals also steps happen where $s'$ is quite different from $s$. This can help to escape from local optima and makes the analysis more difficult. The Metropolis algorithm (MA) with temperature $T$ combines RLS1 with the non-elitist strategy where $p(s, s') = \exp(-(f(s) - f(s'))/T)$. Moreover, MA does not accept non-matchings. All these strategies are static ones. Dynamic search strategies vary their parameters. A dynamic (1+1) EA as analyzed by Droste *et al.* (2001) varies the probability that the GS operator flips a bit. The dynamic version of the Metropolis algorithm which varies the parameter "temperature $T$" is known as simulated annealing (SA). It is interesting to analyze the different search heuristics.

A typical situation for the heuristics is a non-maximum but maximal matching. The situation seems to be more difficult if no augmenting path is "short." MA and SA are based on the operator RLS1 which flips exactly one bit per step. Hence, they have to accept worse matchings to have a chance to find a better matching. RLS and the (1+1) EA do not accept worse matchings. In particular, RLS differs from the "frozen MA", i.e., MA at temperature $T = 0$. RLS has to search on the so-called "plateau" of matchings of the same size. If there is an augmenting path of length $2\ell + 1$, RLS may shorten the augmenting path by accepted 2-bit flips. It is possible to produce a selectable edge within $\ell$ such steps and then to increase the matching size by an accepted 1-bit flip. The (1+1) EA can work in the same way. Steps flipping many bits may lead to search points with a quite different set of augmenting paths. In any case, the search on fitness plateaus (investigated in general by Jansen and Wegener (2001)) is the main problem of RLS and the (1+1) EA if $f$ can take only polynomially many values (which is true for many problems in combinatorial optimization).

A search heuristic is called efficient if the expected optimization time $E(T)$ is small. Our upper bounds on $E(T)$ hold for arbitrary initial search points. Then we can conclude that $T$ is small even with overwhelming probability. By Markov's inequality, the success probability within $2 \cdot E(T)$ steps is at least $1/2$. All disjoint phases of length $2 \cdot E(T)$ independently have this success probability implying that the success probability within $p(n) \cdot (2 \cdot E(T))$ steps is at least $1 - 2^{-p(n)}$.

It is possible that $\text{Prob}(T \leq p_1(m)) \geq 1/p_2(m)$ for polynomials $p_1$ and $p_2$ although $E(T)$ grows exponentially with respect to $m$. Then the following dynamic multi-start variant of the algorithm works in expected polynomial time without knowing $p_1(m)$ and $p_2(m)$.

In Phase $i$, $i \geq 0$, perform $r_i = 2^i r_0$ independent runs of the considered heuristic and stop each run after $t_i = 2^i t_0$ steps.

The cost of Phase $i$ equals $r_i t_i = 4^i r_0 t_0$. Obviously, $r_0$ and $t_0$ should be polynomially bounded. A typical setting would be $r_0 = 1$ and $t_0 = m$. Pessimistically, we assume that the early phases are unsuccessful, more precisely, the Phases $0, \dots, k := \lceil \log p(m) \rceil$ where $p(m) := \max\{\lceil p_1(m)/t_0 \rceil, \lceil p_2(m)/r_0 \rceil\}$. Their total cost is $O(p(m^2) \cdot r_0 \cdot t_0)$. If $i > k$, we have $2^i r_0 \geq 2^{i-k} p_2(m)$ runs working $2^i t_0 \geq 2^{i-k} p_1(m)$ steps. The probability that they all are unsuccessful is bounded above by

$$\left(1 - 1/p_2(m)\right)^{p_2(m) \cdot 2^{i-k}} \; = \; e^{-\Omega(2^{i-k})}.$$

The cost of Phase $i$ is $4^i r_0 t_0 = O(p(m)^2 \cdot r_0 \cdot t_0) \cdot 4^{i-k}$ and this phase has to be preformed with a probability of $e^{-\Omega(2^{i-1-k})}$. Hence the expected cost is bounded by

$$O\big(p(m)^2 \cdot r_0 \cdot t_0\big) \cdot \Big(1 + \sum_{1 \leq j < \infty} 4^j e^{-\Omega(2^j)}\Big) \; = \; O\big(p(m)^2 \cdot r_0 \cdot t_0\big).$$

and, therefore, polynomially bounded. In applications, multi-start variants of randomized search heuristics are quite popular. In order to prove that also these variants are inefficient it is sufficient to prove that the success probability within exponentially many steps is exponentially small. Hence, for negative results, we are mostly interested in results of this kind.

## 3 Previous Results

There is some literature on the theoretical analysis of different kinds of RLS, MA, and SA. Their common feature is the locality of the search operator. Evolutionary algorithms differ mainly in three aspects from these heuristics, namely

– a more global search operator,
– an additional search operator (recombination or crossover) working on more than one search point, and
– a population of more than one search point considered at the same time.

The (1+1) EA focuses on the first aspect and has been analyzed theoretically in several papers. There are papers analyzing the (1+1) EA on classes of functions:

– unimodal functions (Droste *et al.* (1998)),
– linear functions (Droste *et al.* (2002)),
– quadratic polynomials (Wegener and Witt (2004a)), and
– monotone polynomials (Wegener (2001), Wegener and Witt (2004b)).

The analysis of the (1+1) EA on problems of combinatorial optimization is still in its infancy. Scharnow *et al.* (2002) have studied sorting as minimization of unsortedness of a sequence. They have analyzed several measures of unsortedness leading to different optimization problems. Moreover, they have investigated how evolutionary algorithms solve the single-source-shortest-path problem. Neumann and Wegener (2004) have done the same for the minimum spanning tree problem. These problems allow improvements by local steps which is not always the case for the maximum matching problem investigated in this paper.

Sasaki and Hajek (1988) and Jerrum and Sinclair (1989) have shown that MA and SA are polynomial-time randomized approximation schemes for the maximum matching problem. We obtain similar results for RLS and the (1+1) EA in Section 4. There are no classes of graphs where a polynomial bound on the expected time for exact optimization has been proved for any of the heuristics. Our results in Section 5 and Section 6 are first results of this kind.

It is interesting to investigate graphs which are difficult for heuristics without global knowledge. Such a class of graphs $G_{h,\ell}$ has been presented by Sasaki and Hajek (1988), see Section 7. They prove that SA has an exponential expected optimization time on these graphs if one starts with the empty matching and $h$ is large. Sasaki (1991) describes a general lower bound technique for MA which is large if there are many more search points with the same fitness $f^*$ than better search points reachable with positive probability from search points where the fitness is at most $f^*$. Sasaki (1991) has applied his method to maximum matchings for $G_{h,\ell}$, traveling salesperson, and graph bisection. Jerrum (1992) has applied similar ideas to $(2-\varepsilon)$-approximations of maximum cliques in random graphs. The disadvantage of this method is that it ensures only the existence of a search point $s$ such that the expected optimization time is exponentially large if one starts with $s$. We investigate the same class of graphs. In Section 9, we prove that RLS and the (1+1) EA need exponentially many steps with a probability exponentially close to 1. This bound holds for $h = \omega(\log m)$ and $h \leq \ell - 2$ when starting with the empty matching, with a matching containing "several" edges not contained in the unique perfect matching, or a randomly chosen matching. The maximal degree of a node in $G_{h,\ell}$ is $h+1$. It is of interest to prove that the heuristics can be fooled by graphs of constant degree. In Section 8, we obtain weaker results for $2 \leq h \leq \ell - 2$ and, therefore, results for the smallest nontrivial node degree, namely 3. It is worth noticing that $G_{2,\ell}$ is planar. For these cases we obtain an exponential lower bound on the expected optimization time when starting with an arbitrary non-optimal search point. However, the probability of an exponential run time is only proved to be $\Omega(1/m)$. For this result, it is sufficient to prove that an almost perfect matching (one edge less than a perfect matching) is created and then to analyze the plateau of almost perfect matchings. This plateau ensures the existence of a unique augmenting path which simplifies the analysis. In Section 9, we have to analyze earlier stages of the search where the search points correspond to matchings with many augmenting paths.

*Remark.* This paper contains results of two extended abstracts (Giel and Wegener (2003, 2005)).

## 4 RLS and the (1+1) EA are PRAS

In evolutionary computation, the initial search point is typically chosen uniformly at random. For most graphs, it is likely that such a search point describes a non-matching. Regardless of the initial search point, RLS and the (1+1) EA find matchings efficiently.

**Lemma 1.** *RLS and the (1+1) EA find matchings in expected time $O(m \log m)$.*

*Proof.* Let $p = r \cdot k$ be the sum of the vertex penalties for the search point $s$. Then $k$ is less than $2m$, the sum of all vertex degrees. Until a matching is found, the fitness function rewards the decrease of $p$. By definition, there are at least $\lceil k/2 \rceil \leq m$ edges chosen by $s$ whose elimination decreases $k$. The probability for a specific 1-bit flip equals $\Theta(1/m)$ for both algorithms. Hence, the expected waiting time to decrease $k$ is bounded by $O(m/k)$. Summing up for $1 \leq k < 2m$ yields the claim. □

The key to prove that RLS and the (1+1) EA are a PRAS for the maximum matching problem is the following result which follows from the theory on maximum matchings.

**Lemma 2.** *Let $G = (V, E)$ be a graph, $M$ a non-maximum matching, and $M^*$ a maximum matching. Then there exists an augmenting path with respect to $M$ whose length is bounded by $L := 2\lfloor |M|/(|M^*| - |M|) \rfloor + 1$.*

*Proof.* With respect to $M$, the edges not belonging to $M$ are called free. A node is called free or exposed iff all adjacent edges are free. An augmenting path connects two free nodes and alternates between free edges and $M$-edges. Let $G' = (V, E')$ be the graph whose edge set is defined by $E' := M \oplus M^*$ where $\oplus$ denotes the symmetric difference. The graph $G'$ consists of node-disjoint cycles and paths. Each cycle and each path of even length has the same number of $M$-edges and $M^*$-edges. Paths of odd length alternate between $M$-edges and $M^*$-edges. There is no such path starting and ending with an $M$-edge. Otherwise, it would be an augmenting path with respect to $M^*$. Hence, there are $|M^*| - |M|$ disjoint augmenting paths with respect to $M$. At least one has at most $\lfloor |M|/(|M^*| - |M|) \rfloor$ $M$-edges and, therefore, at most $L$ edges. □

**Theorem 1.** *For $\varepsilon > 0$, RLS and the (1+1) EA find a $(1+\varepsilon)$-optimal matching in expected time $O(m^{2\lceil 1/\varepsilon \rceil})$ independently from the choice of the first search point.*

*Proof.* The first phase of the search finishes when a matching is found. By Lemma 1, this phase is short enough. Afterwards, let $M$ be the current matching and let $M^*$ be an arbitrary maximum matching. The search is successful if $|M^*| \leq (1+\varepsilon)|M|$. Otherwise, by Lemma 2, there exists an augmenting path for $M$ whose length is bounded above by $L := 2\lfloor |M|/(|M^*| - |M|) \rfloor + 1$. Since $|M^*| > (1+\varepsilon)|M|$, we conclude that

$$\frac{|M|}{|M^*| - |M|} \; < \; \varepsilon^{-1}.$$

Consequently,

$$\left\lfloor \frac{|M|}{|M^*| - |M|} \right\rfloor \; \leq \; \begin{cases} \lfloor \varepsilon^{-1} \rfloor = \lceil \varepsilon^{-1} \rceil - 1 & \text{if } \varepsilon^{-1} \text{ is not an integer,} \\ \lfloor \varepsilon^{-1} \rfloor - 1 = \lceil \varepsilon^{-1} \rceil - 1 & \text{if } \varepsilon^{-1} \text{ is an integer.} \end{cases}$$

In any case, $L \leq 2\lceil 1/\varepsilon \rceil - 1$.

The probability that the (1+1) EA flips exactly the edges of an augmenting path of length $\ell$ is $\Theta(m^{-\ell})$. The expected waiting time is $\Theta(m^{\ell})$. It is sufficient to wait $|M^*| \leq m$ times for such an event where $\ell$ is always at most $L$. This proves the result for the (1+1) EA.

RLS can flip the augmenting path in $\lfloor \ell/2 \rfloor + 1$ steps. In each of the first $\lfloor \ell/2 \rfloor$ steps, the length of the augmenting path is decreased by 2 by flipping the first two or the last two edges and in the last step the remaining edge of the augmenting path is flipped. The probability that a phase of length $\lfloor \ell/2 \rfloor + 1$ is successful is bounded below by $\Omega((m^{-2})^{\lfloor \ell/2 \rfloor} \cdot m^{-1}) = \Omega(m^{-\ell})$ where we used the fact that the length $\ell$ of an augmenting path is odd. The expected number of unsuccessful phases preceding a successful phase is $O(m^{\ell})$. Again $\ell \leq L$. The difference to the case of the (1+1) EA is that a phase may consist of more than one step. However, in each step the probability that a phase is continued successfully is bounded above by $O(m^{-1})$. Hence, the expected phase length is $O(1)$. This also holds under the assumption that a phase is unsuccessful. The length of the successful phase equals $\lfloor \ell/2 \rfloor + 1$. Hence, the expected number of steps to improve the matching again is bounded by $O(\ell + m^{\ell}) = O(m^{\ell})$ which proves the theorem. □

Let $c$ be a constant such that Theorem 1 holds for the bound $c \cdot m^{2\lceil 1/\varepsilon \rceil}$. The next corollary follows by an easy application of Markov's inequality.

**Corollary 1.** *If we run RLS or the (1+1) EA for $4cm^{2\lceil 1/\varepsilon \rceil}$ iterations of the loop, we obtain a PRAS for the maximum matching problem, i. e., independently from the choice of the first search point, the probability of producing a $(1+\varepsilon)$-optimal solution is at least $3/4$.*

## 5 RLS and the (1+1) EA on Paths

After having seen that RLS and the (1+1) EA find good approximations to maximum matchings in expected polynomial time, we are interested in graphs where even maximum matchings are found in expected polynomial time. We start with the simple graph consisting of a path of $m$ edges. This graph allows a matching of maximal size for connected graphs, namely $\lceil m/2 \rceil$. The analysis of RLS and the (1+1) EA on this graph is not too difficult but it is interesting since it contains already aspects of more difficult analyses performed in later sections.

The first lemma gives the hitting time of a random walk that is essential for our analyses. The random walk is similar to the random walk describing the gambler's ruin problem but has a reflecting barrier. The lemma is proven by standard arguments.

**Lemma 3.** *Given the homogeneous Markov chain with state space $S = \{0, \ldots, \ell\}$, initial state $\ell \geq 2$, and positive transition probabilities $p(0,0) = 1$, $p(1,0) = r$, $p(1,2) = s$, $p(i, i-1) = p$ for $i \in \{2, \ldots, \ell\}$, $p(i, i+1) = q$ for $i \in \{2, \ldots, \ell - 1\}$, and $p(\ell, \ell) = q$, where $0 < p, r < 1$, $q = 1 - p$, and $s = 1 - r$ (see Fig. 1). The expected time to reach state 0 for the first time starting in state $\ell$ is*

$$h_{\ell,0} = \begin{cases} \ell^2 + \left(\frac{2}{r} - 3\right)\ell - \frac{1}{r} + 2 & \text{if } p = q = 1/2, \\ \frac{1}{q-p}\left(\frac{\left(\frac{q}{p}\right)^{\ell} - 1}{\frac{q}{p} - 1} + \frac{s}{r}\left(\frac{q}{p}\right)^{\ell-1} - \ell - \frac{s}{r}\right) + \frac{1}{r} & \text{if } p \neq q. \end{cases}$$

*In particular, for $p = q = r = s = 1/2$, $h_{\ell,0} = \ell^2 + \ell$.*

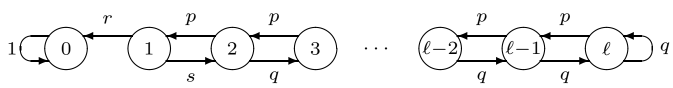


**Fig. 1.** The Markov chain in Lemma 3.

*Proof.* We claim that

$$h_{1,0} = \frac{1}{r} + \frac{s}{r} h_{2,1} \quad \text{and, for } j \geq 2, \quad h_{j,j-1} = \begin{cases} 2(\ell - j + 1) & \text{if } p = q = 1/2, \\ \frac{\left(\frac{q}{p}\right)^{\ell-j+1} - 1}{q - p} & \text{if } p \neq q. \end{cases}$$

Summing up the terms $h_{j,j-1}$ for $j \in \{1, \ldots, \ell\}$ yields the lemma. By the law of total probability, $h_{1,0} = 1 + r \cdot 0 + s \cdot (h_{2,1} + h_{1,0})$. This implies the first part of the claim. We prove the second part by induction on $j$. In accordance with the claim (for $p = q = 1/2$ and for $p \neq q$), $h_{\ell,\ell-1} = 1/p$ since the transition $(\ell, \ell-1)$ is the only transition leaving state $\ell$ with positive probability $p$. For $\ell - 1 \geq j \geq 2$, $h_{j,j-1} = 1 + p \cdot 0 + q(h_{j+1,j} + h_{j,j-1})$ implying that

$$h_{j,j-1} = \frac{1}{p}(1 + q \cdot h_{j+1,j}).$$

We apply the induction hypothesis and consider the cases $p = q = 1/2$ and $p \neq q$ separately. In the first case, we obtain

$$h_{j,j-1} = \frac{1}{p}\Big(1 + q\big(2(\ell - (j+1) + 1)\big)\Big) = 2(\ell - j + 1),$$

and in the second case,

$$h_{j,j-1} = \frac{1}{p}\left(1 + q\frac{\left(\frac{q}{p}\right)^{\ell-j} - 1}{q - p}\right) = \frac{\frac{q}{p} - 1}{p(\frac{q}{p} - 1)} + \frac{\left(\frac{q}{p}\right)^{\ell-j+1} - \frac{q}{p}}{q - p} = \frac{\left(\frac{q}{p}\right)^{\ell-j+1} - 1}{q - p}.$$

This proves the claim and finishes the proof. $\square$

In many situations, we will consider the number $R$ of so-called relevant steps rather than the total number of steps $T$. The definition of a relevant step will depend on the situation. If an expected number of $E(R)$ relevant steps is necessary to reach some target, and every step is relevant with probability at least $p$, then the expected total number of steps $E(T)$ is at most $p^{-1} \cdot E(R)$.

**Theorem 2.** *For a path of $m$ edges, the expected optimization time of RLS is $O(m^4)$ independently from the choice of the first search point.*

*Proof.* By Lemma 1, the expected waiting time for a matching is small enough. The size of a maximum matching equals $\lceil m/2 \rceil$. If the current matching size is $\lceil m/2 \rceil - i$, there exist at least $2i - 1 \geq i$ augmenting paths and one of length at most $\ell := m/i$. In every step, we select a shortest augmenting path $P$. Now a step is called $P$-relevant if it is accepted and $P$ is altered. The probability of a $P$-relevant step is $\Omega(1/m^2)$: If the length of $P$ is at least 3, it is lower bounded by the probability that a pair of edges at one end of $P$ flips (Fig. 2), otherwise, it is even $\Omega(1/m)$. If we can show that an expected number of $O(\ell^2)$ $P$-relevant steps is sufficient to improve the matching by one edge, then $\sum_{1 \leq i \leq \lceil m/2 \rceil} O((m/i)^2) = O(m^2)$ $P$-relevant steps are sufficient, and the expected optimization time is $O(m^4)$.

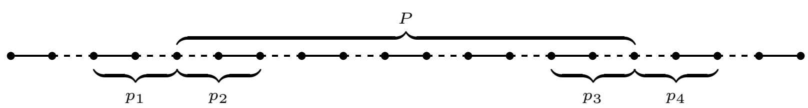


**Fig. 2.** In a $P$-relevant step, any of the pairs $p_1, \ldots, p_4$ flips.

If $|P| \geq 3$, there are no selectable edges. Only mixed mutation steps, where a free edge and a matching edge flip, can be accepted. Since each free edge $e$ has at least one neighbor $e'$ in the matching, $e'$ must flip, too. That means only a free edge $e$ incident upon a free node together with a matching edge $e'$ such that $e$ and $e'$ have an endpoint in common can flip. As our graph is a path, only pairs of neighbored edges located at one end of an alternating path can flip in an accepted step, and only the pairs $p_1, \ldots, p_4$ indicated in Fig. 2 are $P$-relevant. Since, for $|P| \geq 3$, the pairs $p_2$ and $p_3$ are always present, $P$ shrinks with a probability of at least $1/2$ in every $P$-relevant step. If $|P| = 1$, the probability that the length of the path is decreased to 0 in the next step is at least $1/(2m)$, and the probability that it grows at either end is at most $2 \cdot (1/2) \cdot \binom{m}{2}^{-1} = 2/(m(m-1))$. Hence, the conditional probability that the next $P$-relevant step is decreasing is at least $1/(1 + 4/(m-1)) \geq 1/2$, for $m \geq 5$. If we identify the current length $j$, $j \in \{0, 1, 3, 5, \ldots, \ell\}$ of $P$ with the state $\lceil j/2 \rceil$ of the Markov chain in Lemma 3, where $p = r = 1/2$, the states are $\{0, \ldots, \lceil \ell/2 \rceil\}$. Pessimistically assuming that the initial state is $\lceil \ell/2 \rceil$, the expected number of $P$-relevant steps to improve the matching is $O(\ell^2)$. □

The essential difference between RLS and the (1+1) EA is that the (1+1) EA may flip many bits in one step. We are only interested in $P$-relevant steps. For our analysis, we define $P$-clean steps which are $P$-relevant steps causing only small changes of $P$. Then a phase including $\Theta(\ell^2)$ $P$-relevant steps is called $P$-clean if all its $P$-relevant steps are $P$-clean. The idea is to prove that a phase is $P$-clean with probability $\Omega(1)$ and that a $P$-clean phase plus the next $P$-relevant step improve the matching with probability $\Omega(1)$.

**Theorem 3.** *For a path of $m$ edges, the expected optimization time of the (1+1) EA is $O(m^4)$ independently from the choice of the first search point.*

*Proof.* For the definition of $P$, $\ell$, and $P$-relevant steps see the proof of Theorem 2. With the same argument used there, it suffices to prove that the expected number of $P$-relevant steps to improve the matching is $O(\ell^2)$.

$P$-clean steps are only defined for situations without selectable edges. Let $u$ and $v$ be the endpoints of $P$, and let $E_u = \{\{w, z\} \in E \mid \mathrm{dist}(u, w) \leq 3\}$ be the set of edges where one endpoint has at most a distance of 3 to $u$, analogously for $E_v$ (see Fig. 3). Then we call a $P$-relevant step a $P$-clean step if

– at most three edges in $E' := E_u \cup E_v$ flip, and
– at most two of the flipping edges in $E'$ are neighbors.

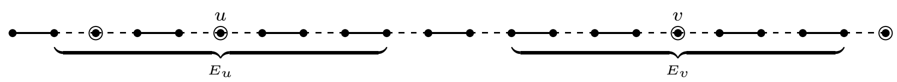


**Fig. 3.** Environments $E_u$ and $E_v$. Free nodes are marked by a circle.

We describe the effect of clean steps on $P$. The free nodes partition the graph into alternating paths (see Fig. 3 for an example). As there is no selectable edge, there is an augmenting paths of at least 3 edges between a free node and the next free node. There may be alternating paths of even length of at least 2 between the first node and the first free node and also between the last free node and the last node. Hence, a $P$-clean step cannot flip all edges of $P$ because this would require flipping a block of three edges in $E'$. Consequently, $P$ cannot vanish in a $P$-clean step; however, it is possible that new free nodes are created between $u$ and $v$. Then we interpret this event as a step shortening $P$ by at least two edges. It is impossible that a $P$-clean step lengthens $P$ by more than two edges i.e., at least 4 edges, since this requires flipping more than 3 edges in $E'$. Thus, $P$-clean steps lengthen $P$ only by 2 and it is necessary to flip one of the pairs $p_1$ or $p_4$ indicated in Fig. 2. For a $P$-clean step decreasing the length of $P$ by at least 2, it is sufficient to flip one of the pairs $p_2$ or $p_3$. Since at most three edges of $E'$ may flip, at most one pair of $p_1, \ldots, p_4$ can flip in a $P$-relevant step. Hence, $P$-relevant steps either lengthen or shorten $P$, and the probability of shortening steps is only larger than the probability of lengthening steps.

As the aim of a phase is to produce an improved matching or some selectable edge, it is convenient to include these good events into $P$-clean steps. We now broaden our definition of $P$-clean steps and call accepted steps that produce a selectable edge or improve the matching $P$-clean, too. Now we upper bound the probability of $P$-relevant but not $P$-clean steps (in situations without selectable edges). A necessary event to violate the first property is that 4 out of at most 16 edges of $E'$ flip. The probability of this event is $O(1/m^4)$. For the second property, let $k$ denote the length of a longest block $B$ of flipping edges in $E'$. The probability that a block of length $k \geq 4$ flips is upper bounded by the probability of the event that one out of a most 10 potential blocks of length 4 in $E'$ flips. The probability of this event is $O(1/m^4)$. A mutation step where $k = 3$ produces a local surplus of either one free edge or one matching edge in $B$. If the surplus is not balanced outside $B$, the step is either not accepted because the fitness decreases or the step is clean because the matching is improved. To compensate a surplus of one free edge, one more free edge than matching edges must flip elsewhere. This may be a free edge next to $B$ but outside $E'$ if $B$ is located at a border of $E'$. The probability of such a step is only $O(1/m^4)$. If $B$ is not located at a border of $E'$, another block $B'$ of at least three edges not neighboring $B$ has to flip. This results in a probability of at most $O((1/m^3)\cdot(m\cdot 1/m^3)) = O(1/m^5)$. If a local surplus of one matching edge has to be balanced, either only another matching edge flips and, because a selectable edge is created, the step is clean. Otherwise, another block of at least three edges must flip. The probability of the last possibility again is $O(1/m^5)$. Altogether, the probability of a $P$-relevant but not $P$-clean step is $O(1/m^4)$, and the conditional probability that a $P$-relevant step is not $P$-clean is $O(1/m^2)$. Hence, a phase of $O(\ell^2) = O(m^2)$ $P$-relevant steps is clean with a probability $\Omega(1)$.

Replacing shortening steps by shortenings by exactly two edges and using the lower bound 1/2 for $p$ and $r$ in Lemma 3, an expected number of $O(\ell^2)$ clean relevant steps reduces the length of $P$ to at most 1. By Markov's inequality, this happens with probability $\Omega(1)$ in $c\ell^2$ clean relevant steps if $c$ is large enough. Afterwards, at least one selectable edge $e$ exists, and a step is $P$-relevant with a probability of $\Omega(1/m)$, namely, if it flips only $e$ and, thereby, improves the matching. In order to destroy $e$, $e$ or a neighbor of $e$ has to flip. This happens only with a probability of $O(1/m)$. Hence, the next $P$-relevant step improves the matching with probability $\Omega(1)$. □

We discuss the results of Theorem 2 and 3. On the one hand, paths are difficult since augmenting paths tend to be rather long in the final stages of optimization. On the other hand, paths are easy since there are not many possibilities to lengthen an augmenting path. The time bound $O(m^4) = O(n^4)$ is huge but can be explained by the characteristics of general (and somehow blind) search. There are many rejected steps and many irrelevant steps which do not alter any augmenting path. In the case of $O(1)$ augmenting paths and no selectable edge, a step is relevant only with a probability of $\Theta(1/m^2)$, and the expected number of relevant steps is $O(m^2) = O(n^2)$. Indeed, the search on the level of second-best matchings is already responsible for this. If the number of edges is odd, the path graph has a unique maximum matching, namely the perfect matching choosing the $\lceil m/2 \rceil$ edges with odd numbers. Therefore, every second-best matching of size $\lfloor m/2 \rfloor$ has only one augmenting path $P$. Our aim is to show that both heuristics have an expected optimization time of $\Omega(m^4)$ if the initial situation is a second-best matching and $P$ is not too short.

First, we investigate a random walk on the set $\{0, 1, \ldots, L\}$, where $L$ is the initial state, $p(L, L-1) = 1$, $p(0,0) = 1$, and, for $1 \le i \le L-1$, $p(i, i-1) = p(i, i+1) = 1/2$. This random walk can be analyzed with the same method used for Lemma 3; however, for $p = r = 1/2$, the Markov chains differ only with respect to the probability to leave the initial state. It is easy to see that, for $\ell = L - 1$, the expected time to reach 0 in the chain in Lemma 3 lower bounds the expected time to reach 0 in the above random walk. Hence, starting in $L$, the expected time to reach 0 is at least $cL^2$ for some constant $c > 0$. For $0 < \varepsilon < 1$, we claim that the success probability within $\lfloor \varepsilon c L^2 \rfloor$ steps is bounded above by $\varepsilon$. If the success probability were larger than $\varepsilon$, this would hold for any initial state. Then, the expected number of phases of length $\lfloor \varepsilon c L^2 \rfloor$ would be less than $1/\varepsilon$ contradicting the expected time of at least $cL^2$.

**Theorem 4.** *For a path of $m$ edges, $m$ odd, the expected optimization time of RLS and the (1+1) EA is $\Theta(m^4)$ if the initial situation is a second-best matching with an augmenting path of length $\Omega(m)$.*

*Proof.* The upper bounds follow from Theorem 2 and Theorem 3. Let the initial length of the unique augmenting $P$ be at least $6L$ for some $L = \Omega(m)$. First, we investigate RLS. The left endpoint of $P$ has to walk at least $L$ steps of length 2 to the right, or the right endpoint has to walk at least the same distance to the left. If we do not allow the endpoints of the augmenting path to come closer to the endpoints of the graph than in the initial configuration, we can apply the result on the random walk considered above. The probability that at least one of these events occurs within $\alpha m^2$ such steps is bounded by $1/2$ if $\alpha > 0$ is small enough. The lower bound for RLS follows since the probability of a $P$-relevant step is $\Theta(1/m^2)$.

For the (1+1) EA, the considered 2-bit flips also have probability $\Theta(1/m^2)$ but we must take into account $2k$-bit flips for $k \geq 2$. We pessimistically assume that the latter only decrease the length of $P$ and show that this additional decrease is at most $L$. Then, the length of $P$ is always at least $L = \Omega(m)$ and the probability of a step flipping exactly the edges of $P$ is small enough. If $2k$ bits flip in an accepted step, they form one or two blocks where the last or first edge of a block is adjacent to one of the exposed endpoints of $P$. Thus, there are $O(k)$ possibilities for an accepted $2k$-bit flip and the expected decrease by means of $2k$-bit flips in a single step is $2k \cdot O(k/m^{2k}) = O(k^2/m^{2k})$. The sum for all $k \geq 2$ is $O(1/m^4)$. Hence, the expected decrease by more than two is $O(1/m^4)$ in each step. Within $\beta m^4$ steps, this expected decrease is $O(1)$ and the decrease is less than $L$ with probability $1 - o(1)$ if the constant $\beta > 0$ is small enough. □

## 6 RLS on Trees

There is a simple direct approach to construct maximum matchings on trees. Randomized search heuristics should be able to find such matchings in expected polynomial time. The expected optimization time $O(m^4)$ for paths, i. e., unary trees, holds, since the step from almost perfect matchings to perfect matchings is essential. Then the unique augmenting path can have length $\Omega(m)$ and it takes an expected number of $\Theta(m^2)$ $P$-relevant steps to overcome a distance of $\Theta(m)$. Paths are trees with maximal diameter. The diameter corresponds to the length of the longest augmenting path. If a free node $v$ has degree $\deg(v)$ and there is no selectable edge, this free node can move with equal probability along each adjacent edge and the following matching edge. If $v$ is an endpoint of an augmenting path, there can be $\deg(v) - 1$ directions which lengthen the augmenting path and there is at least one which shortens it. Increasing $\deg(v)$, the game gets more and more unfair. Many nodes of large degree imply the existence of many leaves, i. e., nodes of degree 1, since the average degree is less than 2. Moreover, this decreases the diameter. We conjecture that the expected optimization time of RLS on trees is bounded by $O(m^4)$.

RLS for the maximum matching problem can be considered as random walks of the free nodes until they almost meet; more precisely, until they produce a selectable edge. We cannot apply results on random walks of one token on a graph (see, e. g., Motwani and Raghavan (1995)) and even not results on the expected time until two randomly walking tokens meet. Coppersmith *et al.* (1993) prove an upper bound of $O(n^3)$ for this scenario. The number of free nodes can be much larger than 2. But even in the case of exactly two free nodes we obtain an exponential lower bound in Section 8. The reason it that the nodes do not walk to a randomly chosen neighbor. They choose a free edge adjacent to an exposed node randomly but then they are forced to choose the adjacent matching edge. Hence, the walk of one free node is influenced by the currently chosen matching and, in particular, by the position of the other free nodes. Hence, the free nodes play a pursuit game on a graph whose connections are influenced by the other players. To investigate such generalized pursuit games is a further motivation to analyze RLS for the maximum matching problem. First, we prove that the essential part is to produce a search point describing a matching with a selectable edge.

**Lemma 4.** *Given an arbitrary tree and a matching with at least one selectable edge. In expected time $O(m)$, RLS has improved the matching or there is no selectable edge but an augmenting path of length* 3*. The matching is improved with probability at least* $1/2$*.*

*Proof.* We prove the lemma by proving the following two claims for all search points describing matchings with a selectable edge: In the next step,

- the probability of improving the matching is at least $1/(2m)$, and
- the probability of destroying all selectable edges without improving the matching is at most $1/(2m)$.

The first claim is obvious, since $1/(2m)$ is the probability of flipping exactly a specified selectable edge. Now we prove the second claim. A step flipping only matching edges is not accepted. A step flipping only free edges is not accepted or it improves the matching. Hence, we only have to consider 2-bit flips choosing a free edge $e$ and a matching edge $e'$. Since $e'$ must not become selectable, $e$ has to be adjacent to $e'$ implying that $e$ is not selectable. To destroy a given selectable edge $e^*$, $e$ must also be adjacent to $e^*$. Summarizing, the free edge $e$ is adjacent to the selectable edge $e^*$ at its free endpoint, and at the other endpoint, $e$ is adjacent to the matching edge $e'$. This shows that an augmenting path of length 3 is created if $e$ and $e'$ flip. By the matching property, the choice of $e$ determines $e'$, and, in a tree, $e'$ determines $e$ between $e'$ and $e^*$. Hence, the possible pairs $\{e, e'\}$ are pairwise disjoint implying that their number is bounded above by $(m-1)/2$. The probability to flip any of these pairs is at most $((m-1)/2) \cdot (1/2) \cdot \binom{m}{2}^{-1} = 1/(2m)$. □

In the next lemma, we prove that it is not too unlikely to shorten an augmenting path.

**Lemma 5.** *Let $P = (x_0, x_1, \ldots, x_\ell)$ be an augmenting path with respect to a tree and a matching without selectable edge. An accepted mutation step of RLS preserves the current matching size and either*

- *leaves $P$ unchanged,*
- *lengthens $P$ by two edges at one end, or*
- *shortens $P$ by a multiple of two edges at one end.*

*If $P$ is changed, the probability that $P$ shrinks is at least $2/(\deg(x_0) + \deg(x_\ell))$.*

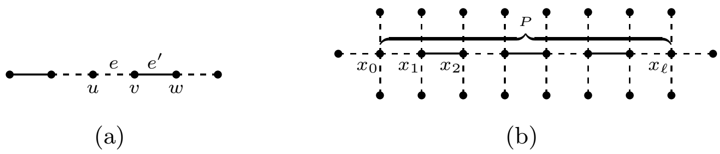


**Fig. 4.** (a) In an accepted step, a free edge $e$ incident upon an exposed node $u$ and a matching edge $e'$ adjacent to $e$ flip. (b) The augmenting path $P$ and its neighborhood. The figure only shows $P$ and nodes and edges adjacent to $P$.

*Proof.* Since no edge is selectable, augmenting paths have at least three edges. Hence, only steps preserving the matching size are accepted. This implies that only 2-bit flips choosing a free edge $e$ and a matching edge $e'$ are accepted. Since $e$ is free but not selectable, we obtain a new matching only if $e'$ is adjacent to $e$, i.e., $e = \{u, v\}$ and $e' = \{v, w\}$, and $u$ is exposed (Fig. 4(a)).

We investigate accepted steps changing $P$. There are four possibilities for the free edge $e = \{u, v\}$ (see Fig. 4(b)):

– $e$ is an "outer" edge of $P$, namely, $e = \{x_0, x_1\}$ or $e = \{x_{\ell-1}, x_\ell\}$,
– $e$ is an "inner" edge of $P$,
– $e$ does not belong to $P$ but its exposed endpoint $u$ does,
– $e$ and its exposed endpoint $u$ do not belong to $P$.

In the first case, $e'$ is adjacent to $e$ and lies on $P$. Flipping $e$ and $e'$ shortens $P$ by two edges. The second case is impossible because no endpoint of $e$ would be exposed. In the third case, either $u = x_0$ or $u = x_\ell$. Flipping $e$ and $e'$ lengthens $P$. Since $w$ becomes an exposed node, the length increases by exactly 2. In the last case, $P$ is only changed if the matching edge $e'$ connects two inner nodes of $P$. (To see this, observe that each edge incident upon a node of $P$ is either free or it is a matching edge belonging to $P$.) Then, $w$ is an inner node of $P$ and becomes an exposed node. Either $(x_0, \ldots, w)$ or $(w, \ldots, x_\ell)$ becomes an augmenting path. Since every augmenting path has odd length, the new path is by an even number of edges shorter than the old path $P$.

We have seen that the length of $P$ can only increase if $u = x_0$ or $u = x_\ell$. In such a step, the flipping free edge $e$ incident upon $x_0$ or $x_\ell$ determines which matching edge can flip. Therefore, there are at most $\deg(x_0) + \deg(x_\ell) - 2$ 2-bit flips increasing the length of $P$ and at least 2 2-bit flips decreasing the length of $P$. This proves the claimed probability. □

Since paths can be considered as complete unary trees, we first investigate complete $k$-ary trees, i.e., rooted trees where inner nodes have $k$ successors and all leaves have the same distance from the root. The diameter of these graphs is $\Theta(\log_k m)$. It is not difficult to prove that each matching $M$ with $m^*$ edges less than a maximum matching implies the existence of an augmenting path whose length is less than $2\log_k(2m/m^*)$.

**Lemma 6.** *Given a complete $k$-ary tree with $m$ edges and $k \geq 2$. If $M^*$ is a maximum matching and $M$ a non-maximum matching with $m^*$ edges less than $M^*$, there is an augmenting path with respect to $M$ whose length is strictly less than $L := 2\log_k(2m/m^*)$.*

*Proof.* The nodes of the tree in distance $h$ from the root are called level $h$. Let $d$ denote the depth of the tree implying $d \leq \log_k m$. Considering $M^* \oplus M$, we obtain $m^*$ node-disjoint augmenting paths. Let us assume that the length of a shortest augmenting path is $\ell$. A simple path in a tree can contain at most two nodes of each level. This implies that each simple path whose length is at least $\ell$ contains at least one node on a level $h \leq d - \lfloor(\ell+1)/2\rfloor = d - \lceil \ell/2 \rceil$. Hence, $m^*$ is bounded above by the number of nodes on the levels $0, \ldots, d - \lceil \ell/2 \rceil$ implying that

$$m^* \leq \frac{k^{d-\lceil \ell/2 \rceil + 1} - 1}{k-1} < \frac{k}{k-1} k^{d - \lceil \ell/2 \rceil} \leq 2mk^{-\lceil \ell/2 \rceil}.$$

Solving for odd $\ell$ yields the proposed bound $\ell < L$. □

Now we are prepared to finish the analysis of RLS on totally balanced trees.

**Theorem 5.** *The expected time until RLS finds a maximum matching on a complete $k$-ary tree, $k \geq 2$, is bounded by $O(m^{7/2})$ independently from the choice of the first search point.*

*Proof.* By Lemma 1, the expected time to find a matching is $O(m \log m)$. Afterwards, we estimate the expected number of $P$-relevant steps to improve the matching. A step is called $P$-relevant for an augmenting path $P$ if it is accepted and changes $P$. We will choose an appropriate path $P$ which may change during the process. Then, the expected number of all steps is only by a factor of $O(m^2)$ larger since the probability of a $P$-relevant step is $\Omega(1/m^2)$. Assume we can guarantee that there is always an augmenting path of length at most $\ell$. Pessimistically, we replace shortenings of the considered path $P$ by shortenings by only two edges. By Lemma 5, this leads to a probability of at least $p = 1/(k+1)$ of shortening this path in $P$-relevant steps in situations without selectable edges. In situations with selectable edges, we apply Lemma 4. In a step changing the situation, the probability to improve the matching is at least $r = 1/2$, and the probability to create a path of length three is at most $1/2$. Now we can represent the current length $j$ of the path by the state $\lceil j/2 \rceil$ of the Markov chain in Lemma 3. If we pessimistically start with a path of length $\ell$, the expected number of $P$-relevant steps until the matching is improved is at most

$$\frac{k+1}{k-1}\Big(\frac{k^{\lceil \ell/2 \rceil} - 1}{k-1} + k^{\lceil \ell/2 \rceil - 1} - (\lceil \ell/2 \rceil + 1)\Big) + 2 \; = \; O(k^{\ell/2}).$$

Until $m^* \leq 2m^{1/2}$, Lemma 6 guarantees that there is an augmenting path $P$ of length $\ell < \log_k m$. This leads to an expected number of $O(m^{1/2})$ $P$-relevant steps for an improvement. Since $|M^*| \leq m$, we apply this bound only $O(m)$ times implying that the expected number of $P$-relevant steps until $m^* \leq 2m^{1/2}$ is

$O(m^{3/2})$. Afterwards, we use the bound $\ell \leq 2\log_k m$ which is a trivial bound on the diameter of the tree. Now we obtain an upper bound of $O(m)$ on the number of $P$-relevant steps to improve the matching. Since we apply this bound only $\lfloor 2m^{1/2} \rfloor$ times, this leads to an additional expected number of $O(m^{3/2})$ $P$-relevant steps. This proves the theorem as argued at the beginning of this proof. □

Our upper bound for RLS on arbitrary trees depends also on the diameter of the tree.

**Theorem 6.** *The expected time until RLS finds a maximum matching in a tree with diameter $D$ is bounded by $O(D^2m^4)$ independently from the choice of the first search point.*

*Proof.* We investigate an arbitrary search point $s$ describing a non-maximum matching. We are interested in the expected time to obtain a search point describing a matching of the same size and where there exists a selectable edge. We choose an arbitrary augmenting path $P$ connecting $u$ and $v$. We have to investigate the movement of all free nodes. We focus our interest on the distance between $u$ and $v$. If this distance is only 1, the edge $\{u, v\}$ is selectable. We may be lucky and produce a selectable edge somewhere else before.

To obtain precise statements we fix some node $w$ as the root of the tree. The nodes $u$ and $v$ move around and $r$ denotes the initial position of $u$. In general, $T(z)$ is the subtree rooted at $z$ and $|T(z)|$ is its number of nodes. A step is called $u$-relevant, if $u$ moves. We are interested in the expected number of $u$-relevant steps until some progress is made. Progress is defined as the creation of a selectable edge, the event that $u$ and $v$ are both in $T(r)$, or the event that $u$ and $v$ are both not in $T(r)$.

*Claim 1.* If $v \notin T(r)$ and $v$ does not enter $T(r)$ before $u$ leaves it, the expected number of $u$-relevant steps until $u$ leaves $T(r)$ or a selectable edge is created is bounded above by $|T(r)|$.

*Proof.* The proof investigates the random walk of $u$ influenced by steps where an adjacent edge flips. Other steps moving $u$ are only in favor to us since $u$ moves closer to $r$ (see Lemma 5). The proof uses the argument that a large degree of $u$ implies that not many subtrees of $T(u)$ can be large. In the considered situation, the conditions of Lemma 5 hold and we may pessimistically assume that the path is never shortened by more than two edges in $u$-relevant steps. It is important to observe that $u$ can only visit nodes in $T(r)$ with an even distance to its initial position, the root $r$ of $T(r)$. The proof is by induction on $|T(r)|$. If $|T(r)| = 1$, then $T(r)$ is a leaf and the expected number of $u$-relevant steps is at most 1. Now let $|T(r)| > 1$ and let $s$ denote the degree of $r$. Let $T_1, \ldots, T_t$ denote those subtrees of $T(r)$ such that there are two edges between $r$ and the root of each $T_i$ (Fig. 5). Whenever $u$ is at $r$, $r$ is connected to $s-1$ nodes $u_1, \ldots, u_{s-1}$ in $T(r)$ and to its parent by free edges. Each node $u_j$ may be adjacent to several roots of the subtrees $T_1, \ldots, T_t$, but each $u_j$ is connected to at most one of these roots

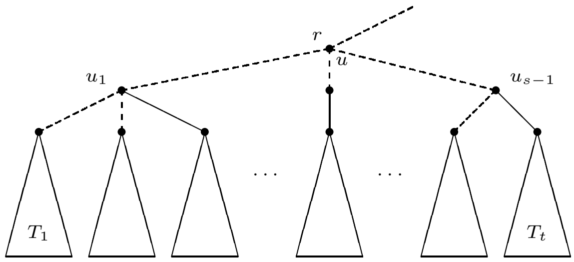


**Fig. 5.** The subtree $T(r)$. Initially, the endpoint $u$ is at the root $r$.

by a matching edge. Hence, at most $s-1$ roots of subtrees $T_i$ are reachable. When $u$ leaves a subtree $T_i$, it can only return to $r$ or an ancestor on the path. W. l. o. g. let $T_1, \ldots, T_{s-1}$ be $s-1$ largest subtrees in $\{T_1, \ldots, T_t\}$. For $T$ a subtree of $T(r)$ and $u$ starting at the root of $T$, let $E(T)$ denote the expected number of $u$-relevant steps until $u$ leaves $T$ (or a selectable edge is created). By the law of total probability,

$$E(T(r)) \leq \frac{1}{s} \cdot 1 + \frac{1}{s}\big(1 + E(T_1) + E(T(r))\big) + \cdots + \frac{1}{s}\big(1 + E(T_{s-1}) + E(T(r))\big).$$

Solving for $E(T(r))$ and using the induction hypothesis yields $E(T(r)) \leq s + |T_1| + \cdots + |T_{s-1}|$. Because $T(r)$ contains all subtrees $T_i$, the nodes $u_1, \ldots, u_{s-1}$, and $r$, the right-hand side of the last inequality is at most $|T(r)|$. $\square$

*Claim 2.* If $v \in T(r)$, the expected number of $u$-relevant steps until one of the following events occurs is bounded by $O(|T(r)|)$:

- there is a selectable edge,
- $u$ leaves $T(r)$ and $v$ remains in $T(r)$,
- $u$ and $v$ are in the same proper subtree of $T(r)$.

*Proof.* It is essential that the distance between $u$ and $r$ remains even. (Then the distance between $v$ and $r$ is odd.) Then one can prove that $v$ does not leave $T(r)$ in the considered phase. In the following, $T_u$ and $T_v$ denote subtrees of $T(r)$ rooted by a node on the level below $r$. If $u$ is in a proper subtree of $T(r)$, $T_u$ denotes the subtree of $T(r)$ currently containing $u$; analogously for $v$ (see Fig. 6). First, we show that $v$ cannot leave $T(r)$ in the time we consider. If $u$ is at $r$, then $v$ cannot leave $T(r)$. If $u$ leaves $T(r)$, the second stopping criterion is fulfilled. If $u$ moves to a node in $T_v$, the last stopping criterion is fulfilled. The endpoint $v$ might only leave $T(r)$ if $u$ first moves to a subtree $T_u \neq T_v$. If this is the case, the path $P$ between $u$ and $v$ visits $r$. By Lemma 5, relevant steps can increase and decrease the length of $P$ only by a multiple of two edges at either end. Since $u$ initially had an even distance to $r$, namely 0, $u$ keeps an even distance and $v$ keeps an odd distance to $r$. Now, if $v$ leaves its subtree $T_v$ in a shortening step, then $v$ moves to some node on the path $P$. Since $v$ cannot visit $r$ (odd distance), the new location of $v$ could only be in $T_u$ and the last stopping criterion would

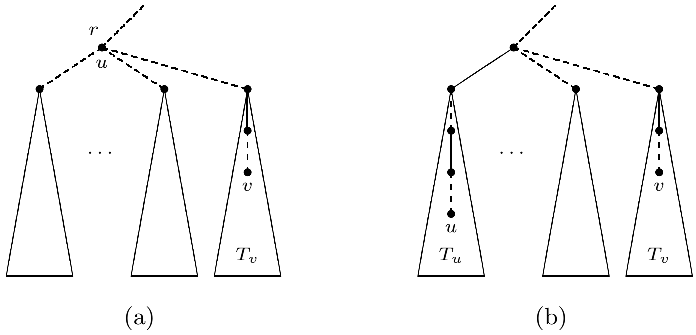


**Fig. 6.** The subtree $T(r)$. A special situation (a) and the situation when $u$ has moved into a subtree of $T(r)$ (b).

be fulfilled first. Hence, $v$ cannot leave $T(r)$ before one of the stopping criteria is fulfilled.

Now it is clear that we can upper bound the expected number of $u$-relevant steps until one of the stopping criteria is fulfilled by the expected number of $u$-relevant steps for the event that $u$ leaves $T(r)$ or moves into $T_v$ given that $v$ stays in its subtree $T_v$. This can be done in the same way as in the proof of Claim 1. The only difference is that the terms for subtrees contained in $T_v$ can now be bounded by 0. This can only improve the upper bound $O(|T(r)|)$ on the expected number of $u$-relevant steps. Note that the case $r = w$ where $u$ cannot leave $T(r)$ is included. □

In the following let $r$ be the root of the smallest subtree containing $u$ and $v$ and let $d(r)$ be the depth of $r$, i. e., the distance between $r$ and the root $w$ of the tree. We consider the random variable $d(r)$ with respect to time. A selectable edge is created if the distance between $u$ and $v$ equals 1. This is certainly the case if $r$ is the root of a subtree of depth 1. This happens if $d(r) \geq D - 1$. The idea is that $d(r)$ has the tendency to grow if we do not create a selectable edge.

*Claim 3.* If $u, v \in T(r)$ and $v$ starts at $r$, it is possible to define disjoint events $E_1, \ldots, E_4$ such that

- $E_1$ is the event that a selectable edge is created,
- $E_2$ implies that $d(r)$ is increased by at least 1,
- $E_3$ implies that $d(r)$ is increased by at least 2, and
- $E_4$ implies that $d(r)$ is decreased by at most 2.

After an expected number of $O(m)$ $u$-relevant steps, one of these events has happened and $\text{Prob}(E_3 \text{ happens first}) \geq \text{Prob}(E_4 \text{ happens first})$.

*Proof.* We consider only situations without selectable edges. Otherwise, $E_1$ has happened and we can stop. A *special situation* is a situation where one endpoint of the considered augmenting path $P$ is at the root $r$ of the smallest subtree $T(r)$

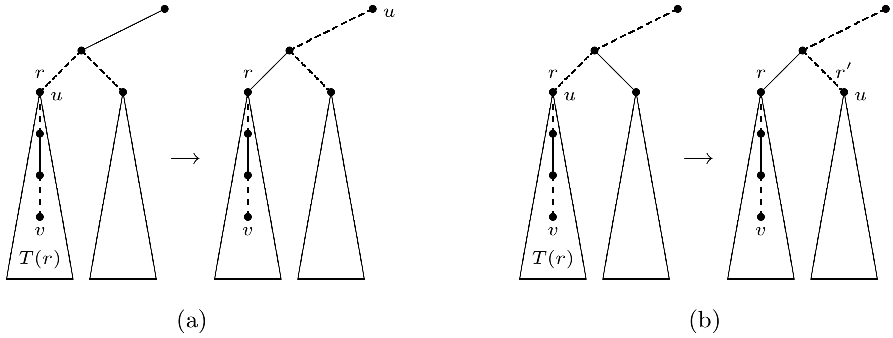


**Fig. 7.** If $d(r)$ decreases, $u$ moves to the pre-predecessor of $r$ (a) or to a sibling $r'$ of $r$ (b).

containing both endpoints $u$ and $v$. W.l.o.g., the endpoint starting at $r$ in a special situation is named $u$. Hence, $u$ keeps an even distance to $r$ (until a new special situation is reached).

By Claim 2, after an expected number of $O(m)$ $u$-relevant steps, both endpoints are in the same proper subtree of $T(r)$, or $u$ has left $T(r)$. First assume that $v$ finishes such a phase, i.e., $v$ leaves $T_v$ (see Fig. 6(b) and the proof of Claim 2 for the definition of $T_u$ and $T_v$). From the proof of Claim 2, we also know that $v$ cannot reach $r$ but a node in $T_u$ on the path $P$. If this happens (event $E_2$), the depth $d(r)$ increases by at least 1 and the new situation is a special situation. Now assume that $u$ finishes the phase and consider the last step. If $T(r)$ includes only three levels, the length of $P$ is at most 3. Then, if $u$ moves, it reaches the root $r$ and $\{u, v\}$ becomes a selectable edge (event $E_1$). Now we consider the case where $T(r)$ includes more than three levels and pessimistically assume that the event $E_1$ does not happen. If the length of $P$ decreases by more than 2 in the last step then $u$ enters $T_v$ and reaches a node on the path $P$ (event $E_{3\mathrm{a}}$). The depth $d(r)$ increases by at least 2 and a special situation is reached. Otherwise, $u$ visits $r$ before the last move. Given that $u$ finishes the phase in the next step, it either moves to $T_v$ with a probability of at least $1/2$ (event $E_{3\mathrm{b}}$) or leaves $T(r)$ with a probability of at most $1/2$ (event $E_4$) (Fig. 6(a)). (If $r = w$, event $E_4$ is not possible.) In the first case, $d(r)$ increases by 2 and a special situation is reached. In the second case, $d(r)$ decreases by at most 2: The endpoint $u$ may move to its pre-predecessor in the tree (Fig. 7(a)). Then a special situation is reached. Another option is that $u$ moves to some sibling of $r$, say $r'$ (Fig. 7(b)). Then the new situation is *not* a special situation. By Claim 1, after an additional waiting time of an expected number of $O(|T(r')|) = O(m)$ $u$-relevant steps, $u$ has left $T(r')$ and is back at $r$ (or even at a node on the path inside $T(r)$) such that a special situation is reached. But within this additional time, $v$ may also have left $T(r)$. If this has happened, $v$ must have visited the parent of $r$ (or even a node on the path inside $T(r')$) such that a special situation has been reached anyway. In any case, the depth $d(r)$ decreases by at most 2.

The events $E_{3\mathrm{b}}$ and $E_4$ can only occur in the same situation (where $u$ finishes the phase and $u$ visits $r$ before the last move), and, given that one of these two events happens in the next step, $\mathrm{Prob}(E_{3\mathrm{a}}) \geq 1/2$. Hence, the probability that event $E_3 := E_{3\mathrm{a}} \cup E_{3\mathrm{b}}$ finishes the phase is only larger than the probability that event $E_4$ finishes the phase. Moreover, we have shown that after an event $E_2$, $E_3$, or $E_4$ has happened, a special situation is reached within an expected number of $O(m)$ $u$-relevant steps where $d(r)$ is not decreased. □

In the proof of Claim 3 we have not only shown that one of the considered events happens within an expected number of $O(m)$ $u$-relevant steps but also that a situation is obtained where one of $u$ and $v$ is at the (new) root $r$ of the smallest subtree containing both nodes. Then, w.l.o.g. let the endpoint $u$ be at $r$.

*Claim 4.* In an expected number of $O(D^2 m)$ $u$-relevant steps a selectable edge is created.

*Proof.* Initially, we choose $w := u$ to be the root of the tree. This implies $d(r) = 0$ and we start in a special situation. When a situation is reached where $T(r)$ includes only two levels, the considered path $P$ is a selectable edge since its length must be odd. In a situation where $T(r)$ includes only three levels, the augmenting path is a selectable edge or the the next $P$-relevant step creates a selectable edge. To see this, observe that the length of $P$ then is 1 or 3. There is nothing to show if the length is 1. If the length is 3, neither $u$ nor $v$ can be at the root of $T(r)$ and the path could be lengthened by at most one edge at $u$ and $v$. By Lemma 5, the next $P$-relevant step shortens $P$. Hence, the expected number of $u$-relevant steps for the event $E_1$ is by at most 1 larger than the expected number of $u$-relevant steps for the event $d(w) \geq D - 2$. By the $O(m)$ bound in Claim 3, it is sufficient to prove an upper bound of $O(D^2)$ on the expected number of the events $E_2$, $E_3$, and $E_4$ until the event $d(w) \geq D - 2$ occurs.

We slow down this stochastic process by assuming that $E_3$ increases $d(r)$ by exactly 2, $E_4$ decreases $d(r)$ by exactly 2, and $\mathrm{Prob}(E_3 \text{ happens first}) = \mathrm{Prob}(E_4 \text{ happens first})$. We prove that the probability that $d(r) \geq D-2$ during a phase including $\lceil D^2/2 \rceil + 2D$ of our events is at least $1/2$. Then the claim follows, since we may start over and choose a new root $w$ if we were unsuccessful. If the number of $E_2$-events in the phase is at least $D - 2$, the result follows since, with probability $1/2$, event $E_3$ happens at least as often as event $E_4$. Otherwise, there are at least $D^2/2 + D$ events $E_3$ and $E_4$, and we ignore the help of $E_2$-events. Now the events $E_3$ and $E_4$ describe a symmetric random walk as analyzed in Lemma 3. Starting there in state $\ell = \lceil (D-2)/2 \rceil$ (standing for $d(r) = 0$), the expected number of steps to reach the state 0 is less than $D^2/4 + D/2$. By Markov's inequality, $D^2/2 + D$ steps are successful with probability at least $1/2$. □

Now, we can combine our arguments to prove the theorem. By Lemma 1, the expected time to create a matching is $O(m \log m)$. Since the probability that a step is $u$-relevant is $\Omega(1/m^2)$, it is sufficient to prove a bound of $O(D^2 m^2)$ on the expected number of $u$-relevant steps. For this, it is sufficient to prove a bound of $O(D^2 m)$ on the expected number of $u$-relevant steps until a larger matching is produced. This follows from the proven $O(D^2 m)$ bound on the expected number of $u$-relevant steps to create a selectable edge and Lemma 4 implying that in a situation with a selectable edge the matching is improved with a probability of at least $1/2$ within the next $O(m)$ steps. (Note that $O(m)$ is less than the expected waiting time for a $u$-relevant step in situations without selectable edges.) $\square$

## 7 A Class of Difficult Graphs for Heuristics

The graphs $G_{h,\ell}$ for odd $\ell = 2\ell' + 1$ have been introduced by Sasaki and Hajek (1988). For an illustrative description the $n := h(\ell+1)$ nodes of $G_{h,\ell}$ are positioned on a grid, i.e., $V = \{(i,j) \mid 1 \le i \le h,\ 0 \le j \le \ell\}$. Between column $j$, $j$ even, and column $j+1$ there are exactly the horizontal edges between $(i,j)$, and $(i,j+1)$. There are complete bipartite graphs between column $j$, $j$ odd, and column $j+1$. The graph $G_{3,11}$ is shown in Fig. 8. The unique perfect matching $M^*$ consists of all horizontal edges between the columns $j$ and $j+1$ for even $j$. The set of all other edges is denoted by $\overline{M}^*$. Obviously, $m = |M^*| + |\overline{M}^*| = (\ell'+1)h + \ell' h^2 = \Theta(\ell h^2)$. We prove some properties of these graphs.

**Lemma 7.** *Let $M$ be a matching in $G_{h,\ell}$. Then the following two properties hold:*

- *$M \oplus M^*$ consists exclusively of $|M^*| - |M|$ node disjoint augmenting paths called special paths, and*
- *all paths in $M \oplus M^*$ run "from left to right", more precisely, they contain at most one node of each column.*

*Proof.* The first property holds for all graphs $G$ with a unique maximum matching $M^*$. Like in the proof of Lemma 2, let $G' = (V, E')$ be the graph whose edge set is defined by $E' := M \oplus M^*$ where $M \ne M^*$ is a matching and $\oplus$ denotes the symmetric difference. The graph $G'$ consists of node-disjoint cycles and paths which are the components of $G'$. No component is an alternating cycle or a path of even length since each such cycle or path allows two maximum matchings in $G$. No alternating path of odd length starts with an $M$-edge because augmenting

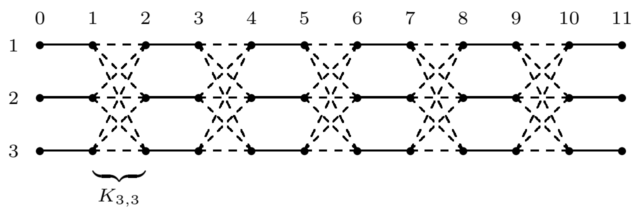


**Fig. 8.** The graph $G_{h,\ell}$, $h = 3$, $\ell = 11$, and its perfect matching.

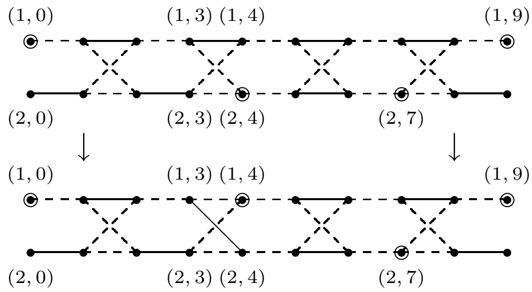


**Fig. 9.** The graph $G_{2,9}$ with a matching and the situation after an accepted 2-bit flip. Free nodes are marked by a circle.

paths for $M^*$ cannot exist since $M^*$ is a maximum matching. Hence, each of the components is an alternating path of odd length starting with an $M^*$-edge, i. e., an alternating path with respect to $M$. Since these paths are node-disjoint, there must be exactly $|M^*| - |M|$ of them.

The second property can be proven by contradiction. Assume that an augmenting path of $M \oplus M^*$ contains two nodes of column $j$. Then there are two columns $j'$ and $j'+1$ such that the path contains two adjacent edges between these columns, i. e., one belongs to $M^*$ and the other one does not belong to $M^*$. This is a contradiction since $M^*$ includes either all edges or no edge between two neighbored columns. $\square$

If $P$ is a special path with respect to $M$ whose endpoints are not in the first or last column and there is no selectable edge, there are 2 2-bit flips which shorten $P$ by 2 and $2h$ 2-bit flips that lengthen $P$ by 2. This makes $G_{h,\ell}$ difficult for heuristics working mostly locally. One idea is to focus the analysis on the length of the longest special path. However, there are accepted 2-bit flips which lengthen a special path but turn it into a non-special one. As an example, consider the following augmenting paths in the situation depicted in the upper part of Fig. 9 where $|M^*| - |M| = 2$. The first two are the special paths:

- $(1, 0), \ldots, (1, 3), (1, 4), \ldots, (1, 9)$ of length 9,
- $(2, 4), \ldots, (2, 7)$ of length 3,
- $(2, 4), (1, 3), (1, 4), \ldots, (1, 9)$ of length 7.

After an accepted 2-bit flip we obtain the situation in the lower part, where we again consider three augmenting paths where the first two ones are special:

- $(1, 0), \ldots, (1, 3), (2, 4), \ldots, (2, 7)$ of length 7,
- $(1, 4), \ldots, (1, 9)$ of length 5,
- $(1, 4), (1, 3), (2, 4), \ldots, (2, 7)$ of length 5.

This is a 2-bit flip which lengthens the special path from $(2, 4)$ to $(2, 7)$ in the upper graph but the lengthening is no longer a special path in the lower graph. The total length of all special paths is unchanged.

The main parameter of our analyses is the number $g = g(M)$ of edges chosen by $M$ which do not belong to the perfect matching. This $\overline{M}^*$-edges counted by $g$ can be considered as the "bad" edges of $M$. It is necessary to decrease the $g$-value to zero in order to obtain the perfect matching. The $g$-value is closely related to the total length of all special paths excluding selectable edges.

## 8 Exponential Expected Optimization Time for RLS and the (1+1) EA

We analyze the situation when the current search point $s$ describes an almost perfect matching, i.e., a matching of size $|M^*| - 1$. First, we estimate the probability of obtaining an almost perfect matching with an augmenting path of length $\ell$. For almost perfect matchings, by Lemma 7, there is always a unique augmenting path that is special (see Fig. 10). In a second step, we estimate the expected time to obtain a perfect matching from an almost perfect matching with an augmenting path of length $\ell$. Finally, we estimate the probability to obtain an almost perfect matching before the perfect one.

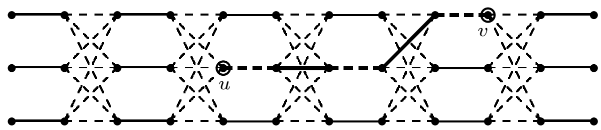


**Fig. 10.** An almost perfect matching and its augmenting path. The free nodes are marked by a circle.

In Section 7 we have shown that a special path between $u$ and $v$ allows $2h$ lenghtenings and 2 shortenings by 2-bit flips if $u$ and $v$ are not in the first or last column. If both endpoints are in outer columns, it is no problem that the augmenting path cannot be lengthened. If only one endpoint is in an outer column there are $h$ lengthenings. The game is still unfair if $h \geq 3$. By these arguments, we need some extra arguments for the case $h = 2$. The case $h = 2$ is of special interest since $G_{2,\ell}$ are the only planar graphs in the considered class and the maximal node degree is 3.

The idea behind the analysis of the last phase of the search is the following. Starting with an almost perfect matching, with overwhelming probability, $O(m^3)$ steps are enough to obtain either the perfect matching or an augmenting path of maximal length, which is $\ell$ by Lemma 7. We estimate the probabilities which of these two events happens first. If the augmenting path has reached length $\ell$, we prove that it is very likely to need exponentially many steps to obtain the perfect matching. To obtain this result it is required that $\ell$ is a polynomial in $m$. We are mainly interested in the case $2 \leq h \leq \ell$ implying that $\ell = \Omega(m^{1/3})$. Then, $2^\ell$ is exponential in $m$ and in a phase of this length, it is likely that the (1+1) EA performs some steps which change the situation globally. Therefore, the analysis is always easier for RLS.

The basis of the analysis of the first part is the gambler's ruin problem (see, e. g., Feller (1968)). Alice owns $A$ \$ and Bob $B$ \$. They play a coin-tossing game with a probability of $p \neq 1/2$ that Alice wins a round in this game and receives a dollar from Bob. Otherwise Alice has to pay a dollar to Bob. The game is finished when one player is ruined. Alice's probability of winning the game, i. e., Bob being ruined, is $(1-t^A)/(1-t^{A+B})$, where $t := (1-p)/p$.

The game here is based on the length of the augmenting path. If we have obtained a selectable edge, it is likely to find the perfect matching in the next step.

**Lemma 8.** *For RLS and the (1+1) EA starting with an almost perfect matching with a $g$-value of 0, the following holds. The probability of reaching an almost perfect matching with a $g$-value of at least* 1 *is* $\Theta(h/m)$.

*Proof.* For an almost perfect matching, $g = 0$ implies that the augmenting path is a selectable edge. To improve the matching, it is sufficient that only the selectable edge flips and it is necessary that this edge flips. Therefore, the probability of creating the perfect matching is $\Theta(1/m)$. To increase $g$, it is sufficient that one of the $h$ or $2h$ edge pairs lengthening the augmenting path flips. (If also the selectable edge flips, the path moves to another position. Then, at least a matching edge has to flip and additionally one of the $h$ or $2h$ pairs lengthening this new augmenting path.) Hence, the probability to reach a situation where $g \geq 1$ equals $\Theta(h/m^2)$. □

Now we investigate RLS. If the $g$-value is at least 1, we consider the coin-tossing game. Alice wins if the $g$-value increases. Alice's probability to win a round equals $2h/(2h+2)$ if none of the endpoints of the augmenting path is in column 0 or column $\ell$. Otherwise, the probability equals $h/(h+2)$. For $h \geq 3$, this value is larger than $1/2$ and we can apply the result on the gambler's ruin problem. For $h = 2$, this value is $h/(h+2) = 1/2$ and we have to use the fact that there is a good chance that the endpoint of the augmenting path in column 0 or column $\ell$ leaves this column. If $g_0 = 1$, then $g_1 = 2$ with probability at least $1/2$. Afterwards, we investigate pairs of steps changing the length of the augmenting path. If one step of a pair is length increasing and the other is length decreasing, the $g$-value is the same as before. Otherwise, $g$ is increased or decreased by 2. A simple case inspection shows that, under the condition that one player wins two rounds, the probability that Alice wins is at least $6/11$. Again we obtain a game in favor of Alice.

**Lemma 9.** *For RLS starting with an almost perfect matching with a $g$-value of $g_0$, the probability of constructing an augmenting path of maximal length before the perfect matching is*

$$\textit{at least } 1-(2/h)^{g_0} \textit{ if } h \geq 3 \textit{ and } g_0 \geq 1, \textit{ and}$$
$$\textit{at least } 1-(5/6)^{\lfloor g_0/2 \rfloor} \textit{ if } h = 2 \textit{ and } g_0 \geq 2.$$

*Proof.* We analyze the coin-tossing game where $A := g_0$, and $B := \ell' - g_0$. We pessimistically consider the matching improved if the $g$-value equals 0, i.e., if Alice loses the game. Let the probability $p(h) = h/(h+2)$, if $h \geq 3$, and $p(2) = 6/11$. Then $t(h) = 2/h$, if $h \geq 3$, and $t(h) = 5/6$ if $h = 2$. The probability that Alice wins the game if $h \geq 3$ equals $(1 - t(h)^{g_0})/(1 - t(h)^{\ell'})$. Since $0 < t(h) < 1$, this probability is at least $1 - t(h)^{g_0}$. If $h = 2$ and $\ell_0 \geq 2$, we obtain the result by the same arguments but we have to observe that steps change $g$ by 2 instead of 1. □

For $h = 2$ and $g_0 \geq 1$, with probability at least 1/2, we reach the situation $g = 2$ before $g = 0$ and, then, we can apply Lemma 9. The success probabilities are close to 1 if $g_0$ is increasing with $\ell$, or if $h$ is not a constant.

For the (1+1) EA, we have to estimate the probabilities of steps where many flipping bits influence the augmenting path. In order to simplify the analysis we interpret the following event as a loss of Alice of the whole game. At least the leftmost $i \leq 4$ and the rightmost $j \geq 4 - i$ edges of the augmenting path flip. The probability of this event is bounded above by $O(1/m^4)$. Now, the only possibility of decreasing the $g$-value by 1 is to flip exactly the two leftmost or the two rightmost edges of the augmenting path. The probability of this event equals $2(1/m)^2(1 - 1/m)^{m-2}$. This leads to the same probabilities as in Lemma 9 but we have to take into account the probability of $\Theta(1/m^3)$ to turn a search point with a short augmenting path of length 3 into the perfect matching. Now we are able to prove the following result. We remark that these bounds are essentially the same as in Lemma 9.

**Lemma 10.** *For the (1+1) EA starting with an almost perfect matching with a g-value of $g_0$, the probability of constructing an augmenting path of maximal length before the perfect matching is*

$$\text{at least } 1 - O(1/m) - \big((2/h) + O(1/m)\big)^{g_0} \text{ if } h \geq 3 \text{ and } g_0 \geq 1, \text{ and}$$
$$\text{at least } 1 - O(1/m) - \big((5/6) + O(1/m)\big)^{\lfloor g_0/2 \rfloor} \text{ if } h = 2 \text{ and } g_0 \geq 2.$$

*Proof.* Since we pessimistically consider the event of a $g$-value of 0 as the event that the perfect matching is created, we can include the event that an augmenting path of length 3 is flipped in the event that Bob wins a round. The probability that a step changes the augmenting path is $\Theta(1/m^2)$. The probability of flipping an augmenting path of length 3 is $\Theta(1/m^3)$. Therefore, it is sufficient to increase the values of $t(h)$ in the proof of Lemma 9 by $O(1/m)$.

In addition, there is a probability of $O(1/m^4)$ for each step that Bob wins the whole game because $g$ changes by more than 1. We claim that, with overwhelming probability, the game is finished within $O(m^3)$ steps. The probability of a step with probability $O(1/m^4)$ in such a phase is $O(1/m)$. This is taken into account by the term "$-O(1/m)$." The essential argument proving the claim is that $p(h) - (1/2) = \Omega(1)$. By Chernoff bounds, for some constant $c$ and $cm$ coin tosses, the probability that Alice wins at least $\ell'$ more rounds than Bob is overwhelming. For some constant $c'$, the probability of $cm$ relevant steps within $c'm^3$ steps is overwhelming, too. This proves the lemma. □

For RLS, it is rather easy to prove that it is very likely to need exponentially many steps to construct the perfect matching if we start with an augmenting path of maximal length.

**Lemma 11.** *Starting with an almost perfect matching and an augmenting path of maximal length, the probability that RLS finds the perfect matching within* $2^{c\ell}$ *steps,* $c > 0$ *an appropriate constant, is bounded above by* $2^{-\Omega(\ell)}$.

*Proof.* We can essentially apply the arguments of the proof of Lemma 9. For $h \geq 3$, we have $g_0 = \ell'$. In order to reach a $g$-value of 0 we have to reach the value $\lceil \ell'/2 \rceil$. Starting there, we have an unfair game between Alice and Bob and Bob's winning probability is bounded above by $2^{-c'\ell}$ for some constant $c' > 0$. The game is repeated until Bob wins for the first time. The probability of winning at least once in $2^{c\ell}$ games is bounded above by $2^{(c-c')\ell} = 2^{-\Omega(\ell)}$ if we choose $c < c'$ small enough. Analogously for $h = 2$. □

The idea behind the search operator of the (1+1) EA is to allow big changes (with small probability). In order to prove that the search needs exponentially many steps we cannot exclude steps with "big changes." The essential argument is that the drift to lengthen augmenting paths is strong enough to obtain the proposed bound. The following drift theory goes back to Hajek (1982). We apply a result due to He and Yao (2001). Analyzing their proof, it follows immediately that they have even proved a stronger result than stated, namely a result on the success probability and not only the expected waiting time for a success. We state this result in Theorem 7.

**Theorem 7.** *Let* $X_0, X_1, X_2, \ldots$ *be the random variables describing a Markov process and let* $g\colon \mathbb{R} \to \mathbb{R}_0^+$, $0 \leq a(\ell) < b(\ell)$, $\lambda > 0$, $D \geq 1$, *and* $r(\ell) > 0$ *a polynomial. Moreover, assume that*

$$g(X_0) \geq b(\ell) \text{ with probability } 1,$$

$$b(\ell) - a(\ell) = \Omega(\ell),$$

$$E\big(e^{-\lambda(g(X_{t+1})-g(X_t))} \mid X_t,\ a(\ell) < g(X_t) < b(\ell)\big) \ \leq\ 1 - 1/r(\ell), \text{ and}$$

$$E\big(e^{-\lambda(g(X_{t+1})-b(\ell))} \mid X_t,\ b(\ell) \leq g(X_t)\big) \ \leq\ D.$$

*Let* $T$ *be the smallest* $t$ *where* $g(X_t) \leq a(\ell)$. *The probability that* $T \leq B$ *is bounded above by* $D \cdot B \cdot e^{\lambda(a(\ell)-b(\ell))} \cdot r(\ell)$.

Since $\lambda(a(\ell) - b(\ell)) = -\Omega(\ell)$ and $p(\ell)$ is a polynomial, this bound is exponentially small for $B = 2^{c\ell}$ if $c > 0$ is a small enough constant.

**Lemma 12.** *Starting with an almost perfect matching and an augmenting path of maximal length, the probability that the (1+1) EA finds the perfect matching within* $2^{c\ell}$ *steps,* $c > 0$ *an appropriate constant, is bounded above by* $2^{-\Omega(\ell)}$.

*Proof.* First, we assume $h \geq 3$ and apply Theorem 7. Our initial Markov process is the (1+1) EA on the matching problem for $G_{h,\ell}$. We apply Theorem 7 for $b(\ell) = \ell', a(\ell) = 1$, and a time bound $B = 2^{c\ell}$. The first two conditions of Theorem 7 are fulfilled. For the search point $s_t$ in the $t$th step let $g(s_t)$ be the $g$-value according to $s_t$. Now we only have to check certain drift conditions which require upper bounds on the expected value of $e^{-\lambda(g(s_{t+1})-g(s_t))}$ for some $\lambda > 0$ given the search point $s_t$ at time $t$. This enables us to concentrate on the probabilities $p_j(s)$ that $g(s_{t+1}) - g(s_t)$ takes some value $j \in \mathbb{Z}$ given $s_t = s$. We only increase the considered expected value if we shift some part of the probability $p_j(s)$ to $p_{j'}(s)$ where $j' < j$. Everything of $p_j(s)$, $j \geq 2$, and a part of $p_1(s)$ is shifted to $p_0(s)$, i.e., we replace $p_1(s)$ by a lower bound. Moreover, a part of $p_0(s)$ is shifted to $p_{-j}(s)$, $j \geq 1$, i.e., we replace $p_{-j}(s)$ by upper bounds.

*Claim 1.* Let $s$ be a search point describing an almost perfect matching with a $g$-value of $1 < g < \ell'$. Let

$$\begin{aligned}
p_1(g) &:= h \cdot (1/m)^2(1-1/m)^{m-2},\\
p_{-1}(g) &:= 2 \cdot (1/m^2)(1-1/m)^{m-2} + 3 \cdot (1/m)^4,\\
p_{-j}(g) &:= (j+1)(1/m)^{2j}, \text{ if } 1 < j < g, \text{ and}\\
p_{-g}(g) &:= (1/m)^{2g+1}.
\end{aligned}$$

Then $p_1(g)$ is a lower bound for the probability to increase the $g$-value by 1, and $p_{-j}(g)$ is an upper bound on the probability to decrease the $g$-value by $j$.

*Proof.* Since there are always at least $h$ possibilities to lengthen the augmenting path, we estimate this probability from below by

$$p_1(g) := h \cdot (1/m)^2(1-1/m)^{m-2}.$$

There is the special case of decreasing the $g$-value to 0. Then exactly the $2g+1$ edges of the augmenting path have to flip. This probability can be estimated from above by

$$p_{-g}(g) := (1/m)^{2g+1}.$$

Finally, we need an upper bound $p_{-j}(g)$ on the probability of decreasing the $g$-value by $j$ in one step. It is necessary to flip the $2k$ leftmost edges and the $2(j-k)$ rightmost edges of the augmenting path for some $k \in \{0, \ldots, j\}$. Hence,

$$p_{-j}(g) := (j+1)(1/m)^{2j}$$

is a correct upper bound. For $p_{-1}(g)$, we need a better bound which also for small values for $h$ is at least by a constant factor smaller than $p_1(g)$. It is sufficient to argue as follows. There are exactly two possibilities by flipping exactly two edges, and otherwise we have to flip at least the $2k$, $0 \leq k \leq 2$, leftmost edges and the $4-2k$ rightmost edges of the augmenting path. Hence,

$$p_{-1}(g) := 2 \cdot (1/m)^2(1-1/m)^{m-2} + 3 \cdot (1/m)^4.$$

is a correct upper bound. □

Now we increase the probability of reaching a $g$-value of at most 0 by not counting the steps not changing the $g$-value. The transition probabilities of the new process are $q_j(g) := p_j(g)/(1-p_0(g))$, if $j \neq 0$, and $q_0(g) := 0$. We have

$$1 - p_0(g) = (h+2)(1/m^2)(1-1/m)^{m-2} + O(1/m^3)$$

since $p_{-g}(g) = O(1/m^3)$ in the considered situations where $g(s_t) \geq 1$.

To check the third condition we have to estimate

$$e^{-\lambda} q_1(g) + e^{\lambda} q_{-1}(g) + \sum_{j \geq 2} e^{j\lambda} q_{-j}(g). \qquad (*)$$

We prove that for some constants $\lambda > 0$ and $\delta > 0$ small enough $e^{-\lambda} q_1(g) + e^{\lambda} q_{-1}(g) \leq 1 - \delta$ and $\sum_{j \geq 2} e^{j\lambda} q_{-j}(g) = o(1)$ which implies the third condition. The second claim is easy to show since $q_{-j}(g) = O(jm^{2-2j})$, if $1 < j < g$, and $q_{-g}(g) = O(m^{1-2g})$. Hence,

$$\sum_{j \geq 2} e^{j\lambda} q_{-j}(i) \;=\; O\Big(e^{g\lambda} m^{1-2g} + \sum_{j \geq 2} e^{j\lambda} j m^{2-2j}\Big) \;=\; o(1).$$

Using Taylor's expansion, we obtain for some $\alpha > 0$ and $\lambda$ small enough,

$$e^{-\lambda} \leq 1 - \lambda + \alpha\lambda^2 \qquad \text{and} \qquad e^{\lambda} \leq 1 + \lambda + \alpha\lambda^2.$$

This implies,

$$\begin{aligned}
&e^{-\lambda} q_1(g) + e^{\lambda} q_{-1}(g) \\
&\leq\; (1 - \lambda + \alpha\lambda^2) q_1(g) + (1 + \lambda + \alpha\lambda^2) q_{-1}(g) \\
&\leq\; (q_1(i) + q_{-1}(g)) - \lambda(q_1(g) - q_{-1}(g)) + \alpha\lambda^2 (q_1(g) + q_{-1}(g)) \\
&\leq\; 1 - \lambda(q_1(g) - q_{-1}(g)) + \alpha\lambda^2.
\end{aligned}$$

Finally, if $h \geq 3$,

$$q_1(g) - q_{-1}(g) \;=\; \frac{(h-2)(1/m^2)(1-1/m)^{m-2} - 3(1/m^4)}{(h+2)(1/m^2)(1-1/m)^{m-2} - O(1/m^3)} \;\geq\; \beta \qquad (**)$$

for some constant $\beta > 0$ and $m$ large enough. Hence, it is sufficient to show that

$$1 - \lambda\beta + \alpha\lambda^2 \leq 1 - \delta.$$

It is easy to choose $\lambda > 0$ and $\delta > 0$ to fulfill this property.

In a situation according to the last condition, the $g$-value cannot increase. Hence, $p_j(g) = 0$ is the only correct lower bound for the probability of increasing $g$ by $j \geq 1$. Our upper bounds $p_{-j}(g)$ in Claim 1 for decreasing steps remain valid. Now, $p_0(g)$ equals $1 - 2 \cdot (1/m)^2(1-1/m)^{m-2} - O(1/m^3)$ and leads to new expressions for the probabilities $q_{-j}(g)$. Formally, we obtain the same sum as $(*)$ but the first term vanishes since $p_1(i) = 0$ implies $q_1(i) = 0$. The second term is bounded by the constant $e^{\lambda}$. The remaining terms of $(*)$ can be bounded by $O(1/m)$ in the same way as before since $1 - p_0(g) = \Omega(1/m^2)$ remains valid. Hence, the sum is bounded above by $D := e^\lambda + 1$ for $m$ large enough.

Now we investigate the case $h = 2$. The inequality $(**)$ is no longer true. We use the trick of combining a special 2-bit flip with the following relevant step, i. e., the next step changing the path. It is possible that the special 2-bit flip produces an augmenting path of length 1. This is not essential since we may choose $a(\ell) = \lfloor \ell'/2 \rfloor$ and consider the matching improved if a $g$-value of less than $\ell'/2$ is reached. Then, the probability that the relevant step following a special 2-bit flip is not a special 2-bit flip is bounded by $O(1/m^2)$. In this case, we rate the special 2-bit flip as a decrease of $g$ by 2. In the other case, we have a pair of two special 2-bit flips which follow one after the other. This implies (see the proof of Lemma 9) that the probability of two increasing steps minus the probability of two decreasing steps is bounded below by a positive constant. This leads to a counterpart of $(**)$ for $q_2(g) - q_{-2}(g)$ and we can complete the proof in the same way as before. □

We summarize our results.

**Theorem 8.** *Starting with an almost perfect matching and an augmenting path of length $2g_0 + 1$, the probability that the (1+1) EA finds the perfect matching within $2^{c\ell}$ steps, $c > 0$ an appropriate constant, is bounded above by*

$$O(1/m) + ((2/h) + O(1/m))^{g_0} \quad \text{if } 3 \le h \le \ell \text{ and } g_0 \ge 1, \text{ and}$$
$$O(1/m) + ((5/6) + O(1/m))^{\lfloor g_0/2 \rfloor} \quad \text{if } h = 2 \text{ and } g_0 \ge 2.$$

*For RLS, the bounds $2^{-\Omega(\ell)} + (2/h)^{g_0}$ resp. $2^{-\Omega(\ell)} + (5/6)^{\lfloor g_0/2 \rfloor}$ hold.*

We return to the question whether an almost perfect matching will be reached.

**Lemma 13.** *If the (1+1) EA or RLS do not start with the perfect matching, an almost perfect matching is constructed before the perfect matching with a probability of $\Omega(1/h)$.*

*Proof.* Let $M$ denote the set of edges selected by the current search point, and let $d := |M \oplus M^*|$ denote the Hamming distance to $M^*$. We investigate the situations when $M$ is neither an almost perfect nor the perfect matching; this includes the case that $M$ is not even a matching. Then, any step producing an almost perfect matching will be accepted.

For the (1+1) EA, the probability to produce $M^*$ in the next step is $\Theta(1/m^d)$. We argue that this probability is at most by a factor of $O(h)$ larger than the probability to produce an almost perfect matching in the next step. If $M \oplus M^*$ contains at least one $M^*$-edge, this edge is not included in $M$ and the step where everything works like in the step creating the perfect matching except for the $M^*$-edge produces an almost perfect matching. The probability $\Theta(1/m^{d-1})$ of this step is even larger than the probability of the step creating $M^*$. If $M \oplus M^*$ contains no $M^*$-edge, all $M^*$-edges are included in $M$ and there are $|M^*|$ possibilities to produce an almost perfect matching by additionally flipping an $M^*$-edge. Their probability is $\Theta(|M^*|/m^{d+1}) = \Theta(1/(hm^d))$.

For RLS, a necessary event is a situation where $d \leq 2$. We argue that in any situation where $d = 1$, the next step produces $M^*$ with a probability that is at most by a factor $O(h)$ larger than the probability that it produces an almost perfect matching. In situations where $d = 2$, the first probability is even smaller than the last probability if we consider the next two steps.

If $d = 1$, we are only interested in the case where $M$ is a superset of $M^*$ because otherwise $M$ was almost perfect. Let $M = M^* \cup \{e\}$ implying that $e$ is an $\overline{M}^*$-edge. The next step produces $M^*$ with probability $\Theta(1/m)$. If $e$ and some other edge of $M$ flip, an almost perfect matching is obtained. This happens with probability $\Theta(|M^*|/m^2) = \Theta(1/(hm))$.

If $d = 2$, a necessary event to produce $M^*$ is that each of the two edges in $M \oplus M^*$ flips at least once in the next two steps. The probability of this event is $\Theta(1/m^2)$. If $M \oplus M^*$ contains two $M^*$-edges, both are selectable and the first step produces an almost perfect matching with a probability $\Theta(1/m)$ by flipping only one of them. If $M \oplus M^*$ contains one $M^*$-edge and one $\overline{M}^*$-edge, the first step removes the latter edge from $M$ with probability $\Theta(1/m)$ and produces an almost perfect matching. If $M \oplus M^*$ contains two $\overline{M}^*$-edges, then $M = M^* \cup \{e_1, e_2\}$ is a non-matching where $e_1$ and $e_2$ are $\overline{M}^*$-edges. Any step flipping $e_1$ and an arbitrary $M^*$-edge in the first step will be accepted since the sum of the penalties decreases by at least $r \geq m + 1$. If the second step flips $e_2$, an almost perfect matching is obtained. The probability of these events is $\Theta((|M^*|/m^2) \cdot (1/m)) = \Theta(1/(hm^2))$. $\square$

Sasaki and Hajek (1988) proved that the expected optimization time of simulated annealing for $G_{h,\ell}$ is exponential if the process starts with the empty matching. Sasaki (1991) proved that there exists a starting point where the expected optimization time for the Metropolis algorithm is exponential. For RLS and the (1+1) EA we obtain much stronger results. Even for the smallest values of $h$ and $g_0$ allowed in Theorem 8, the probability to need more than $2^{c\ell}$ steps is $\Omega(1)$. If our heuristics produce an almost perfect matching where $g_0 = 0$, the next relevant step increases the path length with a probability of $\Omega(1/m)$ (Lemma 8). The second relevant step is increasing with probability at least $1/2$. By this observation and Lemma 13, we obtain that the $2^{\Omega(\ell)}$-bound of Theorem 8 holds with a probability $\Omega(1/(hm))$ if we start with any search point which is not the optimum. If $h \leq \ell$ then $\ell = \Omega(m^{1/3})$, and if $h$ is a constant then $\ell$ is even linear in $m$. Hence, for $2 \leq h \leq \ell$, we obtain an exponential lower bound of $2^{\Omega(\ell)} = 2^{\Omega(m^{1/3})}$ for the expected optimization time if the search does not start with the optimum. This leads to the following theorem summarizing the result of this section.

**Theorem 9.** *For $G_{h,\ell}$, $2 \leq h \leq \ell$, the expected optimization time of RLS and the (1+1) EA is $2^{\Omega(\ell)}$ if the initial search point is not the perfect matching.*

As the graph is unknown, the probability not to start with the perfect matching is large, and, for typical choices of the initial distribution, it is overwhelming. In particular, this probability is 1 if we start with the empty set of edges and $1 - 2^{-m}$ if the initial search point is chosen uniformly at random.

## 9 Exponential Optimization Time with Overwhelming Probability for RLS and the (1+1) EA

These improved results can be proved only if $h = \omega(\log m)$ and $h \leq \ell - 2$. Our investigations assume that the initial search point is a matching with a $g$-value of at least $h/2$. At the end we prove that it is quite likely to obtain such a search point. The result for RLS is much easier to obtain than the result for the (1+1) EA and it clarifies many ideas.

**Theorem 10.** *Let $h = \omega(\log m)$ and $h \leq \ell - 2$. If the initial search point describes a matching with a $g$-value of at least $h/2$, there is a constant $c > 0$ such that the probability that RLS finds the perfect matching within $2^{ch}$ steps is $2^{-\Omega(h)}$.*

*Proof.* First, we classify the relevant steps, i. e., accepted steps changing the search point.

*Claim 1.* All relevant steps of RLS can be assigned to one of the following types.

**Type-1** steps flip only selectable edges.
**Type-2** steps flip a selectable edge and a matching edge (implying that these edges are not adjacent).
**Type-3** steps flip a matching edge and an adjacent free edge, touching three adjacent columns.
**Type-4** steps flip a matching edge and an adjacent free edge between the same columns.

*Proof.* It is easy to see that 1-bit flips are only accepted if they concern a selectable edge (Type-1). Steps flipping two bits are accepted if two non-adjacent selectable edges are chosen (Type-1), if a selectable edge and a matching edge are chosen (Type-2), or if a matching edge and an adjacent free edge are chosen. In the last case, the involved free edge has to be incident upon a free node (Type-3 or Type-4). Since there are no edges with both endpoints in the same column, each accepted step belongs to one and only one of these four types. □

An important observation is that Type-4 steps do not change the $g$-value. Type-2 steps can change the $g$-value by at most 1 and there can be many more $g$-decreasing steps than increasing ones. To control the effect of Type-2 steps we consider them as harmful only if $g \leq h/2$.

*Claim 2.* The probability of more than $h/8$ Type-2 steps decreasing the $g$-value in situations where $g \leq h/2$ is bounded by $2^{-\Omega(h)}$.

*Proof.* Our goal is to show that the negative effect of Type-2 steps is small if $g \leq h/2$. The idea is to compare them to Type-1 steps. Note that Type-1 steps always increase the matching and never decrease $g$. Type-2 steps and Type-1 steps can only occur in situations with selectable edges, which obviously cannot occur after $|M^*| = (\ell' + 1)h$ Type-1 steps. The probability of a Type-1 step

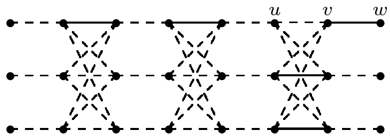


**Fig. 11.** A situation with only one possibility to increase $g$ at $u$ in a Type-3 step.

selecting only a specific selectable edge equals $1/(2m)$. The probability of a Type-2 step selecting the same selectable edge and one of the $g$ selected $\overline{M}^*$-edges equals $g/(m(m-1))$. Hence, if one of the two possibilities occurs, the conditional probability of the first possibility is $\Omega(1)$ and the conditional probability of the second possibility equals $\Theta(g/m)$. Hence, some $\Theta(\ell h)$ steps flipping a selectable edge contain $|M^*|$ Type-1 steps with a probability $1-2^{-\Omega(\ell h)}$. This implies that there are at most $O(\ell h)$ Type-1 or Type-2 steps. For Type-2 steps decreasing $g$, we are only interested in situations where $g \leq h/2$. Their expected number is $O(gh\ell/m) = O(1)$, and, by Chernoff bounds, this random number is less than $h/8$ with a probability of $1 - 2^{-\Omega(h)}$. □

Hence, it is enough to prove that, with overwhelming probability, Type-3 steps in situations where $g \leq h/2$ never decrease the $g$-value by $h/4$ in the considered time interval. Type-3 steps shorten or lengthen special paths with a much larger probability of lengthenings. It is easy to apply results on the gambler's ruin problem to complete the proof. A Type-3 step touching the free node $u$ decreases $g$ if the special path starting at $u$ is shortened, and it increases $g$ if the special path is lengthened. If the considered endpoint $u$ is not in the first or last column, there is at least one possibility of increasing $g$. W.l.o.g. let $u$ be a node in column $j$, and $j$ is odd (see Fig. 11). Since $u$ is free, there can be at most $\min\{h-1, g\}$ matching edges between column $j$ and column $j+1$. Hence, $u$ has a neighbor $v$ in column $j+1$ which is not incident upon a matching edge between the columns $j$ and $j+1$. Flipping $\{u, v\}$ and $\{v, w\}$ increases $g$ (see Fig. 11). If $v$ is also free, then $\{u, v\}$ is selectable and flipping only $\{u, v\}$ increases $g$ and the matching size. Such a 1-bit flip is more likely than the special 2-bit flip considered before, and we may pessimistically assume that $v$ is not free. In a situation where $u$ is not in the last or first column and $g \leq h/2$, there are at least $h - g \geq h/2$ possibilities to increase $g$ at $u$. Since $h < \ell - 2$, at most one endpoint of each special path is in the first or last column if $g \leq h/2$. Then, we have at least $h/2$ possibilities to increase $g$ and only 2 possibilities to decrease $g$ per special path by Type-3 steps or Type-1 steps flipping only selectable $\overline{M}^*$-edges.

Now we can analyze the stochastic process of the $g$-value. After $g$ has reached a value of at least $h/2$, we do not care for situations where $g$ is large and wait for a point of time where $g$ is at most $h/2$. In order to reach 0, a value of $3h/8$ must be reached first. Starting there, we first consider only Type-3 steps and Type-1 steps flipping only selectable $\overline{M}^*$-edges. The conditional probability of a step increasing $g$ is $1 - O(1/h)$. This unfair game can be described by the gambler's ruin problem, see Section 8.

Starting with a $g$-value of $3h/8$ and $p = 1 - O(1/h)$, the probability to reach $h/4$ ("Alice is ruined.") before $h/2$ ("Alice wins.") is $2^{-\Omega(h)}$ and the probability of this event occurring once within $2^{ch}$ repetitions is $2^{-\Omega(h)}$ if the constant $c$ is small enough. By taking into account decreasing Type-2 steps (see Claim 2), the true $g$-value may be only by $h/8$ smaller than assumed, regardless of the number of repetitions. □

We have to work harder to obtain the same results for the (1+1) EA.

**Theorem 11.** *Let $h = \omega(\log m)$ and $h \leq \ell - 2$. If the initial search point describes a matching with a $g$-value of at least $h/2$, there is a constant $c > 0$ such that the probability that the (1+1) EA finds the perfect matching within $2^{ch}$ steps is $2^{-\Omega(h)}$.*

*Proof.* In the analysis of RLS we have bounded the bad effects of Type-2 steps by the good effects of Type-1 steps. Here we combine these two types to so-called *risky* steps flipping at least one selectable $M^*$-edge. We consider a situation with a selectable $M^*$-edge. On the one hand, there is a probability of $\Omega(g/m^2)$ to decrease the $g$-value by flipping the selectable $M^*$-edge and some selected $\overline{M}^*$-edge. On the other hand, with a probability of $\Theta(1/m)$ only the selectable edge flips and the matching increases. The number of the last kind of risky steps is bounded by $|M^*|$.

*Claim 1.* With a probability of $1 - 2^{-\Omega(h)}$, the total decrease of the $g$-value by risky steps where $g \leq h/2$ is at most $h/8$, and there is no step among the first $2^{ch}$ steps decreasing the $g$-value by at least $h/8$.

*Proof.* For the first part of the claim, we work under the conditions that the considered steps are risky and $g \leq h/2$. Then, the matching increases by at least 1 if no further edges flip. This happens with a probability of at least $(1 - 1/m)^{m-1} \geq 1/e$. By Chernoff bounds, $2e\ell' h$ risky steps increase the matching at least $|M^*| = (\ell' + 1)h$ times with a probability of $1 - 2^{-\Omega(\ell h)}$. This bounds the number of risky steps. If the $g$-value decreases by some number, the same number of selected $\overline{M}^*$-edges has to flip. In any of the considered risky steps, at most $h/2$ selected $\overline{M}^*$-edges can flip and the probability that a specific $\overline{M}^*$-edge flips is $1/m$. Hence, the expected number of flipping $\overline{M}^*$-edges in $2e\ell' h$ risky steps where $g \leq h/2$ is at most $2e\ell' h \cdot h/2 \cdot 1/m = O(1)$. By Chernoff bounds, the random number of flipping $\overline{M}^*$-edges is at most $h/8$ with a probability of $1 - 2^{-\Omega(h)}$ (for $h$ large enough).

The probability that a phase of length $2^{ch}$ includes a step decreasing the $g$-value by at least $h/8$ can be estimated in the following way. A necessary event is that at least $h/8$ selected $\overline{M}^*$-edges out of at most $|M^*|$ selected edges flip. The probability of the last event is upper bounded by

$$\binom{(\ell'+1)h}{h/8} \frac{1}{m^{h/8}} \leq \left(\frac{(\ell'+1)h}{m}\right)^{h/8} = 2^{-\Omega(h \log h)}.$$

Within the phase of $2^{ch}$ steps, such an event occurs only with a probability of $2^{-\Omega(h \log h)}$. □

This claim implies the following conclusions. If $g$ underruns $h/2$, there is a point of time where $g$ is between $h/2$ and $3h/8$, and we can ignore all steps where $g > h/2$. Altogether, we end up again with the following problem. Starting with a $g$-value between $h/2$ and $3h/8$, we only have to investigate non-risky steps, i.e., steps not flipping selectable $M^*$-edges. We have to show that these steps decrease $g$ by at most $h/8$. Then, taking into account the risky steps, the $g$-value will always be at least $h/8$.

We have to be careful since we are investigating phases of exponential length. The probability that certain steps lead to "quite global" changes is not small. We investigate the stochastic process of the (1+1) EA whose state space is the set of search points describing matchings. The transition probabilities of this process depend on many aspects of the current search point. To prove that it is quite unlikely to reach a state with a $g$-value of at most $h/4$ from a state with a $g$-value of at least $3h/8$ within the considered time interval and ignoring the risky steps we again apply the drift theorem, namely Theorem 7. Now we proceed in the same way as in the proof of Lemma 12.

*Claim 2.* Let $s$ be a search point describing a matching such that $k := k(s)$ is the number of special paths of length at least 3 and $g := g(s) \leq h/2$. Let

$$\begin{aligned} p_1(k) &:= (kh)/(2em^2), \\ p_{-1}(k) &:= (2k+2)/m^2, \\ p_{-2}(k) &:= 1/(m^2\ell h), \text{ and} \\ p_{-j}(k) &:= 1/m^j, \text{ for } j \geq 3. \end{aligned}$$

For steps not flipping a selectable $M^*$-edge, $p_1(k)$ is a lower bound on the probability to increase the $g$-value by 1, and $p_{-j}(k)$, $j \geq 1$, is an upper bound on the probability to decrease the $g$-value by $j$.

*Proof.* In the following, special paths of length at least 3 are called *long special paths.* With the arguments as in the proof of Theorem 10, there are at least $h/2$ possibilities per long special path to increase $g$ by 1 by either flipping two non-selectable edges or a single selectable $\overline{M}^*$-edge. Hence, we obtain a lower bound of $(kh/2) \cdot (1/m^2)(1-1/m)^{m-2}$ on the probability to increase the $g$-value by 1 and $p_1(k)$ is a correct lower bound for this event.

A necessary event to decrease the $g$-value by 1 is that a selected $\overline{M}^*$-edge $e$ and a free $M^*$-edge $e'$ flip. We distinguish three cases. If $e$ and $e'$ are neighbors, they might be the last or first two edges of a long special path. There are only $2k$ such pairs. The event of flipping such a pair has a probability of at most $2k/m^2$. The second case is that the neighbors $e$ and $e'$ are not the first two or last two edges of a long special path. Then, the free $M^*$-edge $e'$ has another selected $\overline{M}^*$-neighbor $e''$ which must flip, too. Otherwise, the node $u$ was an exposed node and $e$ and $e'$ were the first edges of the long special path starting at $u$ (see Fig. 12). How many possibilities for a free $M^*$-edge $e'$ between two selected $\overline{M}^*$-edges $e$ and $e''$ are there? The $g$ selected $\overline{M}^*$-edges touch $g$ disjoint endpoints in even columns and $g$ disjoint endpoints in odd columns. At most $g$

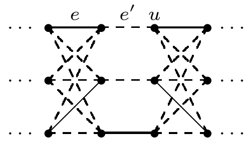


**Fig. 12.** If $u$ was exposed, $e$ and $e'$ were the first two edges of a long special path.

selectable $M^*$-edges connect two such nodes. Hence, there are at most $g$ choices for $e$, $e'$, and $e''$. This leads to a probability of at most $g/m^3$ for the second case. The third case is that the selected $\overline{M}^*$-edge $e$ and the free $M^*$-edge $e'$ are not neighbors. As we work under the condition that no selectable $M^*$-edge flips, we only consider the case that $e'$ is not selectable implying that $e'$ has (at least) one neighbor in the current matching. There are at most $g$ choices for $e$ and at most $(\ell'+1)h$ choices for $e'$. The probability that $e$ flips is $1/m$ and the probability that $e'$ and its neighbors in the matching flip is at most $1/m^2$. Hence, the probability for the last possibility is at most $g(\ell'+1)h/m^3 \le 1/m^2$. In summary, the probability that $g$ decreases by 1 is at most

$$\frac{2k}{m^2} + \frac{g}{m^3} + \frac{1}{m^2} \;\le\; \frac{2k+2}{m^2} \;=:\; p_{-1}(k).$$

To decrease the $g$-value by 2 it is necessary that two selected $\overline{M}^*$-edges $e_1$ and $e_2$ flip. There are at most $\binom{g}{2} \le g^2/2$ choices for $e_1$ and $e_2$. Another necessary event is that two free $M^*$-edges $e_1'$ and $e_2'$ flip; otherwise the step is not accepted. Due to our condition, we only consider the case that $e_1'$ and $e_2'$ are not selectable, implying that they are neighbors of selected $\overline{M}^*$-edges. Since each $\overline{M}^*$-edge has 2 neighbors in $M^*$, there are at most $\binom{2g}{2} \le 2g^2$ choices for $e_1'$ and $e_2'$. As the mutation operator flips $M^*$-edges and $\overline{M}^*$ independently, both necessary events happen with a probability of at most

$$\frac{g^2}{2m^2} \cdot \frac{2g^2}{m^2} \;\le\; \frac{1}{16m^2\ell'^2} \;\le\; \frac{1}{m^2\ell h} \;=:\; p_{-2}(k).$$

The $g$-value decreases by $j \ge 3$ only if $j$ selected $\overline{M}^*$-edges flip and $j$ free $M^*$-edges flip. There are $\binom{g}{j}$ choices for the first $j$ edges and at most $\binom{(\ell'+1)h}{j}$ choices for the last $j$ edges. Hence, the probability is at most

$$\frac{\binom{g}{j}\binom{(\ell'+1)h}{j}}{m^{2j}} \;\le\; \frac{(g(\ell'+1)h)^j}{m^j \cdot m^j} \;\le\; \frac{1}{m^j} \;=:\; p_{-j}(k), \qquad \text{for } j \ge 3. \quad \square$$

Let $p_0(k)$ be the remaining probability. These bounds are independent of the $g$-value of the current search point but the risky steps occurring between the non-risky steps may also change $k$. However, as each search point $s$ has a fixed parameter $k$, we are able to show that the drift conditions hold for all search points with a $g$-value between 1 and $h/2$. The essential reason is that, for all $k$, $p_1(k)$ is by a factor of $\Theta(h)$ larger than $p_{-1}(k)$ and the other probabilities

$p_{-j}(k)$ are small enough. We check the conditions of the drift theorem (where $\ell$ is replaced by $h$). The first two conditions hold since we are interested in a decrease of the $g$-value by $h/8$, and the transition probabilities $p_j(k)$ of the random walk do not depend on $g$. Hence, we can assume $\lfloor h/8 \rfloor$ as the starting point and look for the first point of time where the $g$-value is at most 0. To check the other conditions we increase the probability of reaching a $g$-value of at most 0 by not counting the steps not changing the $g$-value. The transition probabilities of the new process are $q_j(k) := p_j(k)/(1-p_0(k))$, if $j \neq 0$, and $q_0(k) := 0$. We have

$$1 - p_0(k) \;=\; p_1(k) + \sum_{j \geq 1} p_{-j}(k) \;=\; \frac{1}{m^2}\Big(\frac{kh}{2e} + 2k + 2 + O(1/(\ell h))\Big).$$

Hence, for $j = 1$ and $j \leq -1$,

$$q_j(k) \;=\; \frac{m^2 \cdot p_j(k)}{kh/(2e) + 2k + 2 + O(1/(\ell h))} \;\leq\; m^2 \cdot p_j(k).$$

To check the third condition we have to estimate

$$e^{-\lambda} q_1(k) + e^{\lambda} q_{-1}(k) + \sum_{j \geq 2} e^{j\lambda} q_{-j}(k).$$

We prove that for some constants $\lambda > 0$ and $\delta > 0$ small enough

$$e^{-\lambda} q_1(k) + e^{\lambda} q_{-1}(k) \;\leq\; 1 - \delta \qquad \text{and} \qquad \sum_{j \geq 2} e^{j\lambda} q_{-j}(k) \;=\; o(1)$$

which implies the third condition. The second claim is easy to show since $q_{-j}(k) \leq m^2 p_{-j}(k)$ implies that $q_{-2}(k) \leq 1/(\ell h)$ and $q_{-j}(k) \leq 1/m^{j-2}$ for $j \geq 3$. Hence,

$$\sum_{j \geq 2} e^{j\lambda} q_{-j}(k) \;\leq\; \frac{e^{2\lambda}}{\ell h} + e^{2\lambda} \sum_{j \geq 3} \left(\frac{e^{\lambda}}{m}\right)^{j-2} \;=\; o(1).$$

From the proof of Lemma 8 we know that it is sufficient to prove a constant lower bound $\beta > 0$ for $q_1(k) - q_{-1}(k)$ to verify the other condition. For $h$ large enough,

$$\begin{aligned} q_1(k) - q_{-1}(k) \;&=\; \frac{kh/(2e) - (2k+2)}{kh/(2e) + 2k + 2 + O(1/(\ell h))} \\ &\geq\; \frac{h/(2e) - 4}{h/(2e) + 4 + O(1/(\ell h))} \geq 1/2. \end{aligned}$$

The fourth condition of the drift theorem is easy to verify. We get the largest value for $g(s_t) = b(h)$. Then we obtain the same sum as before which is bounded by $D = 1$. To finish the proof we have to take into account that our estimates of the probabilities hold only if the $g$-value is at most $h/2$. Reaching a value larger than $h/2$, the next $g$-value of at most $h/2$ is at least $3h/8$ since we have estimated the probability of larger decreases of $g$ in one step. This may be considered as a new trial. The number of trials is bounded by $2^{ch}$. Choosing $c$ small enough, the probability that at least one trial is successful is still $2^{-\Omega(h)}$. Altogether, the probability that the risky or the non-risky steps decrease the $g$-value by at least $h/8$ is $2^{-\Omega(h)}$. This implies the theorem. □

Theorem 10 and 11 assume that $g \geq h/2$ for the initial search point. Some lower bound on $g$ is necessary for exponential lower bounds that hold with overwhelming probability. The following results show that we can expect to reach a search point with a large $g$-value.

**Theorem 12.** *Let $h = \omega(\log m)$ and $h \leq \ell - 2$. Starting with the empty matching, RLS and the (1+1) EA produce a matching with a $g$-value of $\Omega(h^2)$ with a probability of $1 - 2^{-\Omega(\ell)}$.*

*Proof.* We consider the first $\ell' h/4$ steps of a run and show that it is unlikely to produce $M^*$. Within this phase, there are $|M^*|\ell' h/4$ possibilities to flip an $M^*$-edge for the (1+1) EA, and the expected number of such flips is bounded by $|M^*|\ell' h/(4m) \leq (\ell'+1)/4$. If RLS selects the first edge to be flipped in a step, the probability to choose an $M^*$-edge is $|M^*|/m$, and if RLS decides to flip another edge in the same step, then the probability to choose an $M^*$-edge is at most $|M^*|/(m-1)$. Hence, there are at most $\ell' h/2$ possibilities to flip an $M^*$-edge and the expected number of such flips is at most $|M^*|\ell' h/(2(m-1)) \leq (\ell'+1)/2$. By Chernoff bounds, both algorithms touch at most $\ell' < |M^*|$ of the $M^*$-edges with a probability of $1 - 2^{-\Omega(\ell)}$. In the following, we work under this condition.

As long as less than $\ell' h/4$ $\overline{M}^*$-edges are chosen, by our condition, at most $k := \ell' h/4 + \ell'$ edges are chosen. Each of these $k$ edges has at most $2h$ neighbors. Hence, if at most $k$ edges are chosen, there are at least $\ell' h^2 - \ell' h/4 - 2hk$ selectable $\overline{M}^*$-edges. The last sum is bounded below by

$$\ell' h^2 - \ell' h/4 - 2h(\ell' h/4 + \ell') \;=\; \ell' h^2(1/2 - 9/(4h)) \;\geq\; \ell' h^2/4$$

if $h$ is large enough. Our condition does not decrease the probability of a step flipping solely a selectable $\overline{M}^*$-edge. The probability of such a step is at least $(\ell' h^2/4) \cdot (1/m) \cdot (1 - 1/m)^{m-1} = \Omega(1)$ for both algorithms. Hence, by Chernoff bounds, the $g$-value increases $\Omega(\ell h)$ times with a probability of $1 - 2^{-\Omega(\ell h)}$. Under our condition, the $g$-value may decrease by a total of at most $\ell'$ in steps replacing $\overline{M}^*$-edges with $M^*$-edges. In summary, $g$ obtains a value of $\Omega(\ell h) = \Omega(h^2)$ with a probability of $1 - 2^{-\Omega(\ell)}$. □

**Theorem 13.** *Let $h = \omega(1)$. Choosing a matching uniformly at random, the $g$-value is at least $h$ with a probability of $1 - 2^{-\Omega(\ell h \log h)}$.*

*Proof.* We prove an upper bound $U = 2^{O(\ell h)}$ on the number of matchings with a small $g$-value of at most $h$ and a lower bound $L = 2^{\Omega(\ell h \log h)}$ on the number of matchings. Then $U/L = 2^{-\Omega(\ell h \log h)}$ is an upper bound on the ratio of matchings with a small $g$-value and all matchings.

For the upper bound and $h \geq 3$, there are at most

$$\sum_{0 \leq k \leq h} \binom{\ell' h^2}{k} \leq (h+1) \frac{(\ell' h^2)^h}{h!} \leq (\ell' h^2)^h$$

choices of at most $h$ $\overline{M}^*$-edges. Each choice can be combined with at most each of the $2^{(\ell'+1)h}$ subsets of $M^*$. Hence, we obtain

$$U := (\ell' h^2)^h \cdot 2^{(\ell'+1)h} = 2^{O(\ell h)}.$$

For the lower bound, we consider only the nodes in Columns 1, 5, 9, and so on (see Fig. 8). In a matching, there are $h+2$ options for a node. A node may either be free, covered by the incident $M^*$-edge, or covered by one of the $h$ incident $\overline{M}^*$-edges. Now we take these decisions for $\lceil h/2 \rceil$ nodes in each of the selected columns. As these columns have a distance of three edges our choices in one column do not influence our choices in the next column. But if we choose an $\overline{M}^*$-edge, the number of options for the other nodes in the same column decreases by 1. Hence, for each of the $\Omega(\ell h)$ decisions to take, we can choose between at least $h + 2 - \lceil h/2 \rceil \geq h/2$ options. This leads to at least

$$L := (h/2)^{\Omega(\ell h)} = 2^{\Omega(\ell h \log h)}$$

different matchings. □

## 10 Generalizations

We have seen that RLS and the (1+1) EA find maximum matchings in expected polynomial time only for a subclass of all graphs. We are far from a complete characterization of the class of graphs where our heuristics are efficient. Nevertheless, there are some preliminary ideas. A semi-augmenting path is an alternating path that starts at a free node and that cannot be lengthened to an augmenting path. RLS and the (1+1) EA work more or less locally and cannot distinguish between augmenting and semi-augmenting paths. Our conjecture is that the presence of exponentially many semi-augmenting paths and only polynomially many augmenting paths at many points of time prevent the heuristics from being efficient. Trees have only polynomially many paths whereas, for $G_{h,\ell}$, we have many situations with one augmenting path and exponentially many semi-augmenting paths.

It would also be interesting to investigate other search heuristics. All matchings contain at most $n/2$ edges and, therefore, at least $m - n/2$ non-edges. If $m \gg n$, the typical search operators which treat ones and zeroes in the same way are no longer appropriate. Alternatives look like this:

- If RLS decides to flip two bits, it flips a randomly chosen 0-bit and a randomly chosen 1-bit,
- for the (1+1) EA, each 1-bit is flipped with probability $1/n$ and each 0-bit with probability $1/m$.

The expected waiting time for a 2-bit flip flipping a special 0-bit and a special 1-bit decreases from $\Theta(m^2)$ to $\Theta(nm)$ and we conjecture that the expected optimization time decreases by a factor of $\Theta(n/m)$ for many graphs.

Evolutionary algorithms work in generations where many offspring are generated in parallel. One does not expect to decrease the expected number of fitness evaluations but to decrease the expected parallel time, i. e., the number of generations. Generating $\lambda$ offspring in parallel one may hope to decrease the expected number of generations by a factor of $\Theta(\lambda)$ if $\lambda$ is not too large. For maximum matchings, the last phase is the most difficult one and the probability of producing an accepted offspring can be as small as $\Theta(1/m^2)$ (for the usual search operator) or $\Theta(1/(nm))$ (for the new search operator discussed above). Choosing $\lambda = \Theta(m^2)$ or $\lambda = \Theta(nm)$, resp., we can hope for a saving of a factor $\Theta(\lambda)$ for the expected number of generations.

One may believe that $\lambda$ should be smaller in the first phase of the search. Jansen *et al.* (2004) have presented a scheme to choose $\lambda$ adaptively. For time periods of some length $T$, the fraction $\alpha$ of offspring different from the parent and at least as good as the parent is computed and then $\lambda := \lceil 1/\alpha \rceil$ is chosen for the next period. This seems to be quite adequate for the maximum matching problem.

Finally, one can compare variants with larger populations of size $s$ to $s$ independent runs with populations of size 1. We conjecture that it is better to have independent runs.

## Conclusions

Randomized search heuristics without problem-specific modules are analyzed for the maximum matching problem. The results show how heuristics can "use" algorithmic ideas not known to the designer of the algorithm. For simple graphs they find maximum matchings in expected polynomial time but they can be fooled with overwhelming probability by carefully constructed graphs. The results contribute to the understanding how simple heuristics work. Moreover, this is one of the first results where an evolutionary algorithm is analyzed on a well-known combinatorial problem.

## Bibliography

Bast, H., Mehlhorn, K., Schäfer, G., and Tamaki, H. (2004). Matching algorithms are fast in sparse random graphs. *Proc. of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS),* LNCS 2996, 81–92.

Coppersmith, D., Tetali, P., and Winkler, P. (1993). Collisions among random walks on a graph. SIAM Journal on Discrete Mathematics 6, 363–374.

Droste, S., Jansen, T., and Wegener, I. (1998). On the optimization of unimodal functions with the (1+1) evolutionary algorithm. *Proc. of the 5th Conf. on Parallel Problem Solving from Nature (PPSN V),* LNCS 1498, 13–22.

Droste, S., Jansen, T., and Wegener, I. (2001). Dynamic parameter control in simple evolutionary algorithms. W. N. Martin and W. M. Spears, editors, *Foundations of Genetic Algorithms 6 (FOGA 2000).* Morgan Kaufmann, San Francisco, 275–294.

Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. Theoretical Computer Science 276, 51–82.

Feller, W. (1968). *An Introduction to Probability Theory and Its Applications.* Wiley.

Giel, O. and Wegener, I. (2003). Evolutionary algorithms and the maximum matching problem. *Proc. of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS),* LNCS 2607, 415–426.

Giel, O. and Wegener, I. (2005). Searching randomly for maximum matchings. Submitted for publication.

Hajek, B. (1982). Hitting-time and occupation-time bounds implied by drift analysis with applications. Advances in Applied Probability 14, 502–525.

He, J. and Yao, X. (2001). Drift analysis and average time complexity of evolutionary algorithms. Artificial Intelligence 127, 57–85.

Hopcroft, J. E. and Karp, R. M. (1973). An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing 2, 225–231.

Jansen, T., De Jong, K. A., and Wegener, I. (2004). On the choice of the offspring population size in evolutionary algorithms. Submitted for publication.

Jansen, T. and Wegener, I. (2001). Evolutionary algorithms – how to cope with plateaus of constant fitness and when to reject strings of the same fitness. IEEE Trans. on Evolutionary Computation 5, 589–599.

Jerrum, M. (1992). Large cliques elude the Metropolis process. Random Structures and Algorithms 3, 347–359.

Jerrum, M. and Sinclair, A. (1989). Approximating the permanent. SIAM Journal on Computing 18, 1149–1178.

Jerrum, M. and Sorkin, G. B. (1998). The Metropolis algorithm for graph bisection. Discrete Applied Mathematics 82, 155–175.

Micali, S. and Vazirani, V. V. (1980). An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. *Proc. of the 21st Annual Symp. on Foundations of Computer Science (FOCS)*, 17–27.

Motwani, R. (1994). Average-case analysis of algorithms for matching and related problems. Journal of the ACM 41, 1329–1356.

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. *Proc. of the 6th Genetic and Evolutionary Computation Conference (GECCO),* LNCS 3102, 713–724.

Reetz, M. (2001). Combinatorial and evolution-based methods in the creation of enantioselective catalysts. Angewandte Chemie 40, 284–310.

Sasaki, G. (1991). The effect of the density of states on the Metropolis algorithm. Information Processing Letters 37, 159–163.

Sasaki, G. H. and Hajek, B. (1988). The time complexity of maximum matching by simulated annealing. Journal of the ACM 35, 387–403.

Scharnow, J., Tinnefeld, K., and Wegener, I. (2002). Fitness landscapes based on sorting and shortest paths problems. *Proc. of the 7th Conf. on Parallel Problem Solving from Nature (PPSN VII),* LNCS 2439, 54–63.

Vazirani, V. V. (1994). A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{V}E)$ maximum matching algorithm. Combinatorica 14, 71–109.

Wegener, I. (2001). Theoretical aspects of evolutionary algorithms. *Proc. of the 28th Internat. Colloq. on Automata, Languages and Programming (ICALP),* LNCS 2076, 64–78.

Wegener, I. and Witt, C. (2004a). On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions. Journal of Discrete Algorithms. To appear.

Wegener, I. and Witt, C. (2004b). On the optimization of monotone polynomials by simple randomized search heuristics. Combinatorics, Probability, and Computing. To appear.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'