6

Maximum Matchings

The maximum matching problem is a very well-studied combinatorial optimization problem. Given an undirected graph $G = (V, E)$, a matching is a subset $E' \subset E$ of the edge set such that no two edges in E' share a common endpoint. The maximum matching problem asks for a matching of maximum cardinality. Such problems arise, e.g., in team planning when edges of a graph denote possible collaborations of workers and the aim is to find a biggest partition of the workers into teams of size 2. Therefore, matching problems have numerous generalizations to hypergraphs and weighted graphs, which will not be discussed in this chapter. The maximum matching problem should not be confused with the maximal matching problem, where the aim is to find a subset of edges which is maximum with respect to inclusion, i.e., no proper superset of the matching is a matching.

The maximum matching problem is solvable in polynomial time. The best algorithms for the general case run in time $O(\sqrt{|V|} \cdot |E|)$ (Micali and Vazirani, 1980), which is also the best bound known for the special case of bipartite graphs (Hopcroft and Karp, 1973). However, the algorithms for the latter case are much simpler to describe and to analyze. All of them rely on the fundamental concept of so-called *augmenting paths*, which will be explained in detail below. Augmenting paths represent a way to improve the size of a matching by performing local changes along the path. Hence, there is some hope that locally searching algorithms are able to find maximum matchings. This motivated early analyses of stochastic search algorithms for this problem, most notably a study by Sasakik and Hajek (1988) with respect to simulated annealing.

The contents of this chapter are based on and follow closely the works by Giel and Wegener (2003, 2004, 2006), who concentrate on variants of randomized local search and (1+1) $\mathrm{EA_b}$ for the maximum matching problem. In Section 6.1, we describe the investigated search algorithms and fitness functions precisely and supply additional concepts for the analysis. Section 6.2 deals with a general result on the approximation capability of the algorithms. For certain graph classes, exact solutions to the problem can be found in

F. Neumann, C. Witt, *Bioinspired Computation in Combinatorial Optimization*, Natural Computing Series,
DOI 10.1007/978-3-642-16544-3_6,

expected polynomial time, which will be presented in Section 6.3. However, there are also graph instances for which no optimal solutions are found within polynomial time, with high probability. A worst-case result in this vein is described in Section 6.4.

6.1 Representations and Underlying Concepts

Giel and Wegener (2003, 2004, 2006) work with the following model of the maximum matching problem. Let $n = |V|$ denote the number of vertices and $m := |E|$ the number of edges of the graph for which a maximum matching is sought. Again, the encoding for binary search spaces is straightforward. When working with bitstrings of length m (!), a search point $s = (s_1, \dots, s_m) \in \{0,1\}^m$ is interpreted as the characteristic vector of the chosen subset of edges. If s describes a valid matching, the fitness function $f\colon \{0,1\}^m$ just returns the number of chosen edges, i.e., $f(s) = s_1 + \cdots + s_m$. As in Chapter 5, several ways to handle invalid search points, in this case non-matchings, make sense. One way would be to assign large negative $f$values to them and to force the search algorithm to start from the empty matching, i.e., the all-zeros string. This kind of initialization is used in Sasaki's (1988) definition of simulated annealing. We stick with the uniform initialization used in the common stochastic search algorithms and introduce a component in the fitness function to direct the search towards valid matchings. The following idea is similar to that in Chapter 5 with the MST problem, where the number of connected components was to be minimized first in order to direct the algorithm to trees.

If $d(v) > 1$ edges incident on a vertex v are chosen by the search point s, a penalty $p(v)$ of value $d(v) - 1$ is assigned to the vertex; otherwise, $p(v) = 0$. Hence, exactly the vertices that are in accordance with the definition of a matching have no penalty. The penalty $p(s)$ of the search point is simply the sum of all vertex penalties, and the fitness function equals

$$f(s) = (-p(s), |s|_1).$$

This function has to be *maximized* in lexicographic order. As soon as a value of 0 has been obtained with respect to the first component, only valid matchings are considered.

The stochastic search algorithms studied are a randomized local search algorithm on the one hand and (1+1) $\mathrm{EA_b}$ on the other hand. As for the MST problem studied in Chapter 5, a neighborhood of size 1 is not sufficient. In order for the search algorithm to accept a different matching, it can be necessary to flip out one edge and to include another edge. Of course, if there is an edge that is neither included in the current matching nor incident on another matching edge, then this edge can be chosen and leads to a larger matching. We call such edges *free* edges since both of its vertices are free, i.e., not incident on any matching edges. However, the lack of free edges does not mean that a matching is maximum. Instead, a characterization of optimality is based on the above-mentioned augmenting paths. We call a path through the vertices $v_1, \ldots, v_k$ an *alternating path* of length $k-1$ with respect to a current matching if the edges $\{v_{2i}, v_{2i+1}\}$, $1 \leq i \leq k/2$, belong to the matching and the other edges do not belong to it. If additionally v_1 and v_k are free vertices, which is only possible for even k, i.e., an odd number of edges, then the path is called *augmenting*. In this case, we swap matching and non-matching edges along the augmenting path, which means that we remove from the matching all edges on the path that so far belong to the matching and add to the matching those edges on the path that do not belong to it. This procedure leads to a valid matching of increased (that is, "augmented") cardinality. A single free edge appears as the special case of an augmenting path of length 1. The following theorem is a fundamental characterization of optimal matchings.

Theorem 6.1 (Hopcroft and Karp, 1973). *A matching is of maximum cardinality if and only if there exists no augmenting path with respect to the matching.*

(1+1) $\mathrm{EA_b}$ can flip all edges of an augmenting path at once. A local search algorithm cannot do this in a single step. However, it can approach the improved matching by flipping two adjacent edges. If $v_1, \ldots, v_k$ is augmenting then $\{v_1, v_2\}$ can be turned into a matching edge and $\{v_2, v_3\}$ into a non-matching edge. This results in v_3 becoming free and $v_3, \ldots, v_k$ forming a new and shorter augmenting path. This motivates us to study the algorithm $\mathrm{RLS}_\mathrm{b}^{1,2}$ defined in Definition 2. The only change is that we apply both search algorithms for maximization, i.e., the condition $f(s') \leq f(s)$ is replaced by $f(s') \geq f(s)$ in the definition of both $\mathrm{RLS}_\mathrm{b}^{1,2}$ and (1+1) $\mathrm{EA_b}$ (see Algorithms 2 and 3).

6.2 Approximation Quality for General Graphs

The stochastic search algorithms start from a completely random string. The definition of the fitness function and the elitist selection of (1+1) $\mathrm{EA_b}$ and $\mathrm{RLS}_\mathrm{b}^{1,2}$ ensure that only matchings are accepted as future search points once a valid matching has been found. This happens efficiently as the following lemma shows.

Lemma 6.2. *$RLS_b^{1,2}$ and (1+1) EA_b find search points that represent matchings in expected time $O(m \log m)$.* □

Proof. We argue as in the proof of the coupon collector's problem (Section 4.2.2). Let $k = -p(s)$ be the sum of the vertex penalties with respect to the search point s. Then k is less than $2m$, the sum of all vertex degrees.

Until a valid matching is found, only the first component of the fitness function is relevant, i.e., new search points are only accepted if they have a lower total penalty. By definition, there are at least $\lceil k/2 \rceil \leq m$ edges chosen by s, whose elimination decreases k. The probability of a specific 1-bit mutation equals $\Theta(1/m)$ for both algorithms. Hence, the expected waiting time to decrease k is bounded by $O(m/k)$. Summing up for $1 \leq k < 2m$ and estimating the Harmonic series according to $\sum_{k=1}^{2m} 1/k = O(\ln m)$ yields the claim. □

The aim of this section is to show that the search algorithms are able to find good approximate solutions to the maximum matching problem for arbitrary graphs. The result is also based on the result by Hopcroft and Karp (1973). The main idea is as follows. Given a matching that is far away from optimality, there must not only be one, but many augmenting paths. The pigeonhole principle guarantees the existence of a relatively short augmenting path. This is made precise by the following lemma.

Lemma 6.3. *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 from above by $L := 2\lfloor |M|/(|M^*| - |M|) \rfloor + 1$.*

Proof. Let $G' = (V, E')$ be the graph whose edge set is defined by $E' := M \oplus M^*$, where $\otimes$ denotes the symmetric difference, i.e., the exclusive OR of the search points. The graph G' consists of vertex-disjoint cycles and paths. Each cycle and each path of even length has the same number of M and M^* edges. Paths of odd length alternate between M and M^* edges. There is no odd-length 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. □

By means of the preceding lemma, we arrive at the announced result on the approximation quality of the search algorithms.

Theorem 6.4. *For $\epsilon > 0$, $RLS_b^{1,2}$ and (1+1) EA_b find a $(1 + \epsilon)$ optimal matching in expected time $O(m^{2\lceil 1/\epsilon \rceil})$ independently of the choice of the first search point.*

Proof. The first phase of the search finishes when a matching is found. By Lemma 6.3, this phase is short enough to be captured by the proposed runtime bound. Afterwards, let M be the current matching, and let M^* be an arbitrary maximum matching. The search is successful if $|M^*| \leq (1 + \epsilon)|M|$. Otherwise, by Lemma 2, there exists an augmenting path for M whose length is bounded from above by $L := 2\lfloor |M|/(|M^*| - |M|) \rfloor + 1$. Since $|M^*| > (1 + \epsilon)|M|$, we conclude that

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

Consequently,

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

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

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

$\text{RLS}_\text{b}^{1,2}$ can flip the augmenting path in $\lfloor \ell/2 \rfloor + 1$ steps. In each of the first $\ell/2$ 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 from below by $\Omega((m^{-2})^{\lfloor \ell/2 \rfloor} \cdot m^{-1}) = \Omega(m^{-\ell})$, where we used the fact that the length ℓ of an augmenting path is odd. The expected number of unsuccessful phases preceding a successful phase is $O(m^\ell)$. Again, we have $\ell \leq L$. The difference with the case of (1+1) EA_b 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 from 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 from above by $O(\ell + m^\ell) = O(m^\ell)$, which proves the theorem. □

The previous theorem also implies that the simple stochastic search algorithms are PRASs (polynomial-time randomized approximation schemes) in the sense of Definition 2.8. The following corollary, which follows from Theorem 6.4 by using Markov's inequality, makes this explicit. We just let c be a constant such that Theorem 6.4 holds for the bound $c \cdot m^{2\lceil 1/\epsilon \rceil}$.

Corollary 6.5. *If we run $RLS_b^{1,2}$ or (1+1) EA_b for $4cm^{2\lceil 1/\epsilon \rceil}$ iterations, we obtain a PRAS for the maximum matching problem, i.e., independently of the choice of the first search point, the probability of producing a $(1+\epsilon)$ optimal solution is at least $3/4$.*

6.3 Upper Bounds for Simple Graph Classes

After seeing that $\text{RLS}_\text{b}^{1,2}$ and (1+1) EA_b find good approximations to maximum matchings in expected polynomial time, we are interested in graphs where even maximum matchings can be found in expected polynomial time. We start with the simple graph called *path*. As the name suggests, it consists 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 the search algorithms on this graph contains typical aspects of their behavior on more complicated instances.

As in Chapter 5, there are typically many steps of $\mathrm{RLS}_\mathrm{b}^{1,2}$ and (1+1) EA_b leading to infeasible search points, in this case search points that do not encode a matching. Such steps slow the search algorithms down, but cannot be avoided without introducing problem-specific knowledge, which is not always available. In the forthcoming analysis, we account for such steps by the consideration of so-called *relevant* steps, where the exact definition will depend on the situation. Denoting by R the number of relevant steps and by T the total number of steps until a certain goal is achieved, the following argumentation is typical. If an expected number of $E(R)$ relevant steps is necessary and every step is relevant with probability at least p, then the expected total number of steps $E(T)$ is at most $p^{-1} \cdot E(R)$.

We start with a simple upper bound for $\mathrm{RLS}_\mathrm{b}^{1,2}$ on the path graph.

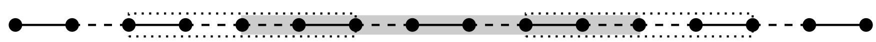

Fig. 6.1. An augmenting path (indicated by gray area) and environments of possible mutations leading to extensions or shortenings (dotted)

Theorem 6.6. *For a path of m edges, the expected optimization time of $RLS_b^{1,2}$ is $O(m^4)$ independently of the choice of the first search point.*

Proof. By Lemma 6.2, the expected waiting time for a matching is small enough to be captured by the $O(m^4)$ bound. 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 i augmenting paths and one of length at most $\ell := m/i$. In every step, we conceptually select a shortest augmenting path P; hence the considered P might be different in the course of optimization. Now a step is called Prelevant if it is accepted and P is altered. The probability of a Prelevant step is $\Omega(1/m^2)$. This is due to the following observations. 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; otherwise the path consists of only a free edge, and the considered probability is even $\Omega(1/m)$. If we can show that an expected number of $O(\ell^2)$ Prelevant steps is sufficient to improve the matching by one edge, then $\sum_{i=1}^{\lceil m/2 \rceil} O((m/i)^2) = O(m^2)$ Prelevant steps are sufficient, and the expected optimization time is $O(m^4)$.

If $|P| \geq 3$, there are no free edges. Only mixed mutation steps, where a non-matching edge and a matching edge flip, can be accepted. Since each non-matching edge e has at least one neighbor e' in the matching, e' must flip, too. That means that only a non-matching edge e incident on a free vertex together with a matching edge e' such that e and e' have an endpoint in common can flip. In the considered case of a path graph, only pairs of neighbored edges located at one end of an alternating path can flip in accepted steps. Such a pair consists of either two neighbored edges outside P or two edges inside P, in both cases with one edge incident on an endpoint of P. (See Figure 6.1

for an illustration, where either the first two or the last two edges in one of the dotted boxes are allowed to flip together.) The first case increases the length of P and the second one decreases it. Since P might be aligned with an endpoint of the whole graph itself, the situation can even be in favor of decreasing steps. Hence, P is shortened with probability at least $1/2$ if a pair of neighbored edges flips. If $|P| = 1$, the probability that the length of P is decreased to 0 is at least $1/(2m)$ since a 1-bit mutation of $\mathrm{RLS}_{\mathrm{b}}^{1,2}$ is sufficient. In contrast, the probability that the path grows at either end is at most $2\cdot(1/2)\cdot\binom{m}{2}^{-1} = 2/(m(m-1))$ in this case. Hence, the conditional probability that the next Prelevant step is decreasing is at least $1/(1+4/(m-1)) \geq 1/2$, for $m \geq 5$.

Taking the two cases together, we are confronted with a random walk on the numbers $\{0, 1, 3, 5, \ldots, \ell\}$ describing the current length of P. This walk goes from a state to the lower neighboring state with probability at least $1/2$ and to the higher neighboring state otherwise. Since we are interested in reaching state 0, we may pessimistically assume the transition probabilities to be exactly $1/2$ and arrive at the scenario relevant for Theorem 4.7. The graph on which the random walk takes place is itself a path; hence its number of edges is trivially bounded by ℓ. The time to reach state 0, i.e., one end of this path, is bounded by the cover time for the graphs, which is $O(\ell^2)$ using Theorem 4.7. $\square$

Basically the same ideas as in the proof of Theorem 6.6 can be used to prove also the bound $O(m^4)$ for (1+1) EA_{b}. However, the analysis is complicated by the fact that the latter search algorithm can flip many bits in a step. We are only interested in Prelevant steps. For our analysis, we define Pclean steps, which are Prelevant steps causing only small changes in P. Then, a phase including $\Theta(\ell^2)$ Prelevant steps is called Pclean if all its Prelevant steps are Pclean. The idea is to prove that a phase is Pclean with probability $\Omega(1)$ and that a Pclean phase plus the next Prelevant step improve the matching with probability $\Omega(1)$.

Theorem 6.7. *For a path of m edges, the expected optimization time of (1+1) EA_b is $O(m^4)$ independently of the choice of the first search point.*

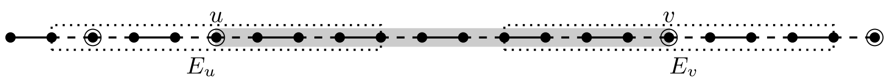

Fig. 6.2. An augmenting path (indicated by gray area) and the environments E_u and E_v (dotted); free vertices are indicated by a circle

Proof. For the definition of P, ℓ, and Prelevant steps, see the proof of Theorem 6.6. With the same arguments as used there, it suffices to prove that the expected number of Prelevant steps to improve the matching is $O(\ell^2)$.

Pclean steps are only defined for situations without free edges. Let u and v be the endpoints of P, and let E_u be the set of edges where at least one endpoint has at most a distance of 3 from u, and analogously for E_v (see Figure 6.2 for an illustration). Then we call a Prelevant step a Pclean 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.

We describe the effect of clean steps on P. The free vertices partition the graph into alternating paths (see also Figure 6.2 for an example). As there is no free edge, there is an augmenting path of at least three edges between a free vertex and the next free vertex. Hence, a Pclean 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 Pclean step; however, it is possible that new free vertices 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 Pclean step lengthens P by more than two edges, i.e., at least four edges, since this requires flipping more than three edges in E'. Thus, Pclean steps lengthen P only by 2, and to this end it is necessary to flip a pair of neighbored edges outside the augmenting path but touching either u or v (the situation already discussed in Figure 6.1). For a Pclean step to decrease the length of P by at least 2, it is sufficient to flip a pair of neighbored edges at either end of the augmenting path (see again Figure 6.1). Since at most three edges of E' may flip, at most one of the discussed pairs of neighbored edges can flip in a Prelevant step. Hence, Prelevant 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 free edge, it is convenient to include these good events into Pclean steps. We broaden our definition of Pclean steps and call accepted steps that produce a free edge or improve the matching Pclean, too. Now, we upper bound the probability of Prelevant but not Pclean steps (in situations without free edges). A necessary event to violate the first property is that four 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 the 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 at most ten 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 non-matching edge or one matching edge in B. If the surplus is not balanced outside B, the step is either not accepted because the fitness would decrease or the step is clean because the matching is improved. To compensate for a surplus of one non-matching edge, one more non-matching edge than the number of flipping matching edges must flip elsewhere. This may be a non-matching 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 free edge is created, the step is clean, or another block of at least three edges flips. The probability of the last possibility is again $O(1/m^5)$. Altogether, the probability of a Prelevant but not Pclean step is $O(1/m^4)$, and the conditional probability that a Prelevant step is not Pclean is $O(1/m^2)$. Hence, a phase of $O(\ell^2) = O(m^2)$ Prelevant steps is clean with probability $\Omega(1)$.

Pessimistically assuming that shortenings shorten the path by exactly two edges and that the probability of shortening in a clean, relevant step is exactly 1/2, we treat this as a fair random walk as in the last paragraph of the proof of Theorem 6.6. Hence, 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 a large enough constant. Afterwards, at least one free edge exists, and a step is Prelevant with a probability of $\Omega(1/m)$, Hence, the next Prelevant step improves the matching with probability $\Omega(1)$. □

The results from the previous two theorems deserve some discussion. 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 ways to lengthen an augmenting path. As indicated above, the relatively large time bound $O(m^4) = O(n^4)$ can the explained by the characteristics of general (and somehow blind) local search. As the search algorithm does not "know" that only matchings are valid search points, it keeps wasting a lot of steps by producing invalid search points and rejecting them immediately. Moreover, while the analysis focuses on a shortest augmenting path, there may be many steps which alter the search point at a completely different place. 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 $O(m^2) = O(n^2)$. Actually, the search on the level of second-best matchings is responsible for this. If the number of edges is odd, the path graph has a unique maximum matching consisting of $\lceil m/2 \rceil$ edges. Therefore, any second-best matching of size $\lfloor m/2 \rfloor = \lceil m/2 \rceil - 1$ has only one augmenting path P. We show that the simple search algorithms have an expected optimization time of $\Omega(m^4)$ if the initial situation is a second-best matching and P is not too short.

Theorem 6.8. *For a path of m edges, m odd, the expected optimization time of $RLS_b^{1,2}$ and (1+1) EA_b 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 Theorems 6.6 and 6.7. For the lower bounds, we would like to exploit the properties of the random walk describing the length of the augmenting path, analyzed in the two theorems and illustrated in Figure 6.1. Note that only 2-bit flips of $\mathrm{RLS}_\mathrm{b}^{1,2}$ are possible in relevant steps. Hence, as long as the augmenting path P is not adjacent to a border of the path graph itself and at most two edges flip, we are confronted with a fair random walk increasing or decreasing the length of the path with probability $1/2$ each in relevant steps. Only if P is at a border can the probability of decreasing the length be greater than $1/2$ in relevant steps. This corresponds to the scenario of the gambler's ruin theorem with $p = q = 1/2$ (see Theorem 4.8) except for the fact that the game might be changed when P touches a border. If we assume P to be at distance $\Omega(m)$ from both borders and to have initial length $\Omega(m)$, then an endpoint of the path has to move by a distance of $\Omega(m)$ or the whole path has to shrink in length by at least $\Omega(m)$ before the process differs from the fair gambler's ruin game. Using $a = \Omega(m)$ and $b - a = \Omega(m)$ in Theorem 4.8, the expected number of steps needed for the process to move by at least $\Omega(m)$ states is $\Omega(m^2)$. It is easy to see that $\Omega(m^2)$ is not only a lower bound on the expectation, but that $\Omega(m^2)$ relevant steps are also needed with probability $\Omega(1)$. (If the latter did not hold, we would immediately obtain a better bound on the expectation by repeating independent phases.) Since a relevant step has probability $\Theta(1/m^2)$, the lower bound for $\mathrm{RLS}_\mathrm{b}^{1,2}$ follows.

For (1+1) EA_b, the considered 2-bit flips have also probability $\Theta(1/m^2)$ but we must also 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 half the initial length of P. Then, the length of P is always at least $\Omega(m)$ and the probability of a step flipping exactly the edges of P is small enough. If $2k$ edges 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 steps flipping more than two bits is $O(1/m^4)$ in each step. Within βm^4 steps, this expected decrease is $O(1)$ and the decrease is less than half the initial length of P with probability $1 - o(1)$ if the constant $\beta > 0$ is small enough. □

Giel and Wegener (2004, 2006) extend the previous results from paths to trees, i.e., connected graphs without cycles. They conjecture that path graphs represent the most difficult instance within the class of tree graphs since a path is a tree with maximal diameter and the diameter bounds the length of a longest augmenting path.

We do not present the complete involved analysis that Giel and Wegener (2004, 2006) perform for $\mathrm{RLS}_\mathrm{b}^{1,2}$ on trees since such a presentation would be beyond the scope of this book. However, it is possible to present the general idea behind why $\mathrm{RLS}_\mathrm{b}^{1,2}$ finds maximum matchings on complete trees in expected polynomial time. More precisely, the authors obtain the following theorem.

Theorem 6.9. *The expected time until* $RLS_b^{1,2}$ *finds a maximum matching on a complete kary tree,* $k \geq 2$*, is bounded by* $O(m^{7/2})$ *independently of the choice of the first search point.*

When $\mathrm{RLS}_\mathrm{b}^{1,2}$ operates on complete $k$ary trees, there are two essential differences with respect to the path graph. Given a situation without free edges and an augmenting path P shorter than the diameter, there must be a free vertex v at one end of P that is not a leaf (vertex of degree 1) of the graph. This means that v must have degree k, which implies that there are $k-1$ ways to lengthen and only one way to shorten P. Each of these possibilities is chosen with the same probability, and for $k \geq 2$, $\mathrm{RLS}_\mathrm{b}^{1,2}$ is confronted with an unfair game that is biased towards increasing the length of P. In terms of the gambler's ruin theorem (Theorem 4.8), the event of P reaching its maximal length D, where D is the diameter of the graph, corresponds to the gambler's ruin. The probability of the gambler's gain, i.e., reaching length 0 before length D, starting from a worst-case length $D-1$, equals

$$1 - \frac{(k-1)^D - (k-1)}{(k-1)^D - 1} = \frac{1}{(k-1)^{D-1}},$$

which is exponentially small in D. On the other hand, it holds that $D \leq 2\lceil \log_k m \rceil$ since the depth of the $k$ary tree is at most $\lceil \log_k m \rceil$. Inserting this into the above formula results in a probability of $\Omega(1/poly(n))$ for the gambler's gain. Since the expected length of the unfair game is also polynomial (Theorem 4.8), we obtain an overall expected polynomial time until the matching is improved.

Finally, using much more sophisticated arguments, Giel and Wegener (2004, 2006) extend the analysis to the case of arbitrary trees. They obtain the following theorem.

Theorem 6.10. *The expected time until* $RLS_b^{1,2}$ *finds a maximum matching in a tree with diameter* D *is bounded by* $O(D^2 m^4)$ *independently of the choice of the first search point.*

The authors also believe that basically the same results hold for (1+1) EA_b, but in the case of arbitrary trees it is much more difficult to control the effect of steps flipping more than two bits than it is for the path graph. This concludes the presentation of the positive results. In the following section, we explore the limits of the search algorithms.

6.4 A Worst-Case Result

The result of Theorem 6.4 shows that $\mathrm{RLS}_\mathrm{b}^{1,2}$ and (1+1) EA_b represent good approximation algorithms for the maximum matching problem. However, in the worst case they are not able to find an optimum in expected polynomial time. The analysis by Giel and Wegener (2003, 2006) is based on a graph class that was introduced by Sasakik and Hajek (1988). The graph $G_{h,\ell}$ for odd $\ell = 2\ell'+1$ is best illustrated by placing its $n := h(\ell+1)$ vertices in h rows and $\ell+1$ columns 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 $\{(i,j),(i,j+1)\}$, $1 \le i \le h$. In contrast, there are complete bipartite graphs between column j and column $j+1$ for odd values of j. The graph $G_{3,11}$ is shown in Figure 6.3. 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, we have $m = |M|+|\overline{M}^*| = (\ell'+1)h+\ell'h^2 = \Theta(\ell h^2)$ for the number of edges.

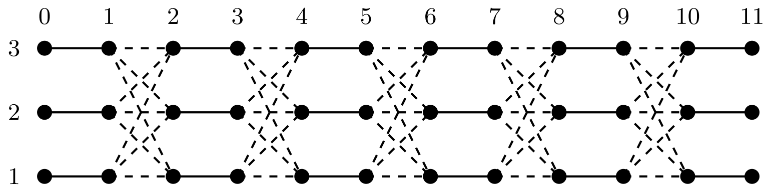

Fig. 6.3. The graph $G_{h,\ell}$, $h = 3$, $\ell = 11$, and its perfect matching

For the forthcoming analyses, it is sufficient to consider second-best (also called *almost perfect*) matchings of size $|M^*|-1$ for the graph $G_{h,\ell}$ and to show that the final improvement takes in expectation an exponential time. Given an almost perfect matching, there is only one unique augmenting path left (a formal proof for this fact is already contained in the proof of Lemma 6.3). This augmenting path has the following properties.

Lemma 6.11. *Let Q be the unique augmenting path for an almost perfect matching in the graph $G_{h,\ell}$. Then*

- *Q "runs from left to right", i.e., it contains at most one vertex from each column,*
- *if the endpoints of Q are not in the first or last column, there are $2h$ lengthenings and two shortenings by 2-bit flips; otherwise there are h lengthenings and still two shortenings.*

Proof. For the first property, assume that two vertices belonging to the same column both lie on Q. Due to the structure of $G_{h,\ell}$, this implies in particular the existence of an odd column j with this property. Then Q runs along adjacent edges $e' = \{(i',j),(i,j+1)\}$ and $e'' = \{(i,j),(i,j+1)\}$, both of which are in $\overline{M}^*$. Either e' or e'' is contained in the almost perfect matching, and w. l. o. g., this is the case for e'. We consider Q as running from left to right in row i', then changing direction via e' and e'', and subsequently running to

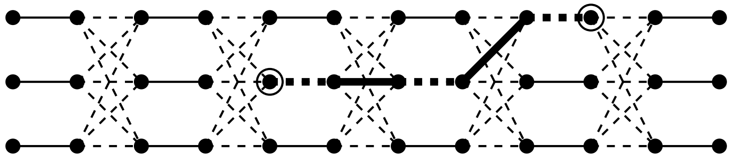

Fig. 6.4. An almost perfect matching in $G_{h,\ell}$ and its augmenting path; the free vertices are marked by a circle

the left in row i. We exploit the fact that e'' is not included in the almost perfect matching. Hence, as long as Q continues along row i to the left, the almost perfect matching can only contain M^* edges. This implies that Q will not change to another row again and ends at a free vertex (i, j') in the very same row. If also $(i, j'-1)$ is a free vertex, then edge $\{(i, j'-1), (i, j')\}$ is free; hence there is another augmenting path in contradiction to the fact that we have an almost perfect matching. If $(i, j'-1)$ is not free, the almost perfect matching must include another $\overline{M}^*$ edge, which again implies the existence of another augmenting path.

As a consequence of the first property and the fact that we are dealing with an almost perfect matching, at least one endpoint of Q is adjacent to h different $\overline{M}^*$ edges that are not contained in the almost perfect matching. If the other endpoint of Q is not in the first or last column, this also holds for the other endpoint. This implies the second property. □

Lemma 6.11 implies that the search algorithms are confronted with an unfair game if $h \geq 3$. The tendency towards increasing the length of Q will provably result in an exponential expected optimization time. To formalize this, we use a potential function, denoted by P, which maps search points (i.e., selections of edges) to integral values. For the sake of convenience, this function is defined only for almost perfect matchings and denotes the current length of the unique augmenting path for such a matching; hence, P takes only odd values. Note that P is not injective. In particular, we cannot determine from the Pvalue whether the augmenting path is adjacent to a border of the $G_{h,\ell}$. Assuming a worst-case perspective, we assume one endpoint of the path to be at a border. Then there are, according to Lemma 6.11, h 2-bit flips increasing and two such flips decreasing the potential. Giel and Wegener (2003, 2004) consider also the case $h = 2$, which is of some special interest since $G_{2,\ell}$ is the only planar graph in this class of graphs. However, much additional effort is required in this case to show that the augmenting path is likely to be far away from a border and that the game is, therefore, still unfair. In the following, we present the analysis only for the case $h \geq 3$.

The proof strategy for an exponential lower bound with respect to the search algorithms $\text{RLS}_\text{b}^{1,2}$ and (1+1) EA_b is as follows. As already mentioned, we consider only second-best matchings. Starting from such a search point, with overwhelming probability, $O(m^3)$ steps are enough to obtain either the perfect matching or an augmenting path of maximal length, which is ℓ according to Lemma 6.11. We estimate the probability of which of these two events happens first. If the augmenting path has reached length ℓ, 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 ℓ be polynomial in m. We are mainly interested in the case $3 \leq h \leq \ell$, implying that $\ell = \Omega(m^{1/3})$. Then, 2^ℓ is exponential in m. In a phase of this length, it is quite likely that (1+1) EA performs certain steps of exponentially small probability, which can change the current matching at a significant number of places and not only locally. Therefore, the following analysis will again be easier to conduct for $\mathrm{RLS}_\mathrm{b}^{1,2}$.

We concentrate now on specific Pvalues. If the Pvalue is 1, i.e., we have a free edge, it is likely we will find the perfect matching in the next step.

Lemma 6.12. *For $RLS_\mathrm{b}^{1,2}$ and (1+1) EA_b starting with an almost perfect matching with a Pvalue of* 1*, the following holds. The probability of reaching an almost perfect matching with a Pvalue of at least* 3 *is $\Theta(h/m)$.*

Proof. Since $P = 1$, the augmenting path consists of a free edge. To improve the matching, it is sufficient that only the free edge flip, and it is necessary that this edge flip. Therefore, the probability of creating the perfect matching is $\Theta(1/m)$. To increase P, it is sufficient that one of the h or $2h$ edge pairs lengthening the augmenting path flip. (If also the free 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 augmenting path.) Hence, the unconditional probability of reaching a situation where $P \geq 3$ equals $\Theta(h/m^2)$. The conditional probability of reaching $P \geq 3$ rather than $P = 0$ is, therefore, $\Theta(h/m)$. □

We prove the worst-case result for $\mathrm{RLS}_\mathrm{b}^{1,2}$ first. Assuming a Pvalue of at least 3, we will apply the results of the gambler's ruin theorem from Theorem 4.8. From the perspective of a lower bound, increasing the Pvalue to its maximum before improving the matching is a success. Hence, increasing P by 2 (recall that only odd values are taken) corresponds to the gambler's winning a unit of money and decreasing P corresponds to his losing a unit. The probability p_h of winning is at least $h/(h+2)$ if the Pvalue is at least 3, pessimistically assuming that one endpoint of the augmenting path is at a border of $G_{h,\ell}$. Since only $h \geq 3$ is considered, we have $p_h \geq 3/5$, i.e., an unfair game. More generally, we obtain $r_h = (1-p_h)/p_h \leq 2/h$ for the setup of Theorem 4.8. Given an initial Pvalue of $P_0 \geq 3$, the probability of reaching a Pvalue of ℓ before a value of at most 1 is at least

$$1 - \frac{r_h^{\ell'} - r_h^{P_0'}}{r_h^{\ell'} - 1} = \frac{r_h^{P_0'} - 1}{r_h^{\ell'} - 1} = \frac{1 - r_h^{P_0'}}{1 - r_h^{\ell'}},$$

where $\ell' := \lfloor \ell/2 \rfloor$ is the number of different values greater than 1 the Pvalue can take and $P_0' := \lfloor P_0/2 \rfloor$ is the number of possible Pvalues greater than 1 and less than or equal to P_0. Since $r_h < 1$, the probability under consideration is at least $1 - r_h^{P'_0}$. Our considerations are summarized by the following lemma, which pessimistically assumes the matching to be improved once the Pvalue has dropped to 1.

Lemma 6.13. *For $RLS_b^{1,2}$ starting with an almost perfect matching with a Pvalue of $P_0 \geq 3$, the probability of constructing an augmenting path of maximal length before the perfect matching is at least* $1 - (2/h)^{\lfloor P_0/2 \rfloor}$.

For (1+1) EA_b, 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 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 from above by $O(1/m^4)$. Now, the only way of decreasing the Pvalue by 1 without losing the game is by flipping 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 basically to the same probabilities as in Lemma 6.13, but we have to take into account the probability of $\Theta(1/m^3)$ of turning a search point with a short augmenting path of length 3 into the perfect matching. We obtain the following result, which provides in essence the same bounds as the preceding lemma.

Lemma 6.14. *For (1+1) EA_b starting with an almost perfect matching with a Pvalue of $P_0 \geq 3$, the probability of constructing an augmenting path of maximal length before the perfect matching is at least* $1 - O(1/m) - ((2/h) + O(1/m))^{\lfloor P_0/2 \rfloor}$.

Proof. Since we pessimistically consider the event of a Pvalue of 1 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 in which the gambler loses one unit of money. Since 2-bits are necessary and sufficient, 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 value of r_h from the above analysis by $O(1/m)$.

In addition, there is a probability of $O(1/m^4)$ for each step that the game is immediately lost because the Pvalue changes by more than 1. If we can prove that the game is finished anyway within $O(m^3)$ steps with probability at least $1 - O(1/m)$, the probability of observing a step of probability of $O(1/m^4)$ until the end of the game is $O(1/m)$. This is accounted for by first the term $-O(1/m)$ in the bound of the lemma.

We are left with the claim on the number of steps until the end of the game. Using $p_h > 1/2$ in Theorem 4.8, we obtain $D_{\lfloor P_0/2 \rfloor} = O(\ell)$, which means that the expected number of steps of (1+1) EA_b is $O(\ell m^2) = O(m^3)$ since a step is relevant, i.e., changes the length of the path, with probability $\Omega(m^2)$. The bound $O(m^3)$ holds also with probability $1 - 2^{-\Omega(m)}$. By Chernoff bounds, there is with probability $1 - 2^{-\Omega(m)}$ a surplus of at least $\lfloor \ell/2 \rfloor$ increasing steps within some cm relevant steps, c an appropriate constant. Also by Chernoff bounds, there are with probability $1 - 2^{-\Omega(m)}$ at least cm relevant steps within $c'm^3$ steps of (1+1) $\mathrm{EA_b}$, c' another appropriate constant. This proves the lemma. □

Putting the previous arguments presented until Lemma 6.13 together, we obtain a first lower bound on the runtime of $\mathrm{RLS}_\mathrm{b}^{1,2}$.

Lemma 6.15. *Starting with an almost perfect matching and an augmenting path of maximal length, the probability that $RLS_b^{1,2}$ finds the perfect matching within $2^{c\ell}$ steps, $c > 0$ an appropriate constant, is bounded from above by $2^{-\Omega(\ell)}$.*

Proof. We essentially apply the argumentation leading to Lemma 6.13. Starting from a Pvalue of ℓ, the value $k := \lceil \ell/2 \rceil = \Omega(\ell)$ or, if k is even, the value $k-1$ has to be taken at least once before 0 is reached. Starting from $k-1$ (analogously for k), we have an unfair game where the probability of the gamber not winning (where winning means returning to a Pvalue of ℓ) is bounded from above by

$$\frac{1 - r_h^{\lfloor (k-1)/2 \rfloor}}{1 - r_h^{\lfloor \ell/2 \rfloor}} \leq 2^{-c'\ell}$$

for some constant $c' > 0$. The game is repeated until it is lost (i.e., a Pvalue of at most 1 is reached) for the first time. The probability of losing at least once in $2^{c\ell}$ games is bounded from above by $2^{(c-c')\ell} = 2^{-\Omega(\ell)}$ if the constant $c < c'$ is chosen small enough. □

To prove a corresponding result for (1+1) $\mathrm{EA_b}$, we still exploit the fact that the game is unfair, i.e., there is a drift towards increasing the Pvalue. However, we cannot apply the gambler's ruin theorem any longer since, as mentioned above, exponentially long phases allow for steps that change the situation in a significant number of places. Therefore, the simplified drift theorem (Theorem 4.9) will be applied.

Lemma 6.16. *Starting with an almost perfect matching and an augmenting path of maximal length, the probability that (1+1) EA_b finds the perfect matching within $2^{c\ell}$ steps, $c > 0$ an appropriate constant, is bounded from above by $2^{-\Omega(\ell)}$.*

Proof. To apply Theorem 4.9, we set $a := 0$ and $b := \lceil \ell/2 \rceil - 1$. The random variables X_t are obtained by taking the random Pvalues at the respective time points, dividing them by 2 and rounding the result up. In this way, we obtain a process on the state space $\{0, 1, \ldots, \lceil \ell/2 \rceil\}$.

Given a current $X_t$value of i, where $i \leq \lceil \ell/2 \rceil - 1$, we need an estimate of the expected change of this value. The probability of increasing the value by 1, i.e., lengthening the augmenting path of length $2i - 1$ by 2, is bounded from below by

$$p_1(i) \;\geq\; \frac{h}{m^2}\left(1 - \frac{1}{m}\right)^{m-2}$$

since at least one end of the path is not at a border of $G_{h,\ell}$ and there are h appropriate 2-bit flips. Here we use the fact that $i \leq \lceil \ell/2 \rceil - 1$, i.e., the augmenting path can still be lengthened. On the other hand, the probability of decreasing the $X_t$value by $j \geq 2$ is bounded from above according to

$$p_{-j}(i) \;\leq\; (j+1) \cdot \left(\frac{1}{m}\right)^{2j}$$

since it is necessary to flip the $2k$ leftmost edges and the $2(j-k)$ rightmost edges of the augmenting path for some $k \in \{0, \ldots, j\}$. For $p_{-1}(i)$, we need a better bound that is at least by a constant factor smaller than $p_1(i)$. We estimate

$$p_{-1}(i) \;\leq\; \frac{2}{m^2}\left(1 - \frac{1}{m}\right)^{m-2} + \frac{3}{m^4}$$

since there are exactly two ways of flipping exactly two edges, and otherwise one has to flip at least the $2k$, $0 \leq k \leq 2$, leftmost edges and the $4 - 2k$ rightmost edges of the augmenting path.

Since most other mutations of (1+1) $\mathrm{EA_b}$ will be rejected in this setting due to worse fitness, we use the condition C_{rel} that a step is *relevant*, meaning it is accepted and changes the matching. Of course, if we obtain a lower bound on the required number of relevant steps, this also bounds the actual number of steps of (1+1) $\mathrm{EA_b}$ from below. The probability p_{rel} of a relevant step is bounded according to

$$\frac{1}{m^2}\left(1 - \frac{1}{m}\right)^{m-2} \;\leq\; p_{\mathrm{rel}} \;\leq\; \frac{2h+2}{m^2}.$$

The lower bound holds because, unless the optimum has been found, there always are two edges that, when flipped, lengthen or shorten the augmenting path. The upper bound holds because there are at most $2(h+1) = 2h+2$ couples of edges adjacent to a border of the augmenting path that, when flipped, lengthen or shorten the path. The probability that more than two bits flip and the step is relevant is lower since at least one of the $2h+2$ couples considered in the bound has to be flipped anyway.

Let $R(i) = (\Delta(i) \mid C_{\mathrm{rel}})$ denote the random increase of the $X_t$value in relevant steps, given a current value of i. We first concentrate on the contribution of steps of length 1, i.e., we consider $R_1(i) := R(i) \cdot \mathbb{1}\{|R(i)| \leq 1\}$. Thus,

$$\begin{aligned} E(R_1(i)) \;&=\; \frac{p_1(i)}{p_{\mathrm{rel}}} - \frac{p_{-1}(i)}{p_{\mathrm{rel}}} \;\geq\; \frac{\frac{h}{m^2} \cdot \left(1 - \frac{1}{m}\right)^{m-2}}{\frac{2h+2}{m^2}} - \frac{\frac{2}{m^2} \cdot \left(1 - \frac{1}{m}\right)^{m-2} + \frac{3}{m^4}}{\frac{2h+2}{m^2}} \\ &=\; \frac{(h-2)\left(1 - \frac{1}{m}\right)^{m-2}}{2h+2} - \frac{3}{m^2(2h+2)} \;\geq\; \frac{1}{8 \cdot e} - O(m^{-2}) \end{aligned}$$

since $h \geq 3$. The unconditional decrease $\Delta^-_{>1}(i) = -\Delta(i) \cdot \mathbb{1}\{\Delta(i) < -1\}$, for negative steps of length greater than 1, is in expectation at most

$$\begin{aligned} E(\Delta^-_{>1}(i)) \;&\leq\; \sum_{j=2}^{\infty} j \cdot p_{-j}(i) \;\leq\; \sum_{j=2}^{\infty} j \cdot (j+1) \cdot \frac{1}{m^{2j}} \\ &\leq\; \frac{6}{m^4} + \sum_{j=3}^{\infty} \frac{2m^2}{m^{2j}} \;=\; O(m^{-4}) \end{aligned}$$

using $p_{-j} \leq (j+1)/m^{2j}$. Hence, the total conditional drift is

$$\begin{aligned} E(R(i)) \;&\geq\; E(R_1(i)) - \frac{E(\Delta^-_{>1}(i))}{p_{\mathrm{rel}}} \;\geq\; \frac{1}{8 \cdot e} - O(m^{-2}) - O(m^{-4}) \cdot em^2 \\ &=\; \frac{1}{8 \cdot e} - O(m^{-2}), \end{aligned}$$

which is bounded from below by a constant such that the first condition of Theorem 4.9 has been established.

The second condition follows with $\delta = 1$ and $r = 8$ from

$$\frac{p_{-j}}{p_{\mathrm{rel}}} \;\leq\; \min\left\{1, \frac{j+1}{m^{2j}} \cdot em^2\right\} \;\leq\; \min\left\{1, \frac{1}{m^{2j-7}}\right\} \;\leq\; 8 \cdot \left(\frac{1}{2}\right)^j$$

for $m \geq 2$. From Theorem 4.9, the lemma follows. □

We summarize our results. Note that the exponentially small failure probability $2^{-\Omega(\ell)} = 2^{-\Omega(m^{1/3})}$ from Lemma 6.15 is captured by the $O(1/m)$ term of the following lemma.

Theorem 6.17. *Starting with an almost perfect matching and an augmenting path of length $2k+1$, the probability that (1+1) EA$_b$ finds the perfect matching within $2^{c\ell}$ steps, $c > 0$ an appropriate constant, is bounded from above by $O(1/m) + ((2/h) + O(1/m))^k$ if $3 \leq h \leq \ell$ and $k \geq 1$. For RLS$_b^{1,2}$, the bound $2^{-\Omega(\ell)} + (2/h)^k$ holds.*

So far, we have only considered the case of almost perfect matchings and shown that it can take exponential time to achieve the final improvement. We return to the question of whether an almost perfect matching will be reached.

Lemma 6.18. *If (1+1) EA$_b$ or RLS$_b^{1,2}$ do no 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 where M is not even a matching. Then, any step producing an almost perfect matching will be accepted.

For (1+1) EA_b, the probability of producing 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 of producing 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. Then the step where everything works as 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 $|M^*|$ ways to produce an almost perfect matching by additionally flipping an M^* edge. Their probability is $\Theta(|M^*|/m^{d+1}) = \Theta(1/(hm^d))$. The ratio of the relevant probabilities is always at least $\Omega(1/h)$.

For $\text{RLS}_\text{b}^{1,2}$, 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 will be proven to be even smaller than the last probability since we investigate the next two steps.

Let us consider the case $d = 1$. Then we are only interested in the case where M is a superset of M^* since otherwise M would be 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 another edge of M flip, an almost perfect matching is obtained. This happens with probabiliy $\Theta(|M^*|/m^2) = \Theta(1/(hm))$. The ratio is $\Omega(1/h)$.

Finally, assume $d = 2$. Then 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 free, and the first step produces an almost perfect matching with a probability of $\Theta(1/m)$ by flipping only one of these edges. 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. Finally, 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 the two $\overline{M}^*$ edges. Any step flipping e_1 and an arbitrary M^* edge in the first step will be accepted even though it leads still to a non-matching. The reason is that the penalty term in the underlying fitness function (cf. Section 6.1) decreases by at least 1 and optimization proceeds in lexicographic order. If the second step flips e_2, an almost perfect matching is obtained. Ths probability of these events is $\Theta((|M^*|/m) \cdot (1/m)) = \Theta(1/(hm^2))$, and the ratio of the relevant probabilities is again bounded from below by $\Omega(1/h)$. □

Taking the previous lemma, Lemma 6.12, and Theorem 6.17 together, we obtain that the $2^{\Omega(\ell)}$ bound of the theorem holds with a probability of $\Omega(1/(hm))$ if we start with any search point which is not the optimum. If $h \leq \ell$ then it holds that $\ell = \Omega(m^{1/3})$, and if h is a constant, then $\ell = \Omega(m)$. Altogether, given a initial search point that is not the optimum, we obtain an exponential lower bound of $2^{\Omega(\ell)} = 2^{\Omega(m^{1/3})}$ for the expected optimization time. This is summarized by the following theorem.

Theorem 6.19. *For $G_{h,\ell}$, $3 \le h \le \ell$, the expected optimization time of $RLS_b^{1,2}$ and (1+1) EA_b is $2^{\Omega(\ell)}$ if the initial search point is not the perfect matching.*

For example, if the initial search point is drawn uniformly at random, the probability of not starting with the perfect matching is $1-2^{-\Omega(m)}$. In general, the precondition of not starting with the optimum is the weakest condition one can think of. Early analyses of simulated annealing for the maximum matching problem (Sasakik and Hajek, 1988) are based on the deterministic choice of the empty matching as initial starting point. Theorem 6.19 is far less restrictive in this sense.

However, Theorem 6.19 contains a result on the expected optimization time, only. This statement goes back to Lemma 6.12 and Theorem 6.17, which imply a lower bound of $\Omega(1/(hm))$ on the probability of observing an exponential optimization time. Giel and Wegener (2004) improve upon this bound and show that an exponential time holds with probability $1 - 2^{-\Omega(\ell)}$ if $h = \omega(\log n)$. Very careful analyses are required to show these improved results, and the interested reader is referred to the works by Giel and Wegener (2003, 2004, 2006).

Conclusions

In this chapter, we have analyzed the simple search algorithms $\mathrm{RLS}_\mathrm{b}^{1,2}$ and (1+1) EA for the maximum matching problem. Optimal solutions are found on simple graph classes like paths in expected polynomial time. More generally, solutions that are only by a factor $1 + \epsilon$ away from optimality can be found in expected polynomial time. This proves that the algorithms are polynomial-time randomized approximation schemes (PRASs) for the problem. Consistently with this result, the limits of the search algorithms have been determined. On a worst-case graph, the expected time until the optimal solution is found was proven to be exponential.

The analyses make use of the techniques presented in Section 4.2. Most notably, the gambler's ruin theorem and the drift theorem were used to investigate the stochastic processes behind the algorithms.