Online Maximum Matching with Recourse

Spyros Angelopoulos

Sorbonne Université, CNRS, Laboratoire d'informatique de Paris 6, LIP6, F-75252 Paris, France
Spyros.Angelopoulos@lip6.fr
https://orcid.org/0000-0001-9819-9158

Christoph Dürr

Sorbonne Université, CNRS, Laboratoire d'informatique de Paris 6, LIP6, F-75252 Paris, France
Christoph.Durr@lip6.fr
https://orcid.org/0000-0001-8103-5333

Shendan Jin

Sorbonne Université, CNRS, Laboratoire d'informatique de Paris 6, LIP6, F-75252 Paris, France
Shendan.Jin@lip6.fr
https://orcid.org/0000-0003-1218-9085

Abstract

We study the online maximum matching problem in a model in which the edges are associated with a known recourse parameter k. An online algorithm for this problem has to maintain a valid matching while edges of the underlying graph are presented one after the other. At any moment the algorithm can decide to include an edge into the matching or to exclude it, under the restriction that at most k such actions per edge take place, where k is typically a small constant. This problem was introduced and studied in the context of general online packing problems with recourse by Avitabile *et al.* [1], whereas the special case $k = 2$ was studied by Boyar *et al.* [3].

In the first part of this paper, we consider the *edge arrival* model, in which an arriving edge never disappears from the graph. Here, we first show an improved analysis on the performance of the algorithm AMP given in [1], by exploiting the structure of the matching problem. In addition, we extend the result of [3] and show that the greedy algorithm has competitive ratio 3/2 for every even k and ratio 2 for every odd k. Moreover, we present and analyze an improvement of the greedy algorithm which we call *L*-Greedy, and we show that for small values of k it outperforms the algorithm of [1]. In terms of lower bounds, we show that no deterministic algorithm better than $1 + 1/(k-1)$ exists, improving upon the lower bound of $1 + 1/k$ shown in [1].

The second part of the paper is devoted to the *edge arrival/departure model*, which is the fully dynamic variant of online matching with recourse. The analysis of *L*-Greedy and AMP carry through in this model; moreover we show a lower bound of $\frac{k^2-3k+6}{k^2-4k+7}$ for all even $k \geq 4$. For $k \in \{2, 3\}$, the competitive ratio is 3/2.

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 8; pp. 8:1–8:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
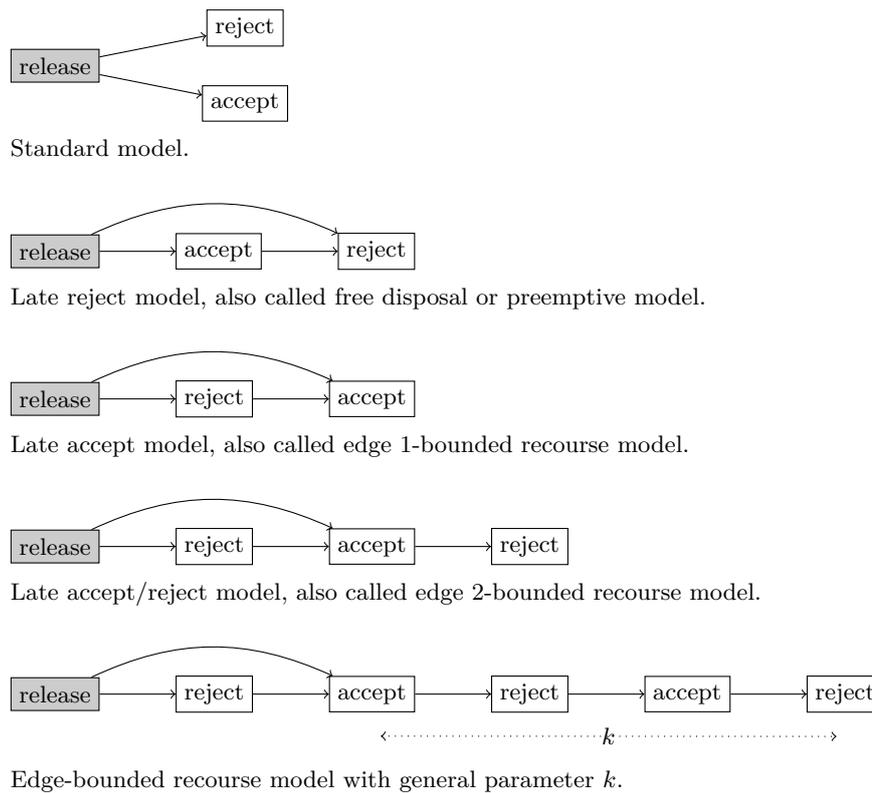
1 Introduction

In the standard framework of online computation, the input to the algorithm is revealed incrementally, i.e., request by request. For each such requested input item, the online algorithm must make a decision that is typically *irrevocable*, in the sense that the algorithm commits, in a permanent manner, to the decision associated with the request. More precisely, the algorithm may not alter any previously made decisions while considering later requests. This rather stringent constraint is meant to capture what informally can be described as "the past cannot be undone"; equally significantly, it is at the heart of adversarial arguments that can be used to argue that the competitive ratio of a given online problem cannot be improved beyond a certain bound.

Nevertheless, there are real-life applications in which some (limited) rearrangement of the online solution during the execution of the algorithm may be doable, or even requisite. For instance, online call admission protocols may sporadically reconfigure the virtual paths assigned in the network. For a different example, in online scheduling (or resource allocation) problems, it may be permissible for a job to be transfered to a processor other than the one specified by the original decision associated with the job. Clearly, a trade-off is to be found between the guaranteed competitive ratio and the cost of re-optimizing the current solution. Different approaches to this objective have been considered. One such approach has studied the minimum total re-optimization cost required in order to maintain an optimal solution, see Bernstein *et al.* [2]. Another approach has focused on the best achievable competitive ratio when there is some bound on the allowed re-optimization, which has been first studied by Avitabile *et al.* [1], and is the main model we consider in this paper.

More specifically, we study the *online maximum cardinality matching* problem, in which the goal is to maintain a vertex disjoint edge set of maximum cardinality for a given graph. Two different online models have been studied in the past. In the *vertex arrival model*, vertices arrive in online fashion, revealing, at the same time, the edges incident to previously arrived vertices. This model has mainly been considered for bipartite graphs, with left side vertices arriving online, and right side vertices being initially known (see the survey [16]). In the *edge arrival model* the edges arrive online in arbitrary order, revealing at the same time incident vertices. We emphasize that in this work we consider the maximum cardinality matching problem; some previous work (with or without recourse) has considered the generalized *weighted matching* problem, in which each edge has a weight and the objective is to maximize the weight of matched edges.

In the standard model, every edge constitutes a request and has to be immediately either accepted in the matching, or rejected. To quantify the impact of recourse, several models have been proposed that relax the irrevocable nature of a decision. In the *late reject* model [3], which is also called the *preemptive model* [5], an edge can be accepted only upon its arrival, but can be later rejected. In the *edge-bounded recourse* model, introduced in [1], the algorithm can switch between accepting and rejecting an edge that has already appeared, but is allowed up to k such modifications per edge. We emphasize that the initial default state of an edge is *rejected*, and therefore rejecting a newly arriving edge does not count as decision modification. However, accepting an edge or rejecting a previously accepted edge does count as a decision modification. For concreteness, we call this problem the *online maximum cardinality matching problem with edge-bounded recourse*. Boyar *et al.* [3] refer to this model for $k = 1$ as the *late accept* model, and for $k = 2$ as the *late accept/reject* model. Clearly, the competitive ratio is monotone in k, and our objective is to quantify this dependency. Figure 1 illustrates the algorithm's actions under the different models.

Figure 1 Illustration of the actions of an online matching algorithm under the different edge-arrival models with recourse.

In this work we also introduce and study the setting in which edges may undergo both arrivals and departures. In other words, edges may not only arrive (in the form of a request), but may also disappear adversarially (subsequently to their appearance). This setting is motivated by similar models that have been studied in the context of the online Steiner tree problem [9]. We call this problem the online maximum matching problem under the *edge arrival/departure model with edge-bounded recourse*. For this problem, we further distinguish two models concerning the edge departures. In the *full departure* model, the adversary is allowed to delete any edge in the graph, and thus also any edge that may have been provisionally accepted by the online algorithm. We show that this model is quite restrictive, since it yields excessive power to the adversary. We thus also study the *limited departure* model, in which the adversary may delete only edges not currently accepted by the online algorithm.

The limited departure setting can also model some natural applications related to resource allocation. For instance, consider a bipartite graph representing compatibility between tasks and workers, and that we seek to maximize the number of tasks assigned to workers. The compatibility of tasks and worker skills can change over time. Then the limited departure model stipulates that if worker w remains assigned to task t by the online algorithm, then w does not lose his qualification for t, in the sense that the worker has a continual occupation with the said task and maintains the required skills for the task. However, once the online algorithm decides to remove worker w from task t (i.e., the online algorithm provisionally rejects edge (w, t)), then the worker might lose his qualification for the task over time.

An online algorithm ALG for a maximization problem is said to be c-competitive if there is a constant d such that $\text{ALG}(\sigma) \geq \text{OPT}(\sigma)/c - d$, for all request sequences σ. Here $\text{ALG}(\sigma)$ stands for the objective value of the solution produced by ALG on σ, while $\text{OPT}(\sigma)$ stands for the value of the optimal solution. If $d = 0$, the algorithm is called *strictly c*-competitive. Note that some previous work on the matching problem has used the reciprocal ratio. The smallest c for which an online algorithm ALG is c-competitive is called the *competitive ratio* of ALG. The strict competitive ratio is defined similarly. If it so happens that this minimum value does not exist, the competitive ratio is actually defined by the corresponding infimum. In this setting an *upper bound* on the (strict or not) competitive ratio establishes the performance guarantee of an online algorithm, whereas a *lower bound* is a negative result.

Both upper and lower bounds in this work are shown for the strict competitive ratio. This implies that the upper bounds carry over to the more general definition, but this generalization does not necessarily hold for the lower bounds. We emphasize, however, that the known lower bounds for edge-bounded recourse problems in [1, 3] are likewise expressed in terms of the strict competitive ratio. This is due, perhaps, to difficulties in applying techniques that extend the lower bounds to the standard definition of the competitive ratio that are inherent to the recourse setting, and which do not arise in the traditional online framework of irrevocable decisions. Specifically, it is not obvious how to use techniques based on multiple copies of an adversarial instance in order to lower-bound the performance of any online algorithm, although this may be possible for specific online algorithms. For convenience, we will henceforth refer to the strict competitive ratio as simply the "competitive ratio".

Related work

Several online optimization problems have been studied under the recourse setting. The broad objective is to quantify the trade-off between the competitive ratio and a measure on the modifications allowed on the solution. Some representative examples include online problems such as minimum spanning trees and TSP [15], Steiner trees [9, 8], knapsack problems [11, 12], assignment problems in bipartite unweighted graphs [10], and general packing problems [1]. In the remainder of this section we review work related to online maximum matching.

Online matching in the standard model.   For online weighted matching in the standard model (without recourse), it is easy to see that no algorithm can achieve a bounded competitive ratio. This holds for both the vertex and the edge arrival models. For unweighted matching, the seminal work of Karp *et al.* [13, 7] gave a randomized online algorithm with competitive ratio $\text{e}/(\text{e} - 1)$ in the vertex arrival model together with a matching lower bound on any online algorithm. For the edge arrival model and the randomized competitive ratio, [4] showed a lower bound of $(3 + 1/\varphi^2)/2$, where φ is the golden ratio, as well as an upper bound of 1.8 for the special case of forests.

It is well known that any inclusion-wise maximal matching has cardinality at least half of the optimal maximum cardinality matching. From this it follows that the greedy online algorithm has competitive ratio at most 2, which in the standard model is optimal among all deterministic online algorithms.

Late reject.   In the vertex arrival model, the greedy algorithm achieves trivially the competitive ratio of 2, which is optimal for all deterministic online algorithms. The situation differs in the edge arrival model. Epstein *et al.* [6] showed that for online weighted matching,

the deterministic competitive ratio is exactly $3 + 2\sqrt{2} \approx 5.828$, as the upper bound of [14] matches the lower bound of [18]. The same paper [6] shows that the randomized competitive ratio is between $1 + \log 2 \approx 1.693$ and 5.356. Chiplunkar *et al.* [5] presented a randomized 28/15-competitive algorithm for trees and a 4/3-competitive algorithm for paths.

Edge k-bounded recourse. This model was introduced and studied by Avitabile *et al.* [1] for the edge arrival setting, in the context of a much broader class of online packing problems. They gave an algorithm, which we call AMP, that combines doubling techniques with optimal solutions to offline instances of the problem, which has competitive ratio $1 + O\left(\frac{\log k}{k}\right)$ (see Section 2.1 for an analysis of AMP). On the negative side, they showed that no randomized algorithm can be better than $1 + 1/(9k - 1)$-competitive; we note also that their construction implies a lower bound of $1 + 1/k$ for all deterministic algorithms.

Boyar *et al.* [3] showed that the deterministic competitive ratio is 2 for $k = 1$ and 3/2 for $k = 2$, and these optimal ratios are achieved by the greedy algorithm. Moreover, [3] studied several other problems for a value of the recourse parameter equal to 2, such as independent set, vertex cover and minimum spanning forest.

Minimizing recourse. Bernstein *et al.* [2] studied a different recourse model in which the algorithm has to maintain an optimal matching, while minimizing a recourse measure, namely the number of times edges enter or leave the matching maintained by the algorithm. They considered the setting of a bipartite graph and the vertex arrival model and showed that a simple greedy algorithm achieves optimality using $O(n \log^2 n)$ replacements, where n is the number of nodes in the arriving bipartition, whereas the corresponding lower bound for any replacement strategy is $\Omega(n \log n)$.

The results of Avitabile *et al.* [1] were originally formulated in a similar dual setting. More precisely, [1] asks the question: how big should the edge budget k be such that there is a $1 + \varepsilon$ competitive online algorithm that makes at most k changes per edge? They showed that $k = O(\log(1/\varepsilon)/\varepsilon)$ suffices.

Contribution of this work

In the first part of this work, we study the online matching problem with edge k-bounded recourse under the edge arrival model. For this problem, we provide improvements on both upper and lower bounds. First, we revisit the doubling algorithm of [1] that was originally analyzed in the general context of online packing problems. We give a better analysis, specifically for the problem at hand, that uses concepts and ideas related to the matching problem; we also show that the AMP algorithm has competitive ratio $1 + O(\frac{\log k}{k})$. On the negative side, we show that no deterministic algorithm is better than $1 + 1/(k - 1)$ competitive, improving upon the known bound of $1 + 1/k$ of [1].

At first sight these improvements may seem marginal; however one should take into consideration that k is typically a small parameter, and thus the improvements are by no means negligible. In this spirit, we propose and analyze a variant of the greedy algorithm which we call L-Greedy. This algorithm applies, at any step, augmenting paths as long as their length is at most $2L + 1$. We show that for a suitable choice of L, this algorithm is $(1 + O(1/\sqrt{k}))$-competitive. While this algorithm is thus not superior to AMP for large k (and more specifically, to its improved analysis in the context of the matching problem), for small k (and in particular, for $k \leq 20$) it does achieve an improved competitive ratio. Boyar *et al.* [3] showed that the greedy algorithm is 3/2-competitive for $k = 2$. We extended this result to all even k, while for odd k, the competitive ratio is 2.

In terms of techniques, we analyze both AMP and L-Greedy using amortization arguments in which the profit of the algorithms is expressed in terms of weights appropriately distributed over nodes in the graph. We achieve these improvements by exploiting properties of augmenting paths in matching algorithms.

The second part of the paper is devoted to the *edge arrival/departure model*, which is the fully dynamic variant of the online matching problem. First, we observe that the analysis of L-Greedy and AMP carries through in this model as well. On the negative side, we show a lower bound of $(k^2 - 3k + 6)/(k^2 - 4k + 7)$ for all even $k \geq 4$. For $k \in \{2,3\}$, the competitive ratio is $3/2$. We obtain the lower bounds by modeling the game between the algorithm and the adversary as a game played over *strings* of numbers 0 up to k.

We note that, for the analysis of AMP and of L-GREEDY, we assume that k is even. This assumption is borrowed from [1] and is required for the analysis. Of course for odd $k \geq 3$ these algorithms can be run with budget $k - 1$, providing a valid upper bound on the competitive ratio. Note that our lower bound in the arrival model holds for all values of k.

Due to space limitations some of the proofs are omitted in this paper.

1.1   Preliminaries

A *matching* in a graph $G = (V, E)$ is a set of edges $M \subseteq E$ with disjoint endpoints. A vertex $v \in V$ is said to be *matched* by M if there is an edge $e \in M$ incident to v, and is *unmatched* otherwise. A key concept in maximum matching algorithms is the notion of an *augmenting alternating path*, or simply *augmenting path*. A path P in G is a sequence of vertices $v_0, v_1, \ldots, v_\ell$ for some length $\ell \geq 2$, such that $(v_i, v_{i+1}) \in E$ for all $i = 0, \ldots, \ell - 1$. It is said to be *alternating with respect to M* if every other edge of P belongs to M. Moreover it is said to be *augmenting* if the first and the last vertex is unmatched by M. Applying such a path P to M consists in removing from M the edges in $M \cap P$ and adding the edges in $P \setminus M$. The resulting matching has cardinality $M + 1$, and every previously matched vertex remains matched.

We define some concepts that will be useful in the analysis of algorithms throughout the paper. We will associate each edge with a *type* which is an integer in $[0, k]$. An edge is of type i if it has undergone i decision flips by the algorithm. Hence, for an edge of type k, where k is the recourse budget, its decision has been finalized, and cannot change further; we call such an edge *blocked*. The type of a path P is defined by the sequence of the types of its edges, and to make this concept unambiguous, we choose between the two orientations of the path the one that results in the lexicographically minimal such sequence. Note that when the algorithm applies some augmenting path P to its current matching M, then the type of every edge in P is increased by 1. Moreover, the two extreme edges of an augmenting path are of type 0, because the endpoints of P are unmatched. We will call a path *blocked* if it contains a blocked edge.

2   The edge arrival model

2.1   The algorithm AMP

In the more general *online set packing problem* sets arrive online and the goal is to maintain a collection of disjoint sets, maximizing their number. Avitabile *et al.* [1] proposed the *doubling algorithm* which is defined for even k only. It has a parameter $r > 1$ and there is a decision variable for every set which can be changed at most k times. The algorithm works in phases, sequentially numbered by an integer p. Initially $p = 0$, and $\mathrm{ALG}_0 = \emptyset$. Let ℓ be the largest

integer such that the optimal solution has value at least r^ℓ, and let ℓ be $-\infty$ if the optimal solution is empty. Whenever this value increases, the algorithms starts a new phase. We define $\ell(p)$ as the value ℓ during phase p. We have $\ell(p) + i \leq \ell(p + i)$ for every positive integer i. At the beginning of a new phase, all decision variables that have been changed fewer than k times are set as in OPT, resulting in the current solution ALG_p (note that the algorithm crucially depends on k being even in order to produce a feasible solution).

Avitabile *et al.* show that the competitive ratio of the algorithm is at most

$$\min_{r>r_0} \frac{r^k(r-1)}{r^{k-1}(r-1) - r}, \tag{1}$$

where r_0 is the solution to the equation $r^{k-1}(r-1) - r = 0$ in $[1, +\infty)$.

We will show how to obtain an improved analysis of the algorithm in the context of the matching problem. Since we know optimal algorithms for $k = 1, 2$ [3] and for $k = 3$ (see Section 2.4), for the analysis we assume $k \geq 4$. We begin by a restatement of the update phase that will help us exploit the structure of solutions obtained via augmenting paths. More specifically, on every edge arrival the algorithm updates a current optimal solution OPT.

At the beginning of a new phase, the algorithm produces a matching ALG_p obtained from ALG_{p-1} as follows: every edge $e \in \text{ALG}_{p-1} \setminus \text{OPT}$ is removed from the current matching, and every edge $e \in \text{OPT} \setminus \text{ALG}_{p-1}$ which is of type strictly smaller than k is added to the current matching. Note that edges adjacent to e have been removed, hence ALG_p is indeed a matching. Also note that all edges added or removed by the algorithm have their type increased by one.

Since ALG_{p-1} and OPT are matchings, their symmetric difference, excluding type k edges, consists of alternating cycles and alternating paths which can be of even or odd length. This means that the algorithm simply applies at the beginning of every phase all those alternating paths and cycles.

We start the analysis of the competitive ratio by bounding OPT during phase p as

$$r^{\ell(p)} \leq \text{OPT} < r^{\ell(p)+1}, \tag{2}$$

which follows by the definition of phases.

During the phase $p \geq 1$ the competitive ratio is OPT/ALG_p. The type of an edge increases by 1 at most with each phase. Hence in the beginning of the k first phases the algorithm synchronizes with OPT as there are no blocked edges yet, and as a result during these phases the ratio is 1 at the beginning and does not exceed r by the upper bound in (2).

For the remaining phases we need the following argument.

▶ **Proposition 1.** *For even k and any phase $p \geq k + 1$, AMP maintains a matching ALG_p of size at least $r^{\ell(p)} - r^{\ell(p-k+1)+1}$.*

Proof. We denote by the *type of a vertex* v the maximum type of the edges adjacent to v, and by $n_{i,p}$ the number of vertices of type i in phase p. In addition, we denote by OPT_p the value of OPT at the beginning of phase p. With every new phase the type of a vertex can increase at most by 1. Hence every vertex of type k in phase p had positive type in phase $p - k + 1$. Thus

$$n_{k,p} \leq \sum_{i=1}^{k} n_{i,p-k+1} \leq 2 \cdot \text{OPT}_{p-k+1},$$

where the last inequality uses the fact that the left hand side counts the number of vertices matched by the algorithm. In phase p, the difference between the optimal matching and the matching of the algorithm is at most the number of blocked augmenting paths, and each of them contains at least two type k vertices. Hence

$$
\begin{aligned}
ALG_p &\geq \mathrm{OPT}_p - \frac{1}{2} \cdot n_{k,p} \\
&\geq \mathrm{OPT}_p - \mathrm{OPT}_{p-k+1} \\
&> r^{\ell(p)} - r^{\ell(p-k+1)+1}.
\end{aligned}
$$

The last inequality holds since $r^{\ell(p)} \leq \mathrm{OPT}_p < r^{\ell(p)+1}$.   ◀

Combining this proposition with the bounds (2) we obtain the following bound.

▶ **Proposition 2.** *The competitive ratio of AMP for $k \geq 4$ is upper bounded by the expression*

$$
\min_{r>1} \frac{r^k}{r^{k-1} - r}. \tag{3}
$$

In addition we can show that this is a better upper bound.

▶ **Proposition 3.** *For all even $k \geq 4$, Expression* (1) *upper bounds Expression* (3)*.*

The following theorem concludes the asymptotic analysis of the performance of AMP.

▶ **Theorem 1.** *For all even k, AMP has competitive ratio $1 + O(\frac{\log k}{k})$.*

Proof sketch. We first sketch a simple argument based on the Puiseux series expansion [17]: this is a type of power series that allows fractional powers, as opposed to only integer ones (e.g., Taylor series). In the full version we provide a second proof that relies only on standard calculus. Let r denote the optimal choice of the parameter, namely the one that minimizes (3). By analyzing the derivative, it follows that $r = (k-1)^{1/(k-2)}$, hence the competitive ratio is at most $\frac{(k-1)^{\frac{k-1}{k-2}}}{k-2}$, whose Puiseux series expansion at $k = \infty$ is $1 + \frac{\log k + 1}{k} + O(\frac{1}{k^2})$.   ◀

2.2   The algorithm Greedy

We consider the algorithm GREEDY, which repeatedly applies an arbitrary augmenting path whenever possible. This algorithm achieves an upper bound of $3/2$ for $k = 2$ as has been shown in [3]. We show that the same guarantee holds for all even k. In what concerns the lower bound, the idea is to force the algorithm to augment an arbitrarily long path in order to create a configuration with an arbitrarily large number of blocked augmenting paths of lengths 5, which locally have ratio $3/2$.

▶ **Proposition 4.** *The competitive ratio of* GREEDY *is $3/2$ for every even k and 2 for every odd k.*

2.3   The algorithm L-Greedy

The greedy algorithm has inferior performance because it augments along arbitrarily long augmenting paths, therefore sometimes sacrificing edge budget for only a marginal increase in the matching size. A natural idea towards an improvement would be to apply only short augmenting paths, as they are more budget efficient. For technical reasons, we restrict the choice of augmenting paths even further.

We define the algorithm L-GREEDY for some given parameter L, which applies any non-blocked augmenting path of length at most $2L+1$ that is in the symmetric difference between the current matching and some particular optimal matching OPT. The latter is updated after each edge arrival by applying an augmenting path for OPT. Note that L-GREEDY may not change its solution even if there is a short augmenting path in the current graph if it contains edges which are not in this particular optimal matching OPT. We will later optimize the parameter L as function of k.

2.3.1 Analysis of L-Greedy

We begin by observing that for $L = 0$ the algorithm collects greedily vertex disjoint edges without any recourse, which is precisely the behavior of GREEDY for $k = 1$ and has competitive ratio 2. For $L = 1$ the algorithm L-GREEDY applies only augmenting paths of length at most 3. In this case, the same argument as in the proof of Proposition 4 shows that the competitive ratio of L-GREEDY is $3/2$.

In what follows we analyze the general case $L \geq 2$. To this end, we assign weights to vertices in such a way that the total vertex weight equals the size of the current matching. Therefore, whenever the size of the matching is increased by 1, a total weight of 1 is distributed on the vertices along the augmenting path. Vertices in this path that were already matched receive a weight α, where $\alpha \geq 0$ is some constant that we specify later. Finally, the two vertices on the endpoints of the augmenting path receive the remaining weight, that is $1/2 - \ell\alpha$, where $2\ell + 1$ is the length of the path. It follows, from this weight assignment, that every unmatched vertex has weight 0, that every matched vertex has weight at least $1/2 - L\alpha$, and that every endpoint of a type k edge has weight at least $1/2 - L\alpha + (k-1)\alpha$.

Suppose that the online algorithm reaches a configuration in which it cannot apply any augmenting path, as specified in its statement. We consider the symmetric difference between the matching produced by the algorithm and the optimal matching maintained internally by the algorithm. This symmetric difference consists of alternating paths and/or alternating cycles, and we will upper bound for each such component the ratio between the number of edges in the optimal matching and the total vertex weight, which we call the *local ratio*. In particular, a component in the symmetric difference falls in one of the following cases: Either it is an augmenting path of length $2\ell + 1 \leq 2L + 1$, or an augmenting path of length $2\ell + 1 > 2L + 1$, or an alternating cycle or alternating path of even length.

Case 1: Augmenting path of length $2\ell + 1 \leq 2L + 1$. Such a path contains at least one edge of type k. It follows that $\ell \geq 2$, since an augmenting path of length 1 is a single type 0 edge, and an augmenting path of length 3 has edge types respectively $0, t, 0$ for some odd t (and k is assumed to be even). The path contains 2ℓ matched vertices, and at least 2 of them are adjacent to a type k edge. Hence the total vertex weight is at least $2\ell\left(\frac{1}{2} - L\alpha\right) + 2(k-1)\alpha$, and the local ratio of this component is at most

$$\frac{\ell + 1}{\ell - 2\ell L\alpha + 2(k-1)\alpha}. \tag{4}$$

Case 2: Augmenting path of length $2\ell + 1 > 2L + 1$. Such a path contains 2ℓ matched vertices and therefore the local ratio is at most

$$\frac{\ell + 1}{\ell - 2\ell L\alpha}. \tag{5}$$

Case 3: Alternating cycle or path of even length.   Such a component contains 2ℓ matched vertices and therefore the local ratio is at most

$$\frac{\ell}{\ell - 2\ell L\alpha}, \tag{6}$$

which is dominated by (5). We choose α so as to minimize the maximum of the local ratios, as defined by (4) and (5). Then for this choice of α we optimize L; namely we choose the value $L = \lfloor \sqrt{k-1} \rfloor$. Note that for $k = 4$, this leads to the choice $L = 1$, which we analyzed in the beginning of the section. We obtain the following performance guarantee.

▶ **Theorem 2.** *The competitive ratio of L-*GREEDY *with $L = \lfloor \sqrt{k-1} \rfloor$ is at most*

$$\frac{k(L+2) - 2}{(L+1)(k-1)} = 1 + O\left(\frac{1}{\sqrt{k}}\right),$$

for even $k \geq 6$ and at most $3/2$ for $k = 4$.

One can show that this analysis of L-GREEDY is essentially tight.

2.4   Lower bound for deterministic algorithms

Boyar *et al.* [3] show that the deterministic competitive ratio of the problem is 2 for $k = 1$ and $3/2$ for $k = 2$. We complete this picture by showing a lower bound of $1 + \frac{1}{k-1}$ for all $k \geq 3$. Note that the lower bound is tight for $k = 3$, as the algorithm GREEDY, which works by assuming that k is only 2, has competitive ratio $3/2$.

▶ **Theorem 3.** *The deterministic competitive ratio of the online matching problem with edge-bounded recourse is at least $1 + \frac{1}{k-1}$ for all $k \geq 3$.*

Proof sketch. We consider three cases, namely the cases $k = 3$, k is even and at least 4, and finally k is odd and at least 5. For each case we present an appropriate adversarial argument. Due to space limitations we only present the case $k = 3$. Suppose, by way of contradiction, that for $k = 3$ some algorithm claims a competitive ratio strictly smaller than $(3n + 2)/(2n + 2)$ for some arbitrary $n \geq 1$. The adversary releases a single edge, creating an augmenting path of length 1. Then the algorithm applies the augmenting path, which the adversary extends by appending one edge on each side, creating an augmenting path of type 0,1,0, as shown in Figure 2(a). Since the current ratio is 2, the algorithm needs to apply this path, which the adversary again extends by appending an edge on each side, creating an augmenting path of type 0,1,2,1,0, as shown in Figure 2(b). Since the current ratio is $3/2$, the algorithm applies this path. In response the adversary appends an edge at each endpoint of the type 3 edge, and at each endpoint of one of the type 1 edges, as shown in Figure 2(c). The resulting graph has a blocked augmenting path of type 0,3,0, and an augmenting path of type 0,1,0, as shown in Figure 2(c). The algorithm needs to apply the latter one as the ratio is currently $5/3 > 3/2$.

At this point, the adversary repeats this construction $n - 1$ times, by identifying the shaded part of Figure 2(c) as the graph of Figure 2(a), and reapplying the above construction. The final graph consists of n blocked augmenting paths of type 0,3,0 and $n + 1$ edges of type 1 that belong both to the optimal and the algorithm's matchings. Figure 2(d) illustrates this final graph for $n = 4$. Hence the competitive ratio is $(3n + 1)/(2n + 1)$, contradicting the claimed ratio and showing a lower bound of $3/2$.   ◀
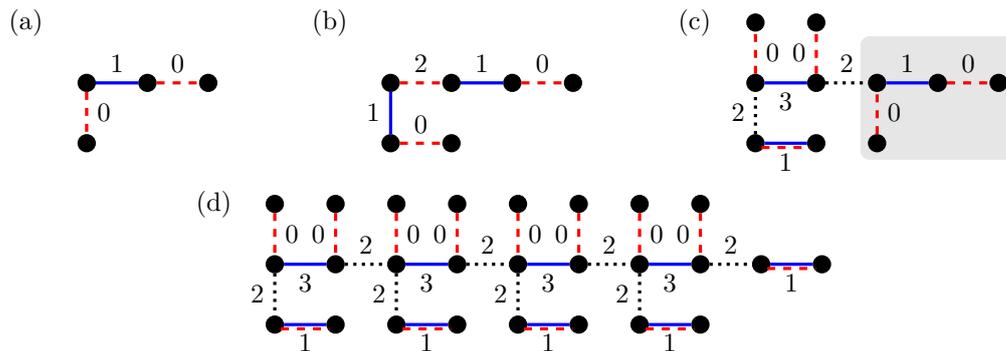
Figure 2 Lower bound construction for the case $k = 3$.

Table 1 Summary of lower bounds (LB) and upper bounds on the competitive ratio for the problem, for all even $4 \le k \le 22$. The lower bounds for the arrival/departure model are discussed in Section 3. The analysis of L-Greedy and AMP carry through to the (limited) edge arrival/departure model. For $k \ge 22$, the upper bound of AMP is superior to the upper bound of L-Greedy.

| k	LB (arr.)	LB (arr./dep.)	L-Greedy	AMP
4	1.333333	1.428571	1.5	2.598076
6	1.2	1.263158	1.466667	1.869186
8	1.142857	1.179487	1.428571	1.613602
10	1.111111	1.134328	1.333333	1.480583
12	1.090909	1.106796	1.318182	1.398080
14	1.076923	1.088435	1.307692	1.341500
16	1.066666	1.075377	1.300000	1.300080
18	1.058823	1.065637	1.247059	1.268330
20	1.052631	1.058104	1.242105	1.243150
22	1.047619	1.052109	1.238095	1.222640

2.5 Comparing the algorithms L-Greedy and AMP

We have analyzed two deterministic online algorithms: the algorithm AMP, which has competitive ratio $1 + O\left(\log(k)/k\right)$, and the algorithm L-Greedy, which has competitive ratio $1 + \Theta(1/\sqrt{k})$. Since the analysis of L-Greedy is tight, it follows that AMP is asymptotically (i.e., for large k) superior to L-Greedy. However, for small values of k, namely $k \le 20$, we observe that L-Greedy performs better, in comparison to the performance bound we have shown for AMP. These findings are summarized in Table 1 and Figure 3.

3 The edge arrival/departure model

In this section we consider the online matching problem in the setting in which edges may arrive but also *depart* online. In this context one can distinguish two models. In the *limited departure model* an edge cannot be removed from the instance while it is matched by the online algorithm, while in the stronger *full departure model* any edge can be removed.

It turns out that the latter model is quite restrictive. This is because it is possible for the adversary to force an online algorithm to augment some augmenting path and then to remove one of the edges in its matching. Eventually the algorithm can end up with blocked edges (type k), without having the chance to augment its matching. This intuition is formalized in the following lemma.
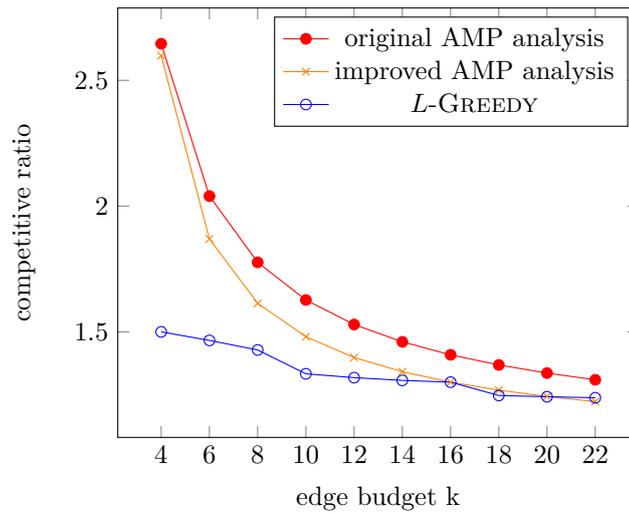
▶ **Lemma 4.** *The competitive ratio in the full departure model is* 2.

Since the full departure model is very restrictive for the algorithm, as shown in Lemma 4, we will concentrate on the limited departure model, as defined in the introduction. For this model, we observe that the algorithms *L*-GREEDY and AMP have the same performance guarantee as in the edge arrival model. This is because the analysis of *L*-GREEDY uses weights on vertices which are not affected by edge departures, and the analysis of AMP is based on an upper bound over the number of type k edges, which still holds under edge departures. We thus focus on obtaining stronger lower bounds in this model (also included in Table 1). We begin by observing that the bound of $3/2$ of the competitive ratio in the edge arrival model for $k \in \{2, 3\}$ still holds for the limited departure model, where the adversary is stronger. Hence, the smallest interesting value for k in this model is $k = 4$, for which we provide the following specific lower bound. The proof will also provide some intuition about the adversarial argument for general k.

▶ **Theorem 5.** *The competitive ratio in the* limited departure model *is at least* $10/7$ *for* $k = 4$.

Proof. We specify a particular adversary that maintains a graph such that the symmetric difference between the matching produced by the algorithm and the optimal matching consists only of augmenting paths and has no alternating cycles or alternating paths of even length.

The edge types along any path form a string over the integers $\{0, 1, \ldots, k\}$ with alternating parity and starting and ending with 0. We call these strings *alternating*. Thus, rather than a game played between the algorithm and the adversary on a *graph*, we will consider the game played on a *collection of alternating strings*.

Whenever the algorithm applies an augmenting path, this translates into the increment of each integer of the corresponding string, for example $01210 \to 12321$. The adversary responds to this action by three types of actions. First, the adversary may possibly split the string into smaller strings, for example $12321 \to 123, 1$ or $12321 \to 1, 3, 1$. This corresponds to deleting some edges which are currently not matched by the algorithm. Second, the adversary may possibly merge some of the resulting strings by concatenating them on both

sides of a 0, for example $1, 1 \rightarrow 101$. Finally the adversary may append 0's to the ends of the strings, where needed to make them alternating, for example $101 \rightarrow 01010$. The last two actions correspond to the insertion of (type 0) edges by the adversary.

The main idea is as follows. Consider an algorithm that claims a competitive ratio at most $(10 - \epsilon)/7$ for some sufficiently small $\epsilon > 0$. We will then show that the adversary can force the algorithm into a final configuration of ratio at least $10/7$, a contradiction.

We describe the current configuration by variables v, w, x, y, z which count the number of the specific strings that are described in Figure 4. In particular, z is the total number of strings 030 and 01410, which we call *bad strings*. The name is motivated by the fact that 01410 has worst local ratio among all blocked strings and is created from 030 strings.

The adversary starts by presenting the string 0, which the algorithm has to augment, resulting in a single string 010. This unique adversarial action has been omitted for simplification from Figure 4. From this moment onwards, we distinguish 3 different phases during the game, decided by the adversary, and depicted in Figure 4. For example, if the algorithm augments the string 01010, then in phase 1 the adversary will replace it with the strings 010 and 01210, while in phases 2 and 3 he will replace it with the string 0121210.

At a high level, the objective in phase 1 is to create a large enough number of bad strings, while phase 2 creates a large enough proportion of bad strings. Last, phase 3 leads the algorithm to a configuration which consists only of blocked strings.

The game starts with phase 1. During this phase the competitive ratio is at least $3/2$, thus forcing the algorithm to continue augmenting strings. After a finite number of steps, the condition $7z + 3 > 4/\epsilon$ holds and the game moves to phase 2. During the second phase we have the invariants (1) $7z + 3 > 4/\epsilon$ and (2) $2x + y + 2w \leq z + 1$. Invariant (1) holds because the left-hand side will not decrease throughout the phase. Moreover, Invariant (2) follows from the definitions of the involved parameters and the transitions between strings, as defined by the statement of the phase. Together both invariants imply that the competitive ratio is strictly larger than $(10 - \epsilon)/7$, forcing the algorithm to augment strings. This is because the competitive ratio can be lower bounded by

$$\frac{3z + 3y + 4(x + w)}{2z + 2y + 3(x + w)} \geq \frac{10z + 2y + 4}{7z + y + 3} \geq \frac{10z + 4}{7z + 3} \geq \frac{10 - \epsilon/2}{7},$$

where the first inequality follows from Invariant (2) and the last one from (1).

In phase 2, at some moment eventually the condition $z \geq 8(x + y + v + w)$ will hold since $x + y + v + w$ does not change, but any sequence of $x + y + 1$ steps increases z. At that moment, the game starts phase 3. We note that the condition is preserved during phase 3 and that it implies a ratio of at least $10/7$, forcing the algorithm to a configuration consisting only of the blocked strings 01410 and 012343210. This is because $z \geq 8(x + y + v + w)$ implies $z + y + x \geq 2w + 8v$, which in turn implies

$$\frac{3(x + y + z) + 4w + 6v}{2(x + y + z) + 3w + 5v} \geq \frac{10}{7},$$

where the left hand side lower bounds the competitive ratio in phase 3.   ◀

We can generalize the ideas in the proof of Theorem 5 so as to obtain a non-trivial lower bound for general even $k \geq 4$, see Figure 5.

▶ **Theorem 6.** *The competitive ratio in the limited departure model is at least $\frac{k^2 - 3k + 6}{k^2 - 4k + 7}$, for all even $k \geq 4$.*

Since $\frac{k^2 - 3k + 6}{k^2 - 4k + 7} > 1 + \frac{1}{k - 1}$ for all $k \geq 4$, Theorem 6 shows a stronger lower bound for even k than Theorem 3 under the limited departure model.
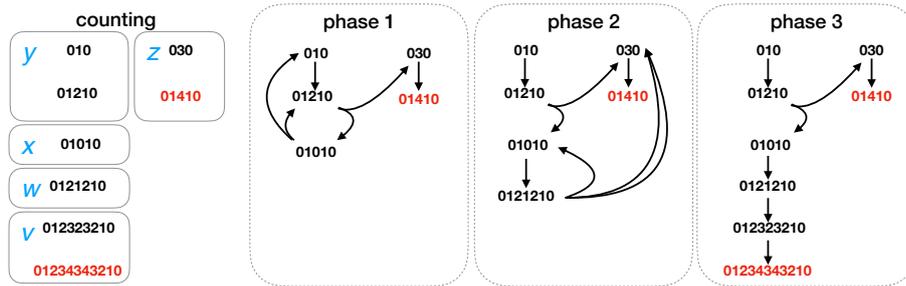
Figure 4 The lower bound construction in the arrival/departure model and $k = 4$.
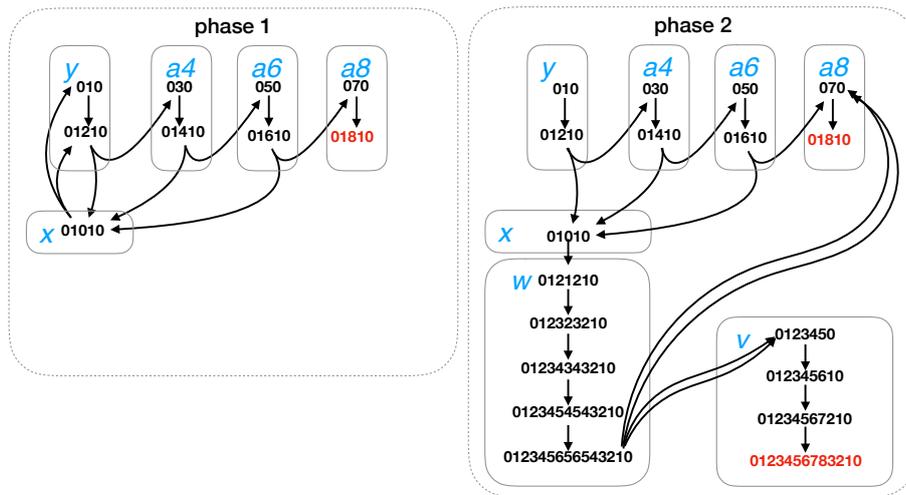
Figure 5 The lower bound construction in the arrival/departure model illustrated for $k = 8$. Blocked strings are depicted in red. The arcs illustrate the adversarial strategy. For example, if the algorithm augments the string 012345656543210, the adversary replaces the resulting string by two strings 070 and two strings 0123450.

References

1 Tess Avitabile, Claire Mathieu, and Laura H. Parkinson. Online constrained optimization with recourse. *Information Processing Letters*, 113(3):81–86, 2013.

2 Aaron Bernstein, Jacob Holm, and Eva Rotenberg. Online bipartite matching with amortized replacements. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 947–959. SIAM, 2018.

3 Joan Boyar, Lene M. Favrholdt, Michal Kotrbčík, and Kim S. Larsen. Relaxing the irrevocability requirement for online graph algorithms. In *Proceedings of the 15th Workshop on Algorithms and Data Structures, (WADS)*, pages 217–228. Springer, 2017.

4 Niv Buchbinder, Danny Segev, and Yevgeny Tkach. Online algorithms for maximum cardinality matching with edge arrivals. In *Proceedings of the 25th Annual European Symposium on Algorithms, (ESA), Vienna, Austria*, pages 22:1–22:14, 2017.

5 Ashish Chiplunkar, Sumedh Tirodkar, and Sundar Vishwanathan. On randomized algorithms for matching in the online preemptive model. In *Proceedings of the 23rd Annual European Symposium on Algorithms, (ESA)*, pages 325–336. Springer, 2015.

6 Leah Epstein, Asaf Levin, Danny Segev, and Oren Weimann. Improved bounds for online preemptive matching. In *Proceedings of the 30th International Symposium on Theoretical*

Aspects of Computer Science, (STACS), pages 389–399, 2013.

7   Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 982–991. Society for Industrial and Applied Mathematics, 2008.

8   Albert Gu, Anupam Gupta, and Amit Kumar. The power of deferral: maintaining a constant-competitive steiner tree online. *SIAM Journal on Computing*, 45(1):1–28, 2016.

9   Anupam Gupta and Amit Kumar. Online steiner tree with deletions. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA), Portland, Oregon, USA*, pages 455–467, 2014.

10 Anupam Gupta, Amit Kumar, and Cliff Stein. Maintaining assignments online: Matching, scheduling, and flows. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 468–479, 2014.

11 Xin Han and Kazuhisa Makino. Online minimization knapsack problem. In *Proceedings of the 7th International Workshop on Approximation and Online Algorithms, (WAOA)*, pages 182–193. Springer, 2009.

12 Kazuo Iwama and Shiro Taketomi. Removable online knapsack problems. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming, (ICALP)*, pages 293–305, 2002.

13 Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, (STOC)*, pages 352–358. ACM, 1990.

14 Andrew McGregor. Finding graph matchings in data streams. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, (APPROX-RANDOM)*, pages 170–181. Springer, 2005.

15 Nicole Megow, Martin Skutella, José Verschae, and Andreas Wiese. The power of recourse for online MST and TSP. *SIAM Journal on Computing*, 45(3):859–880, 2016.

16 Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

17 C.L. Siegel. *Topics in Complex Function Theory, Vol. 1: Elliptic Functions and Uniformization Theory.* New York: Wiley, 1988.

18 Ashwinkumar Badanidiyuru Varadaraja. Buyback problem-approximate matroid intersection with cancellation costs. In *Proceedings of the 38th International Colloquium on Automata, Languages, and Programming, (ICALP)*, pages 379–390. Springer, 2011.