Maximum Matching via Maximal Matching Queries

Christian Konrad
Department of Computer Science, University of Bristol, UK

Kheeran K. Naidu
Department of Computer Science, University of Bristol, UK

Arun Steward
Department of Computer Science, University of Bristol, UK

Abstract

We study approximation algorithms for Maximum Matching that are given access to the input graph solely via an *edge-query maximal matching oracle*. More specifically, in each round, an algorithm queries a set of potential edges and the oracle returns a maximal matching in the subgraph spanned by the query edges that are also contained in the input graph. This model is more general than the *vertex-query model* introduced by binti Khalil and Konrad [FSTTCS'20], where each query consists of a subset of vertices and the oracle returns a maximal matching in the subgraph of the input graph induced by the queried vertices.

In this paper, we give tight bounds for deterministic edge-query algorithms for up to three rounds. In more detail:

1. As our main result, we give a deterministic 3-round edge-query algorithm with approximation factor 0.625 on bipartite graphs. This result establishes a separation between the edge-query and the vertex-query models since every deterministic 3-round vertex-query algorithm has an approximation factor of at most 0.6 [binti Khalil, Konrad, FSTTCS'20], even on bipartite graphs. Our algorithm can also be implemented in the semi-streaming model of computation in a straightforward manner and improves upon the state-of-the-art 3-pass 0.6111-approximation algorithm by Feldman and Szarf [APPROX'22] for bipartite graphs.
2. We show that the aforementioned algorithm is optimal in that every deterministic 3-round edge-query algorithm has an approximation factor of at most 0.625, even on bipartite graphs.
3. Last, we also give optimal bounds for one and two query rounds, where the best approximation factors achievable are $1/2$ and $1/2 + \Theta(\frac{1}{n})$, respectively, where n is the number of vertices in the input graph.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Maximum Matching, Query Model, Algorithms, Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.STACS.2023.41

Funding C.K. is supported by EPSRC New Investigator Award EP/V010611/1. K.K.N. is supported by EPSRC DTP studentship EP/T517872/1.

1 Introduction

In this work, we study approximation algorithms for the Maximum Matching problem (MM) and its bipartite version, the Maximum Bipartite Matching problem (MBM), that are only given query access to the input graph $G = (V, E)$ via a *maximal matching oracle*. A *matching* $M \subseteq E$ in graph G is a subset of vertex-disjoint edges. The matching M is *maximum* if $|M| \geq |M'|$, for every other matching M'. The *matching number* $\mu(G)$ of a graph G is the size of a maximum matching. Furthermore, a matching M is *maximal* if it is inclusion-wise maximal, i.e., $M \cup \{e\}$ is not a matching, for every $e \in E \setminus M$. In each round i of the maximal matching edge-query model, the algorithm sends a set of potential edges $Q_i \subseteq V \times V$, denoted the query edges, to the oracle, which in turn responds with an arbitrary maximal matching in the subgraph $G[Q_i \cap E]$, i.e., the subgraph of G spanned by the query edges that are also contained in G. For brevity, we will refer to this model as the *edge-query model*.

Maximal Matching Queries. The study of algorithms for MM that solely execute maximal matching queries was introduced by binti Khalil and Konrad [8]. They considered a *vertex-query model*, where, in each round i, the algorithm queries a subset of vertices $U_i \subseteq V$, and the oracle responds with an arbitrary maximal matching in subgraph $G[U_i]$, i.e., the subgraph induced by vertices U_i. The edge-query model is more general than the vertex-query model since vertex queries can be simulated in the edge query model: For each query $U_i \subseteq V$ in the vertex-query model, querying the set of edges Q_i that turns U_i into a clique yields an equivalent edge-query algorithm.

The study of maximal matching query models is motivated by the fact that, in many computational models, including the data streaming model [20] and the Massively Parallel Computation model [17], computing maximal matchings is easy, while computing substantially larger matchings is more challenging. Computing maximal matchings can thus be regarded as a black-box subroutine, which allows for the design of matching algorithms that are independent of the underlying computational model.

For example, in the *semi-streaming model* for processing large graphs [12, 20], an algorithm makes few passes over the edges of the input graph in arbitrary order while maintaining a memory of size $O(n \operatorname{poly} \log n)$, where n is the number of vertices in the input graph. The GREEDY matching algorithm, which inserts every arriving edge into an initially empty matching if possible, i.e., if none of its endpoints are already matched, yields a maximal matching and constitutes a one-pass semi-streaming algorithm. Since a maximal matching is at least half the size of a maximum matching, GREEDY can also be regarded as a $\frac{1}{2}$-approximation semi-streaming algorithm for MM. While it is unknown whether it is possible to go beyond the approximation factor of $1/2$ in a single pass even on bipartite graphs (currently only approximation factors beyond $\frac{1}{1+\ln 2} \approx 0.59$ are ruled out [16]), improved results are known for multiple passes, and, indeed, most multi-pass semi-streaming algorithms solely execute GREEDY on carefully selected subgraphs in each pass (e.g. [9, 2, 18, 5]). This includes the state-of-the-art[1] $(1-\epsilon)$-approximation algorithm for MBM by Assadi et al. [5], which executes GREEDY $O(\frac{1}{\epsilon^2})$ times and thus runs in $O(\frac{1}{\epsilon^2})$ passes. This algorithm can easily be implemented in the Massively Parallel Computation model [17] and also constitutes the state-of-the-art result in this model. As such, maximal matching query models capture these algorithms and allow for a systematic study of what can and cannot be achieved.

Our Results. In this paper, we give tight approximation ratios for deterministic edge-query algorithms for MBM for up to three rounds. Our results for one and two rounds as well as the lower bound for three rounds also hold for MM. In Table 1, we illustrate our bounds and compare them to the respective tight bounds that holds in the vertex-query model [8].

One Round. We show that the best approximation factor achievable in a single round for MBM is $\frac{1}{2}$, which matches the vertex-query setting (Theorem 15-1). Querying all potential edges, i.e., the query $V \times V$, yields a matching upper bound, even in general graphs.

[1] We note that the algorithm by [5] yields a $(1-\epsilon)$-approximation in $O(\frac{1}{\epsilon^2})$ passes. Very recently, Assadi et al. [4] gave a $(1-\epsilon)$-approximation algorithms for MBM that operates in $O(\frac{1}{\epsilon} \cdot \log n)$-passes, which, for very small values of ϵ, asymptotically requires fewer rounds than [5].

Table 1 Optimal approximation ratios achievable for deterministic algorithms for MBM in the edge-query (this paper) and vertex-query models ([8]).

# Rounds	Vertex-query model ([8])	Edge-query model (this paper)
1	$\frac{1}{2}$	$\frac{1}{2}$ (Theorem 15-1)
2	$\frac{1}{2}$	$\frac{1}{2} + \Theta(\frac{1}{n})$ (Theorem 15-2)
3	$\frac{3}{5} = 0.6$	$\frac{5}{8} = 0.625$ (Theorems 1 and 15-3)

Two Rounds. The approximation factor can be very slightly improved in two rounds, even in general graphs. Consider the algorithm that queries all edges $V \times V$ in the first round, which produces a maximal matching M_1 in the input graph. Next, pick any edge $uv \in M_1$ and query all edges incident to u and v different to uv in the next round. Then, we will either find a 3-augmenting path that allows us to augment the edge uv, or the edge uv is not 3-augmentable. In both cases, we establish that the resulting matching is a $\frac{1}{2} + \Theta(\frac{1}{n})$-approximation, and we also prove that no algorithm can do better, even in bipartite graphs (Theorem 15-2). While the $\Theta(\frac{1}{n})$ additive term is not significant in terms of an improved approximation guarantee, it nevertheless illustrates that the edge-query and vertex-query models behave slightly differently in the two rounds setting.

Three Rounds. As our main result, we give a deterministic 3-round algorithm for MBM in the edge-query model that produces a 0.625-approximation (Theorem 1), and we show that this is best possible (Theorem 15-3). Our algorithm can be implemented in a straightforward way in the semi-streaming model and improves upon the previously best 3-pass semi-streaming 0.6111-approximation algorithm for MBM by Feldman and Szarf [13].

On 3-pass Semi-streaming Algorithms for MBM. The first 3-pass semi-streaming algorithm for MBM was implicit in [12], explicitly mentioned in [19], and analysed by Kale and Tirodkar [15], who showed that the approximation ratio is 0.6. Subsequently, binti Khalil and Konrad [8] proved that this algorithm constitutes an optimal 3-round vertex-query algorithm (and can thus also be implemented in the edge-query model). Various improvements have since been established via semi-streaming algorithms that cannot be implemented in the edge-query model. First, Esfandiari et al. [10] gave a 0.605-approximation algorithm, which was then further improved by Konrad [18] who gave a randomized 0.6067-approximation algorithm. Very recently, Feldman and Szarf [13] gave a 0.6111-approximation. In this paper, we improve the approximation factor to 0.625, again, with a 3-round deterministic edge-query algorithm.

While edge-query algorithms appear somewhat restricted in how they operate as compared to arbitrary semi-streaming algorithms, the literature illustrates that they are surprisingly powerful as they constitute the state-of-the-art algorithms in the 3-pass (this paper) and $(1-\epsilon)$-approximation [5] streaming settings for MBM.

Techniques. We will first discuss the ideas behind our 3-round algorithm for MBM in the edge-query model, and then give the intuition behind our lower bound results.

3-*round Query Algorithm.* Our 3-round algorithm computes a maximal matching in the first round, and then finds augmenting paths in the subsequent rounds. This is a well-established technique, and almost all known 2-pass and 3-pass streaming algorithms operate in this fashion (e.g. [19, 10, 15, 18, 13]). To this end, denote by M_1 a maximal matching in the bipartite input graph $G = (A, B, E)$ that we obtain by querying all potential edges $A \times B$. We observe that every augmenting path for M_1 starts with an edge in $G_L = G[A(M_1) \cup \overline{B(M_1)}]$

and ends with an edge in $G_R = G[\overline{A(M_1)} \cup B(M_1)]$, where $A(M_1)$ denotes the A-vertices matched in M_1, and $\overline{A(M_1)} = A \setminus A(M_1)$ ($B(M_1)$ and $\overline{B(M_1)}$ are defined similarly). In our second round, we therefore compute maximal matchings M_L and M_R in G_L and G_R, respectively. Observe that we can indeed compute both of these matchings with the single query $\left(A(M_1) \times \overline{B(M_1)}\right) \cup \left(\overline{A(M_1)} \times B(M_1)\right)$ in the edge-query model. At this stage, we are guaranteed that the set $M_1 \cup M_L \cup M_R$ contains various length-2 paths consisting of one edge of M_1 and one additional edge either from M_L or M_R, and the usual idea employed in the literature is to complete these length-2 paths to length-3 augmenting paths using an additional pass over the data/an additional query. Indeed, if we attempted to complete the length-2 paths by computing a maximal matching between the endpoints of length-2 paths in M_1 and the yet unmatched vertices in the third round then we obtain a 0.6-approximation.

Our key idea for the third query round that leads to an improvement over previous work is to simultaneously attempt to complete length-5 augmenting paths. To this end, denote by A' the endpoints of length-2 paths in $A(M_1)$, and by B' the endpoints of length-2 paths in $B(M_1)$. Our third round query consists of all potential edges interconnecting the vertices

$$A' \cup B' \cup \overline{A(M_1)} \cup \overline{B(M_1)} \ ,$$

i.e., we both attempt to complete length-2 paths to length-3 augmenting paths by considering the edges between A' and $\overline{B(M_1)}$ and between B' and $\overline{A(M_1)}$, but we also attempt to join two disjoint length-2 paths by connecting them via an edge between A' and B' to form a length-5 augmenting path. The main challenge in the analysis of this method is to address the complications that arise from the fact that the single maximal matching returned in round 3 completes both length-3 and length-5 augmenting paths.

Lower Bounds. The key idea behind our lower bound arguments is to keep track of the information revealed when the oracle returns a maximal matching M_i as a response to the query edges Q_i in round i. This approach was previously successfully employed by binti Khalil and Konrad [8] for obtaining optimal lower bounds in the vertex-query model. When a matching M_i is returned, the algorithm not only learns that the edges M_i are indeed contained in the input graph, but also that none of the edges of Q_i that connect vertices outside of $V(M_i)$ exist, which is due to the maximality of M_i in the subgraph $G[E \cap Q_i]$ of the input graph $G = (V, E)$. The main challenge lies in keeping track of the information revealed over the course of the algorithm while considering the complexity of *all* possible queries in each round. This is achieved by considering the vertex-induced subgraphs on carefully constructed partitions of the vertices while maintaining a *superset* of the information revealed to the algorithm in each part: We prove that, no matter the sequence of queries, the information about the edges that are guaranteed to exist and those that are guaranteed not to exist in each part of the partition is less than a certain superset of existing edges and non-existing edges that are easy to describe, up to isomorphism. Our arguments are substantially more involved than those for the vertex-query model [8], which is due to the fact that edge queries can have more complex structure.

Further Related Work. The study of graph algorithms with query access to the input dates back to the works of Feige [11] and Goldreich and Ron [14]. The literature distinguishes between *local queries* – such as vertex-degree queries, neighborhood queries, and edge-existence queries [11, 14, 7] – and *global queries* – such as (bipartite) independent set queries [6, 1], linear, or and cut queries [3], and maximal matching queries as studied in this paper [8]. We refer the reader to [1] and the references therein for an overview.

Outline. We first present our main result, a 3-round algorithm for MBM, in Section 2. Our lower bound results are discussed in Section 3, and we conclude with open questions in Section 4.

2 3-Round Algorithm

In this section, we present our 3-round query algorithm for MBM (see Algorithm 1). We prove the following result, which constitutes the main result of this paper:

▶ **Theorem 1.** *Algorithm 1 is a deterministic 3-round $\frac{5}{8}$-approximation algorithm for MBM in the edge-query model.*

Algorithm 1 Three Rounds using Maximal Matching Queries.

Input: A bipartite graph $G = (A, B, E)$ and a maximal matching oracle QUERY
Output: A large matching M_{out} of G

First round

$M_1 \leftarrow \text{QUERY}(G[A \cup B])$
$G_L = G[A(M_1) \cup \overline{B(M_1)}]$
$G_R = G[\overline{A(M_1)} \cup B(M_1)]$

Second round

$M_2 \leftarrow \text{QUERY}(G_L \cup G_R)$
Let $A' \subseteq A(M_1)$ and $B' \subseteq B(M_1)$ be the endpoints of length-2 paths in $M_1 \cup M_2$
$G' = G[A' \cup B' \cup \overline{A(M_1)} \cup \overline{B(M_1)}]$

Third round

$M_3 \leftarrow \text{QUERY}(G')$

Output

return M_{out}, the largest matching in $M_1 \cup M_2 \cup M_3$

Our algorithm operates on a bipartite input graph $G = (A, B, E)$, where only the vertex sets A and B are initially known to the algorithm. It only uses the edge-query maximal matching oracle to compute maximal matchings in subgraphs of G. Our algorithm initially computes a maximal matching M_1 of G in the first round. Then, in the second round, maximal matchings in the subgraphs G_L and G_R are computed (M_2 denotes the union of both matchings), where G_L consists of the edges connecting the B-vertices unmatched in M_1 to the A-vertices matched in M_1 and G_R is defined similarly with the roles of A and B reversed. The matching M_2 w.r.t. M_1 possibly finds some length-3 augmenting paths, which we denote by P, while the remaining ones make up length-2 alternating paths. Last, in the third round, a maximal matching M_3 is computed in the subgraph induced by the vertices unmatched by M_1 and the endpoints of the length-2 paths in $M_1 \cup M_2$ that are also in $A(M_1) \cup B(M_1)$. Each edge of the matching M_3 thus completes length-3 and length-5 augmenting paths, which we denote by Q. See Figure 1 for an example run of the algorithm where $G_{\text{in}} = G[A(M_1) \cup B(M_1)]$.

▶ **Observation 2.** *The size of Q is exactly the size of M_3.*

The goal of our analysis is to show that the size of the returned matching M_{out}, the largest matching in $M_1 \cup M_2 \cup M_3$, is always at least a $\frac{5}{8}$-approximation of a maximum matching M^*. To that end, we can always find a large maximal matching by appropriately augmenting M_1

with the augmenting paths $P \cup Q$. Although each edge in M_1 can be augmented by at most one augmenting path, the augmenting paths $P \cup Q$ are not necessarily vertex-disjoint since the vertices unmatched by M_1 may be incident to an edge in M_2 and also one in M_3. See Figure 1 for an example of this. However, we observe that the intersection multi-graph of the augmenting paths $P \cup Q$ has maximum degree 2 and, in particular, constitutes a collection of paths and even-length cycles. We can thus pick an independent set of non-overlapping augmenting paths of size $\frac{1}{2}(|P| + |Q|)$ and thus obtain the following:

$$|M_{\text{out}}| \geq |M_1| + \frac{1}{2}(|P| + |Q|). \tag{1}$$

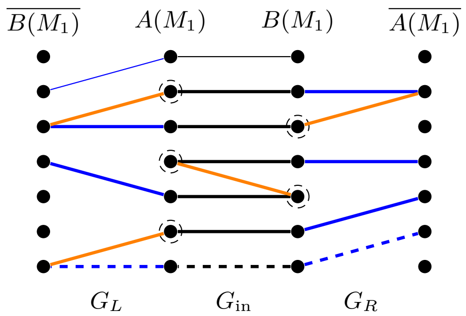


Figure 1 An example run of Algorithm 1 that showcases some possible intersections of the augmenting paths. The edges of M_1 are in black, the edges of M_2 are in blue, and the edges of M_3 are in orange. The dashed thick edges are those which belong to augmenting paths in P, whereas the solid thick edges are those which belong to Q. The vertices of A' and B' are circled.

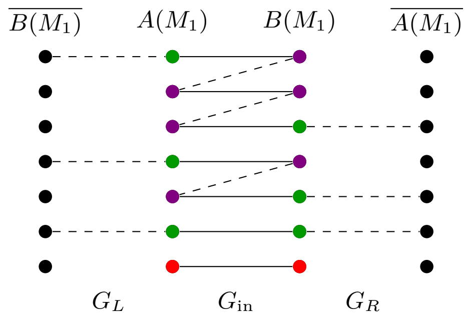


Figure 2 An example of the implied graph structure w.r.t. M^* and M_1. The edges of M_1 are the solid edges and the edges of $M^* \setminus M_1$ are dashed ones. The edges of $M^* \cap M_1$ are the solid edges not incident to any dashed ones. The red vertices represent A_* and B_*, the green ones represent A_{out} and B_{out}, and the violet ones represent A_{in} and B_{in}.

We highlight here that either finding a large matching in the first round or finding many augmenting paths in the second round leaves fewer augmenting paths to be found in the third round. Therefore, the size of Q must be a decreasing function of $|M_1|$ and $|P|$. We subsequently formalise this by systematically bounding the quantities required to bound the size of M_3, which is equivalent to the size of Q (Observation 2). Theorem 1 then immediately follows from Equation (1).

Let M^* be a maximum matching in G with size $\mu(G)$. The first query finds a maximal matching M_1 of G, which is always at least half the size of M^* (Observation 3). By considering a maximum matching M^* such that $M^* \cup M_1$ contains no even-length alternating paths or cycles (Lemma 4), each vertex in $A(M_1)$ and $B(M_1)$ are endpoints of exactly one edge in M^* (Lemma 5) and we have that M_1 relates exactly to the the number of edges of M^* in G_L and G_R, respectively (Lemma 6). We give these proofs in Appendix A for completeness. As such, for the remainder of the analysis, we assume this choice of M^*. See Figure 2 for an example of the implied structure.

▶ **Observation 3.** $|M_1| = (\frac{1}{2} + \epsilon) \cdot \mu(G)$, $0 \leq \epsilon \leq \frac{1}{2}$.

▶ **Lemma 4.** *There exists a maximum matching M^* such that $M^* \cup M_1$ has no even-length alternating paths or cycles.*

▶ **Lemma 5.** *Each vertex of $A(M_1)$ and $B(M_1)$ is the endpoint of an edge in M^*.*

▶ **Lemma 6.** $|M^* \cap G_L| = |M^* \cap G_R| = (\frac{1}{2} - \epsilon) \cdot \mu(G)$.

We now have that the vertices $A(M_1)$ and $B(M_1)$ can be partitioned based on the kind of edge in M^* that they are endpoints of, i.e., by Lemma 5. Let $A_* := A(M^* \cap M_1)$, $A_{\text{out}} := A(M^* \cap G_L)$, and $A_{\text{in}} := A(M_1) \setminus (A_* \cup A_{\text{out}})$. Similarly define $B_* := B(M^* \cap M_1)$, $B_{\text{out}} := B(M^* \cap G_R)$ and $B_{\text{in}} := B(M_1) \setminus (B_* \cup B_{\text{out}})$. See the coloured vertices of Figure 2 for an example of these partitionings.

▶ **Lemma 7.** $|A_{\text{in}}| = |B_{\text{in}}| = 2\epsilon \cdot \mu(G) - |M^* \cap M_1|$.

Proof. By construction of the partitions, we have that $|A_*| = |M^* \cap M_1| = |B_*|$, $|A_{\text{out}}| = |M^* \cap G_L|$ and $|B_{\text{out}}| = |M^* \cap G_R|$. Then, by Lemma 6, it follows that $|A_{\text{out}}| = |B_{\text{out}}| = (\frac{1}{2} - \epsilon) \cdot \mu(G)$. Finally, since $|A(M_1)| = |B(M_1)| = |M_1| = (\frac{1}{2} + \epsilon) \cdot \mu(G)$ by Observation 3, we have that $|A_{\text{in}}| = (\frac{1}{2} + \epsilon) \cdot \mu(G) - |A_{\text{out}}| - |A_*| = 2\epsilon \cdot \mu(G) - |M^* \cap M_1| = |B_{\text{in}}|$. ◀

Intuitively, we have that finding edges of M^* with the first query always puts us in a better situation. Therefore, we consider the effect that these edges have on the quantities that we bound. We introduce some helpful notation in this regard: For any matching M, we define $M(A_*)$ (and $M(B_*)$) as the edges of M that have one endpoint in A_* (resp. B_*), i.e., those which are incident to edges of $M^* \cap M_1$.

The second query finds a maximal matching M_2, which is the union of (vertex-disjoint) maximal matchings M_L in G_L and M_R in G_R. Each edge of M_2 forms the beginning of an alternating path in $M_1 \cup M_2$, some of which may immediately form length-3 augmenting paths P while the rest form length-2 alternating paths P', which are extended in the third round. Note that the endpoints of P' that are in $A(M_1)$ and $B(M_1)$ are the vertex sets A' and B', respectively. We then partition A' and B' such that $A'_{\text{out}} := A' \cap A_{\text{out}}$ and similarly define A'_{in}, A'_*, B'_{out}, B'_{in}, and B'_*.

▶ **Lemma 8.** $|M^* \cap M_1| \geq \frac{1}{2} \cdot (|M_L(A_*)| + |M_R(B_*)|)$.

Proof. By definition, every edge of the matchings $M_L(A_*)$ and $M_R(B_*)$ is incident to an edge of $M^* \cap M_1$. Hence, $|M^* \cap M_1| \geq |M_L(A_*)|$ and $|M^* \cap M_1| \geq |M_R(B_*)|$, which implies the result. ◀

▶ **Lemma 9.** $|M_2| \geq (\frac{1}{2} - \epsilon) \cdot \mu(G) + \frac{1}{2}(|M_L(A_*)| + |M_R(B_*)|)$

Proof. Consider the edges of $M^* \cap G_L$. Every edge of M_L is incident to at most two edges of $M^* \cap G_L$, and every edge of $M_L(A_*)$ is incident to at most one of the edges of $M^* \cap G_L$. By a counting argument, we have that $|M_L| \geq \frac{1}{2}(|M^* \cap G_L| - |M_L(A_*)|) + |M_L(A_*)|$. Then, by Lemma 6, it follows that $|M_L| \geq \frac{1}{2}\left((\frac{1}{2} - \epsilon) \cdot \mu(G) + |M_L(A_*)|\right)$. We similarly bound $|M_R|$ w.r.t. $M_R(B_*)$ and obtain the results since $|M_2| = |M_L| + |M_R|$. ◀

▶ **Lemma 10.** $|P'| = |A'_{\text{in}}| + |B'_{\text{in}}| + |A'_{\text{out}}| + |B'_{\text{out}}| + |A'_*| + |B'_*| = |M_2| - 2|P|$.

Proof. Each length-2 alternating path in P' has an endpoint in either $A(M_1)$ or $B(M_1)$, but not both; thus, we have that $|P'| = |A'| + |B'|$ and the first equality follows by definition of the partitions of A' and B'. For the subsequent equality, we have that every edge of M_L contributes to a length-2 alternating path except for the ones which contribute to length-3 augmenting paths. This gives $|M_L| - |P|$ of them which have an edge in M_L. A similar reasoning w.r.t. M_R shows that $|M_R| - |P|$ of them have an edge in M_R. The result then follows since the paths in P' either have an edge in M_L or in M_R, but never both. ◀

▶ **Lemma 11.** $|A'_*| \leq |M_R(B_*)|$ *and* $|B'_*| \leq |M_L(A_*)|$.

Proof. Consider any vertex in $a \in A'_*$. By definition, a is the endpoint of an edge $(a, b) \in M^* \cap M_1$, which implies that $b \in B_*$. Since a is the endpoint of a path $p \in P'$, we have that $p = (a, b, a_R)$ where (a_R, b) must be an edge of $M_R(B_*)$. Finally, every $a \in A'_*$ has a unique $(a_R, b) \in M_R(B_*)$ since each path in P' is vertex-disjoint and thus the first inequality follows. The second inequality follows similarly w.r.t. B_* and $M_L(A_*)$ instead. ◀

The third query finds a maximal matching M_3 in the graph $G' = G[A' \cup B' \cup \overline{A(M_1)} \cup \overline{B(M_1)}]$, which implies that the size of M_3 is at least half of $\mu(G')$. As such, it is sufficient to bound $\mu(G')$. To that end, we bound the number of edges of M^* in G', which we accomplish by decomposing G' into edge-disjoint subgraphs $G'_L = G[A' \cup \overline{B(M_1)}]$, $G'_{\text{in}} = G[A' \cup B']$ and $G'_R = G[B' \cup \overline{A(M_1)}]$. Using the quantities we have previously bounded, we then obtain our final bound on the size of M_3 as a decreasing function of $|M_1|$, in terms of ϵ, and $|P|$.

▶ **Lemma 12.** $|M^* \cap G'_L| = |A'_{\text{out}}|$, $|M^* \cap G'_R| = |B'_{\text{out}}|$, *and* $|M^* \cap G'_{\text{in}}| \geq |A'_{\text{in}}| + |B'_{\text{in}}| - |A_{\text{in}}|$.

Proof. By definition, every vertex of A'_{out} is incident to an edge in $M^* \cap G'_L$, and vice versa; hence, it holds that $|M^* \cap G'_L| = |A'_{\text{out}}|$. A similar argument shows that $|M^* \cap G'_R| = |B'_{\text{out}}|$ holds. To bound $|M^* \cap G'_{\text{in}}|$, consider an edge $(a, b) \in (M^* \backslash M_1) \cap G_{\text{in}}$. By definition, $a \in A_{\text{in}}$ and $b \in B_{\text{in}}$; however, $(a, b) \in M^* \cap G'_{\text{in}}$ if and only if $a \in A'_{\text{in}}$ and $b \in B'_{\text{in}}$. Thus, there are at most $(|A_{\text{in}}| - |A'_{\text{in}}|) + (|B_{\text{in}}| - |B'_{\text{in}}|)$ edges of $(M^* \backslash M_1) \cap G_{\text{in}}$ which are not in $M^* \cap G'_{\text{in}}$. By definition, it holds that $|(M^* \backslash M_1) \cap G_{\text{in}}| = |B_{\text{in}}|$ and it follows that $|M^* \cap G'_{\text{in}}| \geq |A'_{\text{in}}| + |B'_{\text{in}}| - |A_{\text{in}}|$. ◀

▶ **Lemma 13.** $|M_3| \geq (\frac{1}{4} - \frac{3\epsilon}{2}) \cdot \mu(G) - |P|$.

Proof. By rearranging the equation in Lemma 10 and applying Lemma 11, we have that

$$|A'_{\text{in}}| + |B'_{\text{in}}| + |A'_{\text{out}}| + |B'_{\text{out}}| \geq |M_2| - 2|P| - |M_R(B_*)| - |M_L(A_*)|. \tag{2}$$

We subsequently bound $\mu(G')$, which the result follows from since $|M_3| \geq \frac{1}{2} \cdot \mu(G')$.

$$\begin{aligned}
\mu(G') &\geq |M^* \cap G'_L| + |M^* \cap G'_R| + |M^* \cap G'_{\text{in}}| && \text{(edge-disjoint subgraphs)} \\
&\geq |A'_{\text{out}}| + |B'_{\text{out}}| + |A'_{\text{in}}| + |B'_{\text{in}}| - |A_{\text{in}}| && \text{(by Lemma 12)} \\
&\geq |M_2| - 2|P| - |M_L(A_*)| - |M_R(B_*)| - |A_{\text{in}}| && \text{(by Equation (2))} \\
&\geq (\frac{1}{2} - \epsilon) \cdot \mu(G) - \frac{1}{2}(|M_L(A_*)| + |M_R(B_*)|) - 2|P| - |A_{\text{in}}| && \text{(by Lemma 9)} \\
&\geq (\frac{1}{2} - \epsilon) \cdot \mu(G) - |M^* \cap M_1| - 2|P| - |A_{\text{in}}| && \text{(by Lemma 8)} \\
&= (\frac{1}{2} - \epsilon) \cdot \mu(G) - |M^* \cap M_1| - 2|P| - (2\epsilon \cdot \mu(G) - |M^* \cap M_1|) && \text{(by Lemma 7)} \\
&= (\frac{1}{2} - 3\epsilon) \cdot \mu(G) - 2|P|.
\end{aligned}$$ ◀

We are now ready to bound the size of the returned matching M_{out} w.r.t. $\mu(G)$, the size of the maximum matching M^*, thus proving Theorem 1.

▶ **Lemma 14.** *The large matching M_{out} returned by Algorithm 1 is always at least a $\frac{5}{8}$-approximation of the size of a maximum matching in the input graph G.*

Proof.

$$\begin{aligned}
|M_{\text{out}}| &\geq |M_1| + \frac{1}{2}(|P| + |Q|) && \text{(by Equation (1))} \\
&= (\frac{1}{2} + \epsilon) \cdot \mu(G) + \frac{1}{2}(|P| + |M_3|) && \text{(by Observations 2 and 3)} \\
&\geq (\frac{1}{2} + \epsilon) \cdot \mu(G) + \frac{1}{2}(|P| + (\frac{1}{4} - \frac{3\epsilon}{2}) \cdot \mu(G) - |P|) && \text{(by Lemma 13)} \\
&= (\frac{5}{8} + \frac{\epsilon}{4}) \cdot \mu(G). && \blacktriangleleft
\end{aligned}$$

3 Lower Bound Results

In this section, we give our lower bounds for edge-query algorithms for MM for up to 3 rounds, showing that our 3-round algorithm is best possible:

▶ **Theorem 15.** *There does **not** exist a deterministic algorithm on a n-vertex input graph for MM in the maximal matching edge-query model that achieves a better than*

1. $\frac{1}{2}$*-approximation in* 1 *round,*
2. $\left(\frac{1}{2} + \frac{2}{n}\right)$*-approximation in* 2 *rounds, and*
3. $\left(\frac{5}{8} + \frac{24}{n}\right)$*-approximation in* 3 *rounds.*

We prove our lower bounds by considering a game between a player, i.e., the algorithm, and an oracle in the edge-query model. The goal of the player is to learn a large matching in the underlying bipartite graph $G = (A, B, E)$ that is adversarially constructed by the oracle along the way. The player initially only knows the vertices A and B and is allowed to query the oracle with any set of edges $Q \subseteq A \times B$ in each round, typically basing the query on any information about G revealed in previous rounds. The oracle then returns an adversarially chosen maximal matching in the subgraph $G[E \cap Q]$, revealing as little information about a large matching as possible. Throughout the game, once information about G is revealed, it may not be altered in subsequent rounds.

Let Q_i be the player's query and let M_i be the maximal matching returned by the oracle in round i. The player learns that the edges M_i are present in G and that the edges in Q_i with both endpoints unmatched by M_i do not exist in G. The player thus learns about both *edges* and *non-edges*. As such, we use *structure graphs* to encapsulate the information known by the player up to graph isomorphisms, providing a simple representation in which to prove our lower bounds – similar to the work by binti Khalil and Konrad [8].

▶ **Definition 16** (Structure Graph [8])**.** *A* 4*-tuple* (A, B, E, F) *is a* bipartite structure graph *if* E *and* F *are disjoint sets of edges such that* (A, B, E) *and* (A, B, F) *are bipartite graphs. The set* E *corresponds to the set of edges learnt by the algorithm, and the set* F *corresponds to the set of non-edges learnt.*

A player always begins with the empty structure graph $H_0 = (A, B, E_0, F_0)$ where $E_0 = F_0 = \emptyset$. In a game of r rounds, the structure graphs $H_1, H_2, \ldots, H_r$ represent the information (edges and non-edges) learned by the player after each round, which are based purely on the player's queries $Q_1, Q_2, \ldots, Q_r$ and the oracle's adversarially returned matchings $M_1, M_2, \ldots, M_r$. Consider a player's structure graph H_i w.r.t. H_{i-1} for any $i \in [r]$. It consists of the edges $E_i = E_{i-1} \cup M_i$ and the non-edges $F_i = F_{i-1} \cup N_i$ where $N_i = Q_i \cap (\overline{A(M_i)} \times \overline{B(M_i)})$ ensures that M_i is maximal. The player's information at the end of a round is then necessarily a superset of the information known at the end of the previous round, i.e., $E_i \supseteq E_{i-1}$ and $F_i \supseteq F_{i-1}$. Hence, the structure graph H_i *dominates*

H_{i-1}, which we say in general for any structure graph that is a superset of the information of another up to graph isomorphism. The underlying graph G may then be any graph that is *consistent* with *all* the information H_r revealed to the player by the end of round r.

▶ **Definition 17** (Consistent). *Let $G = (A, B, E)$ be any bipartite graph and let $H_i = (A, B, E_i, F_i)$ be a bipartite structure graph. G is* consistent *with H_i iff $E \supseteq E_i$ and $E \cap F_i = \emptyset$.*

The largest matching a player who knows H_r may output is the maximum matching M_r^{out} in (A, B, E_r). Therefore, the oracle adversarially constructs the graph G so that G is consistent with H_r and so that G has the largest possible maximum matching. This implies that the approximation factor of H_r is $\frac{|M_r^{\text{out}}|}{\mu(G)}$. Note that H_r is strongly dependent on the player's sequence of queries $Q_1, Q_2, ..., Q_r$. Altering even a single query could alter H_r, the player's largest matching M_r^{out} and, most importantly, the approximation factor of H_r. Hence, the goal for proving our lower bound results is to find an upper bound on the approximation factor achieved by any sequence of queries $Q_1, \ldots, Q_r$ for $r = 1, 2$ and 3.

Before proceeding with our analysis, we first present the ideas that we employ to prove our lower bounds. To generally consider all possible queries in each round $i \in [r]$, we allow the oracle to commit to *more* information than is revealed to the player, denoted by the structure graph $\tilde{H}_i$. In particular, we show that $\tilde{H}_i$ dominates the player's structure graph H_i learned regardless of the query Q_i. Then, at the end of round i, the player is assumed to have knowledge of the oracle's structure graph $\tilde{H}_i$. This implies that $\tilde{H}_r$ dominates the structure graph learned by the player for any sequence of queries $Q_1, \ldots, Q_r$. We also allow the oracle to partition the vertices of the graph and consider the vertex-induced subgraph of each part independently. By making the partition a function of the query, we create desirable properties in each part. This, however, does not consider the edges that cross the partition, which are thus asserted as non-edges. Formally, we recombine the structure graphs learned in each part using the *disjoint union* (Definition 18) at the end of round r. Then, the approximation factor of the recombined structure graph follows naturally from the independent parts by Observations 19 and 20.

▶ **Definition 18** (Disjoint Union). *Let (A_x, B_x) and (A_y, B_y) represent an arbitrary partitioning of the vertices A and B into two parts and let $H_x = (A_x, B_x, E_x, F_x)$ and $H_y = (A_y, B_y, E_y, F_y)$ be any bipartite structure graphs. Then, their disjoint union is $H_x \dot\cup H_y = (A, B, E_x \cup E_y, F_x \cup F_y \cup (A_x \times B_y) \cup (A_y \times B_x))$.*

▶ **Observation 19.** *Let H_x and H_y be bipartite structure graphs on disjoint sets of vertices with largest output matchings M_x^{out} and M_y^{out}, respectively. Then, the largest output matching of $H_x \dot\cup H_y$ is of size $|M_x^{out}| + |M_y^{out}|$.*

▶ **Observation 20.** *Let H_x and H_y be bipartite structure graphs on disjoint sets of vertices with consistent graphs G_x and G_y, respectively. Then, there exists a graph G consistent with $H_x \dot\cup H_y$ such that $\mu(G) = \mu(G_x) + \mu(G_y)$.*

We begin our analysis with the following simplifying assumption, where its justification is given in Appendix B for completeness:

▶ **Assumption 21.** *In each round $1 \leq i \leq r$, we assume that the query Q_i does not contain any edges or non-edges already learned by the player.*

Let $A = A_{\text{in}} \cup A_{\text{out}}$ and $B = B_{\text{in}} \cup B_{\text{out}}$ be such that A_{in}, A_{out}, B_{in} and B_{out} are disjoint sets of vertices of size $\frac{n}{4}$ where n is the number of vertices and a multiple of 4. We further assert that $\frac{n}{4}$ is odd. Then, the player begins with only the knowledge of A and B, i.e., the empty structure graph H_0.

First Round. Let $\tilde{M}_1$ be a matching of size $\frac{n}{4}$ that matches A_{in} to B_{in}. We assert its maximality by letting $\tilde{N}_1^{\text{max}} = A_{\text{out}} \times B_{\text{out}}$ be non-edges. Additionally, the non-edges $\tilde{N}_1^{\text{ind}} = (A_{\text{in}} \times B_{\text{in}}) \setminus \tilde{M}_1$ assert that it is an induced matching[2]. Then, we define $\tilde{H}_1 = (A, B, \tilde{E}_1, \tilde{F}_1)$ where $\tilde{E}_1 = \tilde{M}_1$ and $\tilde{F}_1 = \tilde{N}_1 = \tilde{N}_1^{\text{max}} \cup \tilde{N}_1^{\text{ind}}$. See Figure 3 for an illustration.

▶ Remark. $\tilde{H}_1$ can be defined on any subset of vertices $A' \subseteq A$ and $B' \subseteq B$ such that $|A'_{\text{in}}| = |B'_{\text{in}}| = |A'_{\text{out}}| = |B'_{\text{out}}|$. We later use such generalisations, denoted as $\tilde{H}_1(A', B')$.

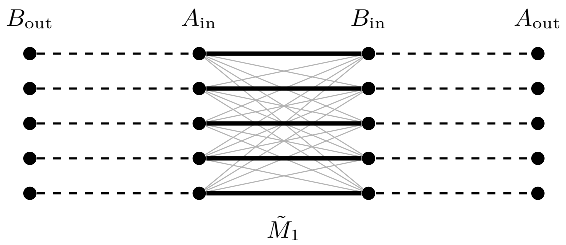


Figure 3 An illustration of structure graph $\tilde{H}_1$. The thick solid black edges represent the matching $\tilde{M}_1$. The non-edges $\tilde{N}_1^{\text{max}}$ are implicit by the layout of the vertices and the non-edges $\tilde{N}_1^{\text{ind}}$ are the grey edges. The dashed black edges are a perfect matching in a worst-case underlying graph.

▶ **Lemma 22.** *Any structure graph H_1 learned by the player is dominated by $\tilde{H}_1$.*

Proof. Let Q_1 be any arbitrary query and let $M^*(Q_1)$ be a maximum matching in the query graph (A, B, Q_1). If $|M^*(Q_1)| \geq |\tilde{M}_1|$, then let $M_1 \subseteq M^*(Q_1)$ be a subset of size $|\tilde{M}_1|$. We assert the maximality of matching M_1 with the non-edges $N_1 = \overline{A(M_1)} \times \overline{B(M_1)}$. Otherwise, $|M^*(Q_1)| < |\tilde{M}_1|$ and we let $M_1 = M^*(Q_1)$, which is trivially a maximal matching among the edges of the query Q_1; hence, $N_1 = \emptyset$.

Finally, in either case, let σ be a graph isomorphism such that $M_1 \subseteq \sigma(\tilde{M}_1)$, which implies that $N_1 \subseteq \sigma(\tilde{N}_1)$. Therefore, the player's H_1 learned is always dominated by $\tilde{H}_1$. ◀

▶ **Lemma 23.** *The approximation factor of the structure graph $\tilde{H}_1$ is $\frac{1}{2}$.*

Proof. The largest matching $\tilde{M}_1^{\text{out}}$ that the player who knows $\tilde{H}_1$ may output is the matching $\tilde{M}_1$, which matches half of the vertices. Since no information regarding the edges in $A_{\text{in}} \times B_{\text{out}}$ or $A_{\text{out}} \times B_{\text{in}}$ has been revealed, we can choose a graph G consistent with $\tilde{H}_1$ that has a perfect matching, which is of size $2 \cdot |\tilde{M}_1|$, thus proving the result. In particular, it has a matching that arbitrarily matches A_{in} to B_{out} and A_{out} to B_{in} (see Figure 3). ◀

Lemma 22 shows that $\tilde{H}_1$ dominates all possible structure graphs learned by the player by the end of round 1, i.e., after query Q_1. Thus, Lemma 23 immediately implies Theorem 15-1.

Second Round. Let $\tilde{M}_L \subseteq (A_{\text{in}} \times B_{\text{out}})$ and $\tilde{M}_R \subseteq (A_{\text{out}} \times B_{\text{in}})$ be matchings of size $\left\lfloor \frac{|\tilde{M}_1|}{2} \right\rfloor$ such that $\tilde{M}_1 \cup \tilde{M}_L \cup \tilde{M}_R$ has no length-3 paths. Let $\tilde{N}_L^{\text{max}} = A_{\text{in}} \setminus A(\tilde{M}_L) \times B_{\text{out}} \setminus B(\tilde{M}_L)$ and $\tilde{N}_R^{\text{max}} = A_{\text{out}} \setminus A(\tilde{M}_R) \times B_{\text{in}} \setminus B(\tilde{M}_R)$ be the non-edges that assert the maximality of the matching $\tilde{M}_2 = \tilde{M}_L \cup \tilde{M}_R$ among the unknown edges. Let $\tilde{N}_L^{\text{ind}} = \left(A(\tilde{M}_L) \times B(\tilde{M}_L)\right) \setminus \tilde{M}_L$ and $\tilde{N}_R^{\text{ind}} = \left(A(\tilde{M}_R) \times B(\tilde{M}_R)\right) \setminus \tilde{M}_R$ be the non-edges required to make the matching

[2] Committing to an induced matching simplifies the arguments in the subsequent rounds without affecting the approximation factor of the player's structure graphs.

induced. Additionally, if $|\tilde{M}_1|$ is odd, then let $e^*_{\text{in}} \in \tilde{M}_1$ be the only edge with both endpoints unmatched by $\tilde{M}_L$ and $\tilde{M}_R$, and similarly let $a_{\text{out}} \in A_{\text{out}}$ and $b_{\text{out}} \in B_{\text{out}}$ be any vertices unmatched by $\tilde{M}_L$ and $\tilde{M}_R$. We assert that e^*_{in} is an isolated edge and that a_{out} and b_{out} are isolated vertices[3], which implies the non-edges $\tilde{N}_* = (A(e^*_{\text{in}}) \times B \backslash B(e^*_{\text{in}})) \cup (A \backslash A(e^*_{\text{in}}) \times B(e^*_{\text{in}})) \cup (A \times \{b_{\text{out}}\}) \cup (\{a_{\text{out}}\} \times B)$. Otherwise, if $|\tilde{M}_1|$ is even, we let $\tilde{N}_* = \emptyset$. Then, we define $\tilde{H}_2 = (A, B, \tilde{E}_1 \cup \tilde{M}_2, \tilde{F}_1 \cup \tilde{N}_2)$ where $\tilde{N}_2 = \tilde{N}^{\max}_L \cup \tilde{N}^{\max}_R \cup \tilde{N}^{\text{ind}}_L \cup \tilde{N}^{\text{ind}}_R \cup \tilde{N}_*$. See Figure 4 for an illustration where the non-edges $\tilde{N}_*$ have been removed for clarity.

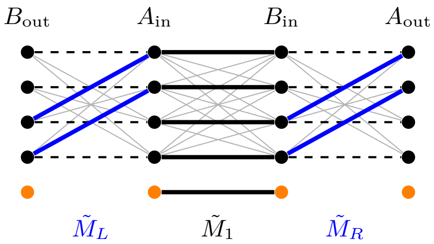


Figure 4 An illustration of the structure graph $\tilde{H}_2$. The thick blue edges represent the matching $\tilde{M}_2$ and the grey ones are the non-edges $\tilde{N}_2 \backslash \tilde{N}_*$. The orange vertices and their incident edge are isolated and only present if $|\tilde{M}_1|$ is odd. The black dashed edges represent a large maximum matching in a worst-case underlying graph.

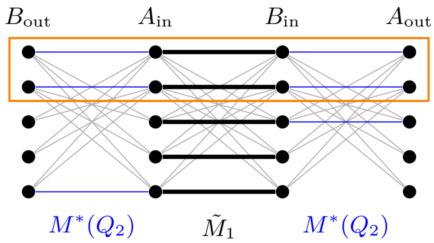


Figure 5 An example of the partitioning based on $M^*(Q_2)$. The blue edges represent the matching $M^*(Q_2)$. The vertices in the orange box represent part (A^+, B^+) and the remaining unboxed ones represent part (A^-, B^-). The grey edges show the additional non-edges asserted by the disjoint union.

Let Q_2 be an arbitrary query of round 2 and let $M^*(Q_2)$ be a maximum matching of (A, B, Q_2). By Assumption 21, we have that Q_2 only has edges in $A_{\text{in}} \times B_{\text{out}}$ and $A_{\text{out}} \times B_{\text{in}}$, which implies that the edges of $M^*(Q_2)$ either form vertex-disjoint length-3 or length-2 alternating paths w.r.t. $\tilde{M}_1$. As such, we partition the vertices of A and B to consider the length-3 and length-2 paths separately: Part (A^+, B^+) with N^+ many vertices consists of all vertices that lie on a length-3 path, whereas part (A^-, B^-) consists of the remaining $N^- = n - N^+$ many vertices, whose induced subgraph includes all the length-2 paths. See Figure 5 for an example of the partitioning.

▶ **Lemma 24.** *$\tilde{H}_1$ is partitioned w.r.t. parts (A^+, B^+) and (A^-, B^-) into structure graphs $\tilde{H}^+_1 = \tilde{H}_1(A^+, B^+)$ and $\tilde{H}^-_1 = \tilde{H}_1(A^-, B^-)$, respectively, where $\tilde{H}^+_1 \dot\cup \tilde{H}^-_1$ dominates $\tilde{H}_1$.*

With Lemma 24, whose proof is given in Appendix C for completeness, the player's information in each part at the start of round 2 is exactly the respective generalisations of $\tilde{H}_1$. As such, we show that the respective generalisations of $\tilde{H}_2$ dominate the player's structure graphs H^+_2 and H^-_2 learned in each part after query Q_2.

▶ **Lemma 25.** *Any structure graph H^+_2 learned by the player is dominated by $\tilde{H}^+_2 = \tilde{H}_2(A^+, B^+)$.*

▶ **Lemma 26.** *Any structure graph H^-_2 learned by the player is dominated by $\tilde{H}^-_2 = \tilde{H}_2(A^-, B^-)$.*

[3] Committing to the isolated edge and vertices simplifies the arguments in the subsequent round without affecting the approximation factor of the player's structure graphs.

Lemmas 25 and 26 are proven similarly to Lemma 22; hence, we give their proofs in Appendix C to save space. By Lemmas 25 and 26, the player's overall information at the end of round 2 is dominated by the structure graph $\tilde{H}_2^+ \dot{\cup} \tilde{H}_2^-$.

▶ **Lemma 27.** *The approximation factor of the structure graph $\tilde{H}_2^+ \dot{\cup} \tilde{H}_2^-$ is at most $\frac{1}{2} + \frac{2}{n}$.*

Proof. We claim that the largest matching is of size $|\tilde{M}_1|$ and that there exists a consistent graph G such that $\mu(G) = 2 \cdot |\tilde{M}_1| - 1$. Then, the approximation factor of $\tilde{H}_2^+ \dot{\cup} \tilde{H}_2^-$ is at most $\frac{|\tilde{M}_1|}{2\cdot|\tilde{M}_1|-1} = \frac{1}{2} + \frac{1}{4\cdot|\tilde{M}_1|-2} \leq \frac{1}{2} + \frac{1}{2\cdot|\tilde{M}_1|} = \frac{1}{2} + \frac{2}{n}$ for large enough n.

We now prove the first claim. Since there are no augmenting paths in $\tilde{M}_1^+ \cup \tilde{M}_2^+$ or $\tilde{M}_1^- \cup \tilde{M}_2^-$, the largest output matching is $\tilde{M}_2^{\text{out}} = \tilde{M}_1^+ \cup \tilde{M}_1^- = \tilde{M}_1$ by Observation 19. It remains to prove the second claim. Since $|\tilde{M}_1| = \frac{n}{4}$ is odd, w.l.o.g., $|\tilde{M}_1^+|$ is odd and $|\tilde{M}_1^-|$ is even; hence, we have that $|\tilde{M}_2^+| = |\tilde{M}_1^+| - 1$ and $|\tilde{M}_2^-| = |\tilde{M}_1^-|$. Since both matchings in both parts are maximal and induced, the only unknown edges are those in $(A_{\text{in}} \times B_{\text{out}}) \cup (A_{\text{out}} \times B_{\text{in}})$ that have only one endpoint matched by $\tilde{M}_2^+$ or $\tilde{M}_2^-$ and are not incident to the isolated edge or vertices. Thus, we may use these to construct consistent graphs in $\tilde{H}_2^+$ and $\tilde{H}_2^-$ with maximum matchings of size $2 \cdot |\tilde{M}_2^+| + 1 = 2 \cdot |\tilde{M}_1^+| - 1$, which includes the isolated edge, and $2 \cdot |\tilde{M}_2^-| = 2 \cdot |\tilde{M}_1^-|$, respectively. By Observation 20, this gives the graph G as required. ◀

Overall, we have that any arbitrary query Q_2 can be used to construct the relevant partition of the vertices where Lemmas 25–27 always hold, thus proving Theorem 15-2.

Third Round. We continue to consider the partition w.r.t. the query Q_2 where the player now knows generalisations of $\tilde{H}_2$ in each part. As such, it is sufficient to find a structure graph $\tilde{H}_3$ that dominates $\tilde{H}_2$ and then apply the disjoint union as before. Note that we consider only the even case of $\tilde{H}_2$ since, by Assumption 21, the endpoints of the isolated edge e^*_{in} and the isolated vertices a_{out} and b_{out} in the odd case of $\tilde{H}_2$ (see Figure 4) are not endpoints of any edge in a third round query, thus reducing it to an even case of $\tilde{H}_2$.

With knowledge of an even case of $\tilde{H}_2$ at the start of round 3, the player is aware of two edge-disjoint maximal and induced matchings $\tilde{M}_1$ and $\tilde{M}_2$, both of which are half the size of a perfect matching, such that $\tilde{M}_1 \cup \tilde{M}_2$ is exactly the edges of $|\tilde{M}_1|$ many vertex-disjoint length-2 paths, denoted by P. Therefore, by Assumption 21, any third round query may contain only the edges that either (a) *extend* a path in P, denoted by K^{ext}, or (b) provide a *replacement* edge for a path in P, denoted by K^{rep}. Observe that the *extending* or *replacement* edges are either incident to A_{in} (on the left) or incident to B_{in} (on the right). As such, we define the set of possible query edges as a union of (left and right) complete graphs, respectively:

$$K^{\text{ext}} := K_L^{\text{ext}} \cup K_R^{\text{ext}} = (A_{\text{in}}(P) \times B_{\text{out}}(P)) \cup (A_{\text{out}}(P) \times B_{\text{in}}(P)),$$
$$K^{\text{rep}} := K_L^{\text{rep}} \cup K_R^{\text{rep}} = \left(\overline{A_{\text{in}}(P)} \times \overline{B_{\text{out}}(P)}\right) \cup \left(\overline{A_{\text{out}}(P)} \times \overline{B_{\text{in}}(P)}\right)$$

where, for any set of vertices U, $U(P)$ denotes the U-endpoints of paths in P and $\overline{U(P)} := U \backslash U(P)$. We illustrate this in Figure 6.

Let Q_3 be an arbitrary query of round 3. We partition the vertices A and B to consider the paths in P that form vertex-disjoint 6-cycles with edges in $Q_3^{\text{ext}} = Q_3 \cap K^{\text{ext}}$ separately from the ones that do not: Part (A°, B°) with N° many vertices consists of all vertices that lie on the vertex-disjoint 6-cycles, including, for each 6-cycle, a vertex from $\overline{A_{\text{out}}(P)}$ and one from $\overline{B_{\text{out}}(P)}$, whereas part $(A^{\overline{\circ}}, B^{\overline{\circ}})$ consists of the remaining $N^{\overline{\circ}} = n - N^\circ$ many vertices, whose induced subgraph includes all the paths in P that do not form any 6-cycles with each other using the query edges Q_3^{ext}. See Figure 7 for an example of the partitioning.

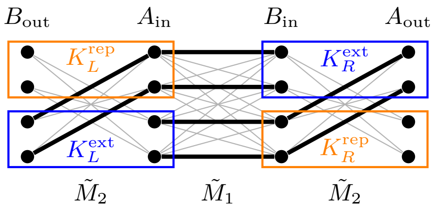


Figure 6 An illustration of the possible query edges by a player who knows $\tilde{H}_2$. The black edges represent the induced maximal matchings $\tilde{M}_1$ and $\tilde{M}_2$ and the grey ones are their corresponding non-edges $\tilde{N}_1$ and $\tilde{N}_2$. The possible query edges K_L^{ext}, K_R^{ext}, K_L^{rep} and K_R^{rep} are complete graphs on the vertices of their respective boxes.

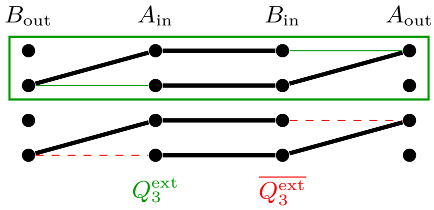
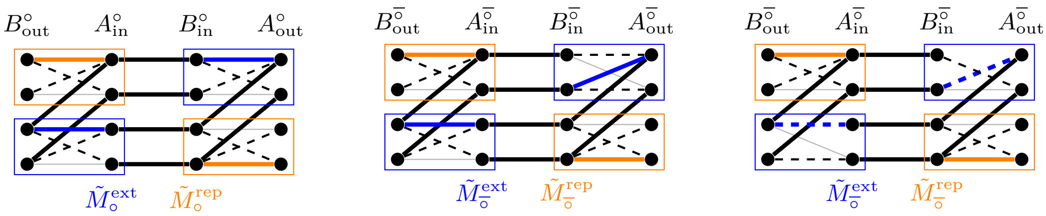


Figure 7 An example of the partitioning based on Q_3^{ext}. The green edges represent the edges necessarily in Q_3^{ext} and the dashed red edges represent the edges necessarily not in Q_3^{ext}, i.e., the edges in $\overline{Q_3^{\text{ext}}} = K^{\text{ext}} \backslash Q_3^{\text{ext}}$. The vertices in the green box represent part (A°, B°) and the remaining unboxed ones represent part $(A^{\overline{\circ}}, B^{\overline{\circ}})$.

▶ **Lemma 28.** *$\tilde{H}_2$ is partitioned w.r.t. parts (A°, B°) and $(A^{\overline{\circ}}, B^{\overline{\circ}})$ into structure graphs $\tilde{H}_2^\circ = \tilde{H}_2(A^\circ, B^\circ)$ and $\tilde{H}_2^{\overline{\circ}} = \tilde{H}_2(A^{\overline{\circ}}, B^{\overline{\circ}})$, respectively, where $\tilde{H}_2^\circ \dot\cup \tilde{H}_2^{\overline{\circ}}$ dominates $\tilde{H}_2$.*

With Lemma 28, whose proof is given in Appendix C for completeness, the player's information in each part at the start of round 3 is exactly the respective generalisations of $\tilde{H}_2$. We show that the structure graphs H_3° and $H_3^{\overline{\circ}}$ learned by the player in each part are dominated by distinct structure graphs $\tilde{H}_3^\circ$ and $\tilde{H}_3^{\overline{\circ}}$, respectively, after query Q_3. We defer the corresponding proofs of Lemmas 29 and 30 to Appendix C to save space.



(a) A gadget of the structure graph $\tilde{H}_3^\circ$.

(b) First case of a gadget of the structure graph $\tilde{H}_3^{\overline{\circ}}$.

(c) Second case of a gadget of the structure graph $\tilde{H}_3^{\overline{\circ}}$.

Figure 8 Illustrations of the gadgets of the structure graphs $\tilde{H}_3^\circ$ and $\tilde{H}_3^{\overline{\circ}}$, respectively. The thick blue edges (solid and dashed) represent the matchings $\tilde{M}_\circ^{\text{ext}}$ and $\tilde{M}_{\overline{\circ}}^{\text{ext}}$. The thick orange edges represent the matchings $\tilde{M}_\circ^{\text{rep}}$ and $\tilde{M}_{\overline{\circ}}^{\text{rep}}$. The grey edges represent the non-edges $\tilde{N}_\circ^{\max}$ and $\tilde{N}_{\overline{\circ}}^{\max}$. The dashed edges (black and blue) represent maximum matchings in worst-case underlying graphs.

Let P° be the set of the $|\tilde{M}_1^\circ|$ many vertex-disjoint length-2 paths in part (A°, B°). Let $\mathcal{G}^\circ$ be a collection of at least $\frac{|\tilde{M}_1^\circ|}{4} - 1$ many vertex-disjoint gadgets such that each gadget is made up of two distinct vertices from $\overline{A_{\text{out}}^\circ(P^\circ)}$ and two from $\overline{B_{\text{out}}^\circ(P^\circ)}$, and four distinct paths from P° where two have their endpoints in A° while the other two have theirs in B°. Let $\tilde{M}_\circ^{\text{ext}} \subseteq K_\circ^{\text{ext}}$ be a matching such that each gadget has two edges that from a 6-cycle with two of the paths. Let $\tilde{M}_\circ^{\text{rep}} \subseteq K_\circ^{\text{rep}}$ be another matching such that each gadget has two edges, one from K_L^{rep} and one from K_R^{rep}, where only one of them is incident to the 6-cycle. Let the inter-gadget edges of $K_\circ^{\text{ext}} \cup K_\circ^{\text{rep}}$ be the non-edges $\tilde{N}_\circ^{\text{gad}}$. We assert the maximality of the matching $\tilde{M}_3^\circ = \tilde{M}_\circ^{\text{ext}} \cup \tilde{M}_\circ^{\text{rep}}$ by committing the edges of $K_\circ^{\text{ext}} \cup K_\circ^{\text{rep}}$ within each gadget that have both endpoints unmatched by $\tilde{M}_3^\circ$ to be the non-edges $\tilde{N}_\circ^{\max}$. Then, we define $\tilde{H}_3^\circ = (A^\circ, B^\circ, \tilde{E}_2^\circ \cup \tilde{M}_3^\circ, \tilde{F}_2^\circ \cup \tilde{N}_3^\circ)$ where $\tilde{N}_3^\circ = \tilde{N}_\circ^{\text{gad}} \cup \tilde{N}_\circ^{\max}$. See Figure 8a for an illustration of a single gadget.

▶ **Lemma 29.** *Any structure graph H_3° learned by the player is dominated by $\tilde{H}_3^{\circ}$.*

Let $P^{\overline{\circ}}$ be the set of the $|\tilde{M}_1^{\overline{\circ}}|$ many vertex-disjoint length-2 paths in part $(A^{\overline{\circ}}, B^{\overline{\circ}})$. Let $\mathcal{G}^{\overline{\circ}}$ be a collection of at least $\frac{|\tilde{M}_1^{\overline{\circ}}|}{4} - \frac{1}{2}$ many vertex-disjoint gadgets such that each gadget is made up of two distinct vertices from $\overline{A^{\overline{\circ}}_{\text{out}}(P^{\overline{\circ}})}$ and two from $\overline{B^{\overline{\circ}}_{\text{out}}(P^{\overline{\circ}})}$, and four distinct paths from $P^{\overline{\circ}}$ where two have their endpoints in $A^{\overline{\circ}}$ while the other two have theirs in $B^{\overline{\circ}}$. Let $\tilde{M}_{\overline{\circ}}^{\text{ext}} \subseteq K_{\overline{\circ}}^{\text{ext}}$ be a matching such that each gadget has two edges that form a length-8 path with three of the paths, leaving the other path unmatched. Let $\tilde{M}_{\overline{\circ}}^{\text{rep}} \subseteq K_{\overline{\circ}}^{\text{rep}}$ be another matching such that each gadget has two edges, one from K_L^{rep} and one from K_R^{rep}, where one of them is incident to the path unmatched by $\tilde{M}_{\overline{\circ}}^{\text{ext}}$. Let the inter-edges of $K_{\overline{\circ}}^{\text{ext}} \cup K_{\overline{\circ}}^{\text{rep}}$ be the non-edges $\tilde{N}_{\overline{\circ}}^{\text{gad}}$. We assert the maximality of matching $\tilde{M}_{\overline{\circ}}^{\text{rep}}$ among the edges of $K_{\overline{\circ}}^{\text{rep}}$ in each gadget by committing its edges that have both endpoints unmatched by $\tilde{M}_{\overline{\circ}}^{\text{rep}}$ to be the non-edges $\tilde{N}_{\overline{\circ}}^{\text{rep}}$. We assert the maximality of the matching $\tilde{M}_{\overline{\circ}}^{\text{ext}}$ among the edges of $K_{\overline{\circ}}^{\text{ext}}$ with the non-edges $\tilde{N}_{\overline{\circ}}^{\text{ext}}$ in two distinct cases of a gadget: $\tilde{N}_{\overline{\circ}}^{\text{ext}}$ is such that each gadget either has (1) the only two edges of $K_{\overline{\circ}}^{\text{ext}}$ with both endpoints unmatched by $\tilde{M}_{\overline{\circ}}^{\text{ext}}$, or (2) two edges of $K_{\overline{\circ}}^{\text{ext}}$ that form a length-8 path and are both incident to only the A-vertices or only the B-vertices of $\tilde{M}_{\overline{\circ}}^{\text{ext}}$. Note that the gadgets in the latter case are indeed maximal since, by construction of part $(A^{\overline{\circ}}, B^{\overline{\circ}})$, there are no 6-cycles among its query edges. Then, we define $\tilde{H}_3^{\overline{\circ}} = (A^{\overline{\circ}}, B^{\overline{\circ}}, \tilde{E}_2^{\overline{\circ}} \cup \tilde{M}_3^{\overline{\circ}}, \tilde{F}_2^{\overline{\circ}} \cup \tilde{N}_3^{\overline{\circ}})$ where $\tilde{N}_3^{\overline{\circ}} = \tilde{N}_{\overline{\circ}}^{\text{gad}} \cup \tilde{N}_{\overline{\circ}}^{\max}$. See Figures 8b and 8c for illustrations of the two cases of a gadget.

▶ **Lemma 30.** *Any structure graph $H_3^{\overline{\circ}}$ learned by the player is dominated by $\tilde{H}_3^{\overline{\circ}}$.*

By Lemmas 29 and 30, the player, whose structure graph is $\tilde{H}_2$ at the start of round 2, has its structure graph dominated by $\tilde{H}_3^{\circ} \dot\cup \tilde{H}_3^{\overline{\circ}}$ by the end of round 3. Note that, since there are no inter-gadget edges, $\tilde{H}_3^{\circ} \dot\cup \tilde{H}_3^{\overline{\circ}}$ is made up of the collection of gadgets $\mathcal{G} = \mathcal{G}^{\circ} \cup \mathcal{G}^{\overline{\circ}}$. As such, each gadget is either a gadget from $\tilde{H}_3^{\circ}$, the first case gadget from $\tilde{H}_3^{\overline{\circ}}$, or the second case gadget from $\tilde{H}_3^{\overline{\circ}}$. It follows then that there are at least $\frac{|\tilde{M}_1^{\circ}|}{4} - 1 + \frac{|\tilde{M}_1^{\overline{\circ}}|}{4} - \frac{1}{2} = \frac{N^{\circ}}{16} + \frac{N^{\overline{\circ}}}{16} - \frac{3}{2} = \frac{|\tilde{M}_1|}{4} - \frac{3}{2}$ many gadgets in $\mathcal{G}$ and, since each gadget has exactly four edges of $\tilde{M}_1 = \tilde{M}_1^{\circ} \cup \tilde{M}_1^{\overline{\circ}}$, there are at most $\frac{|\tilde{M}_1|}{4}$ many gadgets in $\mathcal{G}$.

▶ **Lemma 31.** *The largest matching in $\tilde{H}_3^{\circ} \dot\cup \tilde{H}_3^{\overline{\circ}}$ is of size at most $\frac{5 \cdot |\tilde{M}_1|}{4}$.*

Proof. Since each gadget is vertex-disjoint, we only need to consider the number of edges that each case of a gadget contributes to a largest output matching $\tilde{M}_3^{\text{out}}$. By Berge's theorem or similar, every edge with an endpoint of degree 1 is included in a largest output matching and thus it is easy to see that each gadget contributes exactly 5 edges to $\tilde{M}_3^{\text{out}}$. Finally, any edge of $\tilde{M}_1$ not included in a gadget contributes fewer edges, hence; we assume that all form part of a gadget, which implies that $|\mathcal{G}| = \frac{|\tilde{M}_1|}{4}$ and the result. ◀

▶ **Lemma 32.** *There exists a graph consistent with $\tilde{H}_3^{\circ} \dot\cup \tilde{H}_3^{\overline{\circ}}$ that has a maximum matching of size at least $2 \cdot |\tilde{M}_1| - 12$.*

Proof. Any graph consistent with $\tilde{H}_3^{\circ} \dot\cup \tilde{H}_3^{\overline{\circ}}$ may only consist of the edges within the vertex-disjoint gadgets. As such, we construct a maximum matching w.r.t. to each case of a gadget independently. Observe that all cases of a gadget, on 16 vertices, can be partitioned into 4 vertex-disjoint parts such that each consists of two A-vertices and two B-vertices, and they respectively correspond to the edges K_L^{ext}, K_R^{ext}, K_L^{rep} and K_R^{rep} unknown to the player who knows $\tilde{H}_2$. After query Q_3, at most a single non-edge f in each part is learned; hence, the remaining two edges incident to f can be used to construct a perfect matching in each part, which is thus a perfect matching, of size 8, in each gadget. Finally, we have that, since there are $|\mathcal{G}| \geq \frac{|\tilde{M}_1|}{4} - \frac{3}{2}$ many vertex-disjoint gadgets, the result holds. ◀

Overall, we have that the structure graph learned by the player after round 1 is dominated by $\tilde{H}_1$, for any arbitrary query Q_1. Then, in round 2, any arbitrary query Q_2 made by the player partitions the vertices into parts (A^+, B^+) and (A^-, B^-) such that the structure graphs learned in each part is dominated by the respective generalisations of $\tilde{H}_2$, i.e., $\tilde{H}_2^+$ and $\tilde{H}_2^-$, by the end of round 2. In round 3, each part is dominated by the respective generalisations of $\tilde{H}_3^\circ \,\dot\cup\, \tilde{H}_3^{\bar{\circ}}$, which we denote as $\tilde{H}_3^+$ and $\tilde{H}_3^-$, respectively, for any arbitrary query Q_3. As such, the player's overall information by the end of round 3 is $\tilde{H}_3^+ \,\dot\cup\, \tilde{H}_3^-$ for any arbitrary sequence of queries Q_1, Q_2, Q_3. Finally, we prove Lemma 33, which implies Theorem 15-3.

▶ **Lemma 33.** *The approximation factor of the structure graph $\tilde{H}_3^+ \,\dot\cup\, \tilde{H}_3^-$ is at most $\frac{5}{8} + \frac{24}{n}$.*

Proof. Recall that $\frac{n}{4}$ is odd; hence, $\tilde{H}_2^+$ and $\tilde{H}_2^-$ are such that, w.l.o.g., $|\tilde{M}_1^+|$ is even and $|\tilde{M}_1^-|$ is odd, and are dominated be the respective generalisations of $\tilde{H}_3^\circ \,\dot\cup\, \tilde{H}_3^{\bar{\circ}}$, that is, $\tilde{H}_3^+$ and $\tilde{H}_3^-$. As such, we consider both cases, particularly paying attention to $\tilde{H}_3^-$ and considering its isolated edge and vertices. It follows by Lemma 31 that $\tilde{H}_3^+$ has a largest matching of size at most $\frac{5\cdot|\tilde{M}_1^+|}{4}$; however, $\tilde{H}_3^-$ has one of size at most $\frac{5\cdot|\tilde{M}_1^-|}{4} - \frac{1}{4}$. Then, Lemma 32 immediately implies that $\tilde{H}_3^+$ has a consistent graph G_3^+ such that $\mu(G_3^+) \geq 2 \cdot |\tilde{M}_1^+| - 12$; however, it implies that $\tilde{H}_3^-$ has a consistent graph G_3^- such that $\mu(G_3^-) \geq 2 \cdot |\tilde{M}_1^-| - 13$. Finally, by Observations 19 and 20, the approximation factor of $\tilde{H}_3^+ \,\dot\cup\, \tilde{H}_3^-$ is at most $\frac{\frac{5}{4}|\tilde{M}_1| - \frac{1}{4}}{2\cdot|\tilde{M}_1| - 25} \leq \frac{5}{8} + \frac{8}{|\tilde{M}_1|} = \frac{5}{8} + \frac{24}{n}$ for large enough n. ◀

4 Conclusion

In this paper, we gave tight results on the approximation factor achievable by deterministic algorithms in the maximal matching edge-query model for up to 3 rounds. Our main result is a 0.625-approximation algorithm for MBM, which operates in three query rounds, and we proved that this is best possible. This algorithm can be implemented in the semi-streaming model and constitutes an improvement over the previously best 3-pass algorithm with approximation factor 0.6111 by Feldman and Szarf [13]. The best approximation factors achievable in one and two rounds are $\frac{1}{2}$ and $\frac{1}{2} + \Theta(\frac{1}{n})$, respectively, even in general graphs.

We conclude with three open questions:

1. *Randomization.* Our paper only considers deterministic query algorithms. Does randomization allow us to improve upon the results obtained in this paper?
2. *Adaptivity.* The algorithms considered in this paper are *adaptive* in the sense that the ith query can depend on the maximal matchings returned in rounds $1, \ldots, i-1$. Can we obtain interesting results if we allow multiple *non-adaptive* queries, i.e., queries that do not depend on the output produced by other queries?
3. *Semi-streaming Algorithms.* Is there a 3-pass semi-streaming algorithm for MBM with approximation factor better than 0.625 (that necessarily cannot be implemented as a deterministic edge-query algorithm)?

References

1 Raghavendra Addanki, Andrew McGregor, and Cameron Musco. Non-adaptive edge counting and sampling via bipartite independent set queries. *CoRR*, abs/2207.02817, 2022. doi:10.48550/arXiv.2207.02817.

2 Kook Jin Ahn and Sudipto Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput.*, 222:59–79, 2013. doi:10.1016/j.ic.2012.10.006.

3 Sepehr Assadi, Deeparnab Chakrabarty, and Sanjeev Khanna. Graph Connectivity and Single Element Recovery via Linear and OR Queries. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ESA.2021.7`.

4 Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Semi-streaming bipartite matching in fewer passes and optimal space. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9–12, 2022*, pages 627–669. SIAM, 2022. `doi:10.1137/1.9781611977073.29`.

5 Sepehr Assadi, S. Cliff Liu, and Robert E. Tarjan. An auction algorithm for bipartite matching in streaming and massively parallel computation models. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 165–171. SIAM, 2021. `doi:10.1137/1.9781611976496.18`.

6 Paul Beame, Sariel Har-Peled, Sivaramakrishnan Natarajan Ramamoorthy, Cyrus Rashtchian, and Makrand Sinha. Edge estimation with independent set oracles. *ACM Trans. Algorithms*, 16(4):52:1–52:27, 2020. `doi:10.1145/3404867`.

7 Soheil Behnezhad. Time-optimal sublinear algorithms for matching and vertex cover. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 873–884. IEEE, 2021. `doi:10.1109/FOCS52979.2021.00089`.

8 Lidiya Khalidah binti Khalil and Christian Konrad. Constructing large matchings via query access to a maximal matching oracle. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14-18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference)*, volume 182 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.FSTTCS.2020.26`.

9 Sebastian Eggert, Lasse Kliemann, Peter Munstermann, and Anand Srivastav. Bipartite matching in the semi-streaming model. *Algorithmica*, 63(1-2):490–508, 2012. `doi:10.1007/s00453-011-9556-8`.

10 Hossein Esfandiari, MohammadTaghi Hajiaghayi, and Morteza Monemizadeh. Finding large matchings in semi-streaming. In Carlotta Domeniconi, Francesco Gullo, Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu, editors, *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain*, pages 608–614. IEEE Computer Society, 2016. `doi:10.1109/ICDMW.2016.0092`.

11 Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM J. Comput.*, 35(4):964–984, 2006. `doi:10.1137/S0097539704447304`.

12 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005. `doi:10.1016/j.tcs.2005.09.013`.

13 Moran Feldman and Ariel Szarf. Maximum matching sans maximal matching: A new approach for finding maximum matchings in the data stream model. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 33:1–33:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.APPROX/RANDOM.2022.33`.

14 Oded Goldreich and Dana Ron. Approximating average parameters of graphs. *Random Struct. Algorithms*, 32(4):473–493, 2008. `doi:10.1002/rsa.20203`.

15 Sagar Kale and Sumedh Tirodkar. Maximum matching in two, three, and a few more passes over graph streams. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 15:1–15:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.15.

16 Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10–13, 2021*, pages 1874–1893. SIAM, 2021. doi:10.1137/1.9781611976465.112.

17 Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for mapreduce. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 938–948. SIAM, 2010. doi:10.1137/1.9781611973075.76.

18 Christian Konrad. A simple augmentation method for matchings with applications to streaming algorithms. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, volume 117 of *LIPIcs*, pages 74:1–74:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.74.

19 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques – 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, volume 7408 of *Lecture Notes in Computer Science*, pages 231–242. Springer, 2012. doi:10.1007/978-3-642-32512-0_20.

20 Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Rec.*, 43(1):9–20, 2014. doi:10.1145/2627692.2627694.

A Proofs of Lemmas 4–6

▶ **Lemma 4.** *There exists a maximum matching M^* such that $M^* \cup M_1$ has no even-length alternating paths or cycles.*

Proof. Let $P(M^*)$ be the set of even length alternating paths or cycles in $M^* \cup M_1$. If it is non-empty, we can use any $p \in P(M^*)$ to find another maximum matching $\hat{M}^*$.

Observe first that the edges of p alternate between edges in M_1 and M^* and is of even length. Therefore, we construct $\hat{M}^*$ from M^*, without decreasing its size, by replacing the edges of $M^* \cap p$ with the edges of $M_1 \cap p$. Finally, $\hat{M}^*$ is indeed a maximum matching since the edges of $M_1 \cap p$ are vertex disjoint from the edges of $M^* \backslash (M^* \cap p)$, otherwise p would not be an even length alternating path.

The only difference between the edges in $M^* \cup M_1$ and $\hat{M}^* \cup M_1$ are the removed edges $M^* \cap p$ which means that p is no longer an alternating path in $\hat{M}^* \cup M_1$ whereas all others remain unchanged. Therefore, $|P(\hat{M}^*)| = |P(M^*)| - 1$. Repeating this process until no such paths exist produces a maximum matching where the claim holds. ◀

▶ **Lemma 5.** *Each vertex of $A(M_1)$ and $B(M_1)$ is the endpoint of an edge in M^*.*

Proof. Since M^* is a maximal matching, every edge of M_1 must be incident to at least one of its edges. By Lemma 4, there are no even length alternating paths in $M^* \cup M_1$; hence, every $(a, b) \in M_1$ is either an edge of M^* or part of an augmenting path in $M^* \cup M_1$. In either case, a and b are each endpoints of exactly one edge of M^*. ◀

▶ **Lemma 6.** $|M^* \cap G_L| = |M^* \cap G_R| = (\frac{1}{2} - \epsilon) \cdot \mu(G)$.

Proof. Firstly, every edge of $M^* \oplus M_1$ is part of an alternating path or cycle since M_1 is maximal. Then, by Lemma 4, there are only odd length alternating paths, i.e., augmenting paths. Finally, any augmenting path must begin with an edge of M^* in G_L and end with one in G_R (or vice versa) with all other edges along the path belonging to G_{in}. See Figure 2 for an example. Thus, the number of edges of M^* in G_L and G_R, respectively, is the number of vertex-disjoint augmenting paths, which we subsequently show is exactly $(\frac{1}{2} - \epsilon) \cdot \mu(G)$ and thus implies the result.

Since M_1 and M^* are maximal matchings, every edge of M^* must be incident to at least one edge of M_1, and vice versa. In the case that $e \in M^* \cap M_1$, both these conditions are satisfied and e is an isolated edge in $M^* \cup M_1$ as both are matchings. Hence, in all other cases, the edges of M^* and M_1 belong to an alternating path in $M^* \oplus M_1$, all of which are necessarily vertex-disjoint since no two edges from a matching may share the same endpoint. Then, by Lemma 4, these are necessarily (odd length) augmenting paths. Finally, since M^* is a maximum matching of size $\mu(G)$ and each vertex-disjoint augmenting path increases the size of M_1 by 1, there must be exactly $\mu(G) - |M_1|$ many paths and the claim follows by Observation 3. ◀

B Reason for Assumption 21

▶ **Assumption 21.** *In each round $1 \leq i \leq r$, we assume that the query Q_i does not contain any edges or non-edges already learned by the player.*

Reason. Let $H = (A, B, E, F)$ be the structure graph known by the player. Let $e \in E$ and $f \in F$. If $e \in Q_i$, then the oracle can add e to the returned matching M_i without revealing any information about the edges incident to e that the player could have otherwise learned; thus, the query $Q_i \backslash \{e\}$ could never reveal less information than Q_i. If $f \in Q_i$, then f would never be in M_i; hence, $Q_i \backslash \{f\}$ would be an equivalent query. ◀

C Proofs of Lemmas 24–26 and 28–30

▶ **Lemma 24.** *$\tilde{H}_1$ is partitioned w.r.t. parts (A^+, B^+) and (A^-, B^-) into structure graphs $\tilde{H}_1^+ = \tilde{H}_1(A^+, B^+)$ and $\tilde{H}_1^- = \tilde{H}_1(A^-, B^-)$, respectively, where $\tilde{H}_1^+ \dot\cup \tilde{H}_1^-$ dominates $\tilde{H}_1$.*

Proof. Consider part (A^+, B^+) and its vertex-induced subgraph. Since (A^+, B^+) consists of the vertices of length-3 alternating paths w.r.t. $\tilde{M}_1$, the inclusion of the vertices of each path includes a matching edge from $\tilde{M}_1$ and two unmatched vertices, one in A_{out} and one in B_{out}. Thus, the matching edges $\tilde{M}_1^+ = \tilde{M}_1 \cap (A^+ \times B^+)$ and the non-edges $\tilde{N}_1^+ = \tilde{N}_1 \cap (A^+ \times B^+)$ are exactly the edges and non-edges of $\tilde{H}_1(A^+, B^+)$. Part (A^-, B^-) follows similarly since the remaining vertex-induced subgraph must have two unmatched vertices for every matching edge. Finally, since no edges of $\tilde{E}_1$ cross the partition, the disjoint union of each part dominates the original structure graph. ◀

▶ **Lemma 25.** *Any structure graph H_2^+ learned by the player is dominated by $\tilde{H}_2^+ = \tilde{H}_2(A^+, B^+)$.*

Proof. Let $Q_2^+ := Q_2 \cap (A^+ \times B^+)$ be the query edges relevant to part (A^+, B^+). By construction of the partition, $M^*(Q_2) \cap (A^+ \times B^+) \subseteq Q_2^+$ is a perfect matching in (A^+, B^+) such that every edge in $\tilde{M}_1^+$ is incident to two edges in $M^*(Q_2)$. Let W_L^+ be the perfect matching edges incident to $A_{\text{in}}^+ = A(\tilde{M}_1^+)$ and let W_R^+ be the ones incident to $B_{\text{in}}^+ = B(\tilde{M}_1^+)$.

As such, $\tilde{M}_1^+ \cup W_L^+ \cup W_R^+$ is exactly the edges of the the $|\tilde{M}_1^+|$ many vertex-disjoint length-3 paths used to construct part (A^+, B^+). Therefore, we can construct matchings $M_L^+ \subseteq W_L^+$ and $M_R^+ \subseteq W_R^+$ of size $\left\lfloor \frac{|\tilde{M}_1^+|}{2} \right\rfloor$ such that $\tilde{M}_1^+ \cup M_L^+ \cup M_R^+$ has *no* length-3 paths. We then assert the maximality of $M_2^+ = M_L^+ \cup M_R^+$ by committing $N_L^+ = A_{\text{in}}^+ \backslash A(M_L^+) \times B_{\text{out}}^+ \backslash B(M_L^+)$ and $N_R^+ = A_{\text{out}}^+ \backslash A(M_R^+) \times B_{\text{in}}^+ \backslash B(M_R^+)$ as non-edges.

Finally, we let σ be a graph isomorphism[4] that relates this to $\tilde{H}_2^+$ where $M_L^+ = \sigma(\tilde{M}_L^+)$ and $M_R^+ = \sigma(\tilde{M}_R^+)$. Then, we have that $M_2^+ \subseteq \sigma(\tilde{M}_2^+)$ and $N_2^+ = N_L^+ \cup N_R^+ \subseteq \sigma(\tilde{N}_2^+)$. ◀

▶ **Lemma 26.** *Any structure graph H_2^- learned by the player is dominated by $\tilde{H}_2^- = \tilde{H}_2(A^-, B^-)$.*

Proof. Let $Q_2^- := Q_2 \cap (A^- \times B^-)$ be the query edges relevant to part (A^-, B^-). By construction of the partition, $M^*(Q_2) \cap (A^- \times B^-) \subseteq Q_2^-$ is a maximum matching in (A^-, B^-) such that every edge in $\tilde{M}_1^-$ is incident to at most one edge in $M^*(Q_2)$. Let W_L^- be the matching edges incident to $A_{\text{in}}^- = A(\tilde{M}_1^-)$ and let W_R^- be the ones incident to $B_{\text{in}}^- = B(\tilde{M}_1^-)$. If $|W_L^-| \geq \left\lfloor \frac{|\tilde{M}_1^-|}{2} \right\rfloor$, then we let $M_L^- \subseteq W_L^-$ be of size $\left\lfloor \frac{|\tilde{M}_1^-|}{2} \right\rfloor$ and assert its maximality among the query edges $Q_2^- \cap (A_{\text{in}}^- \times B_{\text{out}}^-)$ by letting $N_L^- = A_{\text{in}}^- \backslash A(M_L^-) \times B_{\text{out}}^- \backslash B(M_L^-)$. Otherwise, if $|W_L^-| < \left\lfloor \frac{|\tilde{M}_1^-|}{2} \right\rfloor$, then we let $M_L^- = W_L^-$ which is trivially maximal; thus, $N_L^- = \emptyset$. We similarly consider W_R^- to construct the maximal matching M_R^- with non-edges N_R^-. Thus, there are no length-3 paths in $\tilde{M}_1^- \cup M_L^- \cup M_R^-$.

Finally, we let σ be a graph isomorphism that relates this to $\tilde{H}_2^-$ where $M_L^- \subseteq \sigma(\tilde{M}_L^-)$ and $M_R^- \subseteq \sigma(\tilde{M}_R^-)$; thus, we have that $M_2^- = M_L^- \cup M_R^- \subseteq \sigma(\tilde{M}_2^-)$ and $N_2^- = N_L^- \cup N_R^- \subseteq \sigma(\tilde{N}_2^-)$. ◀

▶ **Lemma 28.** *$\tilde{H}_2$ is partitioned w.r.t. parts (A°, B°) and $(A^{\overline{\circ}}, B^{\overline{\circ}})$ into structure graphs $\tilde{H}_2^\circ = \tilde{H}_2(A^\circ, B^\circ)$ and $\tilde{H}_2^{\overline{\circ}} = \tilde{H}_2(A^{\overline{\circ}}, B^{\overline{\circ}})$, respectively, where $\tilde{H}_2^\circ \dot\cup \tilde{H}_2^{\overline{\circ}}$ dominates $\tilde{H}_2$.*

Proof. Consider part (A°, B°) and its vertex-induced subgraph. Since (A°, B°) consists of the vertices of length-6 cycles each with a corresponding vertex unmatched by both matchings $\tilde{M}_1$ and $\tilde{M}_2$, we have that, for every two edges of $\tilde{M}_1$ included, two edges of $\tilde{M}_2$, one incident to A_{in} and one to B_{in}, and two unmatched vertices, one in A_{out} and one in B_{out}, are added to (A°, B°). Therefore, it is a generalisation of $\tilde{H}_2$. Part $(A^{\overline{\circ}}, B^{\overline{\circ}})$ follows similarly since the remaining vertices must have the same properties. ◀

▶ **Lemma 29.** *Any structure graph H_3° learned by the player is dominated by $\tilde{H}_3^\circ$.*

Proof. Let $Q_3^\circ = Q_3 \cap (A^\circ \times B^\circ)$ be any query w.r.t. part (A°, B°). By construction, every path in P° forms a 6-cycle with another one using the edges of $Q_\circ^{\text{ext}} = Q_3^\circ \cap K_\circ^{\text{ext}}$. Let C be a set representing the $\frac{|\tilde{M}_1^\circ|}{2}$ many 6-cycles. Each 6-cycle in C has one vertex in $\overline{A_{\text{in}}^\circ(P^\circ)}$ and one in $\overline{B_{\text{in}}^\circ(P^\circ)}$ such that every edge of $Q_\circ^{\text{rep}} = Q_3^\circ \cap K_\circ^{\text{rep}}$ is incident to exactly one of these vertices while the other is in $\overline{A_{\text{out}}^\circ(P^\circ)} \cup \overline{B_{\text{out}}^\circ(P^\circ)}$. As such, we represent each 6-cycle as an edge between its endpoints in $\overline{A_{\text{in}}^\circ(P^\circ)}$ and $\overline{B_{\text{in}}^\circ(P^\circ)}$ and delete all its other vertices. This exactly simulates a structure graph that, using a graph isomorphism σ', is dominated by the generalisation $\tilde{H}_1(A', B')$ where $A' = \overline{A_{\text{in}}^\circ(P^\circ)} \cup \overline{A_{\text{out}}^\circ(P^\circ)}$, $B' = \overline{B_{\text{in}}^\circ(P^\circ)} \cup \overline{B_{\text{out}}^\circ(P^\circ)}$ and C has a one-to-one correspondence with $\tilde{M}_1'$. There is then a one-to-one correspondence of the edges $K_\circ^{\text{rep}}$ to the edges $(A'_{\text{in}} \times B'_{\text{out}}) \cup (A'_{\text{out}} \times B'_{\text{in}})$; hence, any query Q_3^{rep} w.r.t. $\tilde{H}_2^\circ$ has a one-to-one correspondence to a query Q' w.r.t. $\tilde{H}_1(A', B')$.

[4] Note that the player knows $\tilde{H}_1$ at the start of round 2 and $\tilde{H}_2$ is specified w.r.t. $\tilde{H}_1$; thus, the graph isomorphism from round 1 is implicitly considered in H_2 and $\tilde{H}_2$.

Using the round 2 analysis in a white-box manner, we partition the vertices A' and B' into parts (A^+, B^+) and (A^-, B^-) using the query Q', which then, using isomorphisms σ^+ and σ^-, are dominated by $\tilde{H}_2^+$ and $\tilde{H}_2^-$ by Lemmas 25 and 26, respectively. Let C^+ and C^- be the corresponding partition of C w.r.t. the partitioning of $\tilde{M}_1'$ into $\tilde{M}_1^+$ and $\tilde{M}_1^-$. Let $\tilde{M}_\circ^+, \tilde{M}_\circ^-$ be the matchings and $\tilde{N}_\circ^+, \tilde{N}_\circ^-$ be the non-edges in $\tilde{H}_3^\circ$ that correspond to the matchings $\tilde{M}_2^+, \tilde{M}_2^-$ and the non-edges $\tilde{N}_2^+, \tilde{N}_2^-$ in $\tilde{H}_2^+, \tilde{H}_2^-$, respectively. Recall that $\tilde{M}_2^+$ (and $\tilde{M}_2^-$) has an equal number of edges incident to A'_{in} and B'_{in} such that each edge in $\tilde{M}_1^+$ (resp. $\tilde{M}_1^-$), except for at most one, is incident to exactly one edge in $\tilde{M}_2^+$ (resp. $\tilde{M}_2^-$); hence, $\tilde{M}_\circ^+$ (resp. $\tilde{M}_\circ^-$) has an equal number of edges in K_L^{rep} and K_R^{rep} such that each 6-cycle in C^+ (resp. C^-), except for at most one, is incident to exactly one edge in $\tilde{M}_\circ^+$ (resp. $\tilde{M}_\circ^-$).

We now construct each vertex-disjoint gadget of $\mathcal{G}^\circ$ to consist of the endpoints of two distinct edges in $\tilde{M}_\circ^+$ (or $\tilde{M}_\circ^-$), one from K_L^{rep} and one from K_R^{rep}, including the vertices of the two respective cycles in C^+ (resp. C^-) that they are incident to, and two distinct unmatched vertices corresponding to vertices in (A^+, B^+) (resp. (A^-, B^-)), one from $\overline{A^\circ_{\text{out}}(P^\circ)}$ and one from $\overline{B^\circ_{\text{out}}(P^\circ)}$. Thus, the non-edges $\tilde{N}_\circ^+$ (resp. $\tilde{N}_\circ^-$) are such that each gadget only has non-edges where both vertices are unmatched by $\tilde{M}_\circ^+$ (resp. $\tilde{M}_\circ^-$). Then, $\tilde{M}_\circ^+ \cup \tilde{M}_\circ^- = \tilde{M}_\circ^{\text{rep}}$ and $\tilde{N}_\circ^+ \cup \tilde{N}_\circ^- = \tilde{N}_\circ^{\text{rep}}$

We currently have that each gadget consists of two 6-cycles, each using two vertex-disjoint edges of $Q_\circ^{\text{ext}}$. If the number of 6-cycles $|C^+|$ (or $|C^-|$) in part (A^+, B^+) (resp. (A^-, B^-)) is odd, then there is one 6-cycle that would not be in any gadget, i.e., the one without an incident edge in $\tilde{M}_\circ^+$ (resp. $\tilde{M}_\circ^-$); hence, in any case, there are at least $\frac{|C^+|+|C^-|-2}{2} = \frac{|\tilde{M}_1^\circ|}{4} - 1$ many gadgets in $\mathcal{G}^\circ$. We complete each gadget by letting the edges of $Q_\circ^{\text{ext}}$ for only one of the 6-cycles be the edges of the matching $\tilde{M}_\circ^{\text{ext}}$ while the edges for the other are committed as the non-edges $\tilde{N}_\circ^{\text{ext}}$, which are vertex-disjoint from $\tilde{M}_\circ^{\text{ext}}$ since the 6-cycles are vertex-disjoint. Then, with the non-edges $\tilde{N}_\circ^{\text{gad}}$, we have that $\tilde{M}_\circ^{\text{ext}} \cup \tilde{M}_\circ^{\text{rep}} = \tilde{M}_3^\circ$ is a maximal matching since $\tilde{N}_\circ^{\text{rep}} \cup \tilde{N}_\circ^{\text{ext}} \cup \tilde{N}_\circ^{\text{gad}} = \tilde{N}_\circ^{\max} \cup \tilde{N}_\circ^{\text{gad}} = \tilde{N}_3^\circ$. Therefore, any structure graph H_3° learned w.r.t. the player's query Q_3° is dominated by $\tilde{H}_3^\circ$ such that $M_3^\circ \subseteq \sigma(\tilde{M}_3^\circ)$ and $N_3^\circ \subseteq \sigma(\tilde{N}_3^\circ)$ using the graph isomorphism $\sigma = \sigma' \circ \sigma^+ \circ \sigma^-$. ◀

▶ **Lemma 30.** *Any structure graph $H_3^{\bar{\circ}}$ learned by the player is dominated by $\tilde{H}_3^{\bar{\circ}}$.*

Proof. Before making a query, the player knows $\tilde{H}_2^{\bar{\circ}}$ in part $(A^{\bar{\circ}}, B^{\bar{\circ}})$; thus, we have that there are an equal number of paths in $P^{\bar{\circ}}$ that have their endpoints in $A^{\bar{\circ}}$ and in $B^{\bar{\circ}}$, which are also the same as the number of vertices in $\overline{A^{\bar{\circ}}_{\text{out}}(P^{\bar{\circ}})}$ and $\overline{B^{\bar{\circ}}_{\text{out}}(P^{\bar{\circ}})}$, respectively. As such, we first construct each vertex-disjoint gadget of $\mathcal{G}^{\bar{\circ}}$ to consist of the endpoints of four distinct paths in $P^{\bar{\circ}}$, two with their endpoints in $A^{\bar{\circ}}$ and two in $B^{\bar{\circ}}$, and four distinct vertices, two from $\overline{A^{\bar{\circ}}_{\text{out}}(P^{\bar{\circ}})}$ and two from $\overline{B^{\bar{\circ}}_{\text{out}}(P^{\bar{\circ}})}$. Since $|P^{\bar{\circ}}| = |\tilde{M}_1^{\bar{\circ}}|$ is even, there are $\left\lfloor \frac{|\tilde{M}_1^{\bar{\circ}}|}{4} \right\rfloor \geq \frac{|\tilde{M}_1^{\bar{\circ}}|}{4} - \frac{1}{2}$ many gadgets in $\mathcal{G}^{\bar{\circ}}$.

Let $Q_3^{\bar{\circ}} = Q_3 \cap (A^{\bar{\circ}} \times B^{\bar{\circ}})$ be any query w.r.t. part $(A^{\bar{\circ}}, B^{\bar{\circ}})$. At this point, each gadget of $\mathcal{G}^{\bar{\circ}}$ has the same structure and receives a subset of the query edges $Q_3^{\bar{\circ}}$ that is independent from the other gadgets. As such, it is sufficient to consider all possible queries $Q'_{\bar{\circ}} \subseteq K_{\bar{\circ}}^{\text{ext}} \cup K_{\bar{\circ}}^{\text{rep}}$ w.r.t. a single gadget. In general, $Q'_{\bar{\circ}}$ can be partitioned into four smaller parts: $Q_L^{\text{ext}} = Q'_{\bar{\circ}} \cap K_L^{\text{ext}}$, $Q_R^{\text{ext}} = Q'_{\bar{\circ}} \cap K_R^{\text{ext}}$, $Q_L^{\text{rep}} = Q'_{\bar{\circ}} \cap K_L^{\text{rep}}$ and $Q_R^{\text{rep}} = Q'_{\bar{\circ}} \cap K_R^{\text{rep}}$, each being any possible query w.r.t. two A-vertices and two B-vertices.

▷ Claim 34. Let Q be any arbitrary set of edges and let g be any arbitrary edge w.r.t. two A-vertices and two B-vertices where the player knows only the empty structure graph. Then, the structure graph with one edge $e \neq g$ and one non-edge f such that e and f are vertex-disjoint dominates the structure graph learned by the player after query Q.

Proof. If Q is the empty query, then no information is learned by the player and the claim holds. Otherwise, let $e \neq g \in Q$ be an arbitrary edge in the query, if one exists. Since there are only two A-vertices and two B-vertices, there exists only one possible edge f which is vertex-disjoint from e. If $f \in Q$ then we commit f as a non-edge, which makes the edge e a maximal matching. Otherwise, $f \notin Q$ and we have that e is already maximal. In the case where $\nexists e \neq g \in Q$, we commit $f = g$ as a non-edge, which implies that the empty matching is maximal since Q has no other edges. ◁

By Claim 34, we have that the player learns at most the edges $e_L^{\text{rep}} \in K_L^{\text{rep}}$ and $e_R^{\text{rep}} \in K_R^{\text{rep}}$, possibly with the non-edges $f_L^{\text{rep}} \in K_L^{\text{rep}}$ and $f_R^{\text{rep}} \in K_R^{\text{rep}}$, after the queries Q_L^{rep} and Q_R^{rep}, respectively. Therefore, the player learns the matching $M_{\bar{\circ}}^{\text{rep}} \subseteq \tilde{M}_{\bar{\circ}}^{\text{rep}} = \{e_L^{\text{rep}}, e_R^{\text{rep}}\}$ and non-edges $N_{\bar{\circ}}^{\text{rep}} \subseteq \tilde{N}_{\bar{\circ}}^{\text{rep}} = \{f_L^{\text{rep}}, f_R^{\text{rep}}\}$. As such, in each gadget, edge e_L^{rep} is incident to a path p with its endpoints in $B^{\bar{\circ}}$ while edge e_R^{rep} is incident to a path q with its endpoints in $A^{\bar{\circ}}$. Furthermore, we have that there exists a unique edge $g_L^{\text{ext}} \in K_L^{\text{ext}}$ and a unique edge $g_R^{\text{ext}} \in K_R^{\text{ext}}$ each of which are incident to p and q – we never want these edges to belong to the player's maximal matching $M_{\bar{\circ}}^{\text{ext}}$.

We now consider the pair of queries Q_L^{ext} and Q_R^{ext} that, by construction of part $(A^{\bar{\circ}}, B^{\bar{\circ}})$, do not contain edges that form 6-cycles with any paths in $P^{\bar{\circ}}$. This implies that if $g_L^{\text{ext}} \in Q_L^{\text{ext}}$ (or $g_R^{\text{ext}} \in Q_R^{\text{ext}}$) then $g_R^{\text{ext}} \notin Q_R^{\text{ext}}$ (resp. $g_L^{\text{ext}} \notin Q_L^{\text{ext}}$). Furthermore, since every pair of Q_L^{ext} and Q_R^{ext} has a symmetrical pair, we only need to consider the pairs of queries where $g_L^{\text{ext}} \notin Q_L^{\text{ext}}$ and show that the edges and non-edges learned in a single gadget are subsets of the edges $\tilde{M}_{\bar{\circ}}^{\text{ext}}$ and (either case of) the non-edges $\tilde{N}_{\bar{\circ}}^{\text{ext}}$ w.r.t. a single gadget.

If either query is empty, by Claim 34, we have that the edges, which avoid g_R^{ext}, and non-edges learned are thus subsets of a first case gadget in $\tilde{H}_3^{\bar{\circ}}$. Otherwise, both queries are non-empty and there exists query edges $e_L \neq g_L^{\text{ext}} \in Q_L^{\text{ext}}$ and $e_R \in Q_R^{\text{ext}}$. We then have that they either form a length-8 path in the gadget or do not. Consider first the simpler latter case where the query edges necessarily form two length-5 paths and are the only query edges since they do not form length-8 paths or 6-cycles. Let $M_{\bar{\circ}}^{\text{ext}} = \{e_L^{\text{ext}}\}$ and $N_{\bar{\circ}}^{\text{ext}} = \{f_R^{\text{ext}}\}$ where $e_L^{\text{ext}} = e_L$ and $f_R^{\text{ext}} = e_R$. $M_{\bar{\circ}}^{\text{ext}}$ is naturally maximal, and the edges and non-edges learned are a subset of the second case gadget in $\tilde{H}_3^{\bar{\circ}}$.

Consider now the case where the query edges $e_L \neq g_L^{\text{ext}}$ and e_R form a length-8 path. If $e_R \neq g_R^{\text{ext}}$, let $M_{\bar{\circ}}^{\text{ext}} = \{e_L^{\text{ext}}, e_R^{\text{ext}}\}$ where $e_L^{\text{ext}} = e_L$ and $e_R^{\text{ext}} = e_R$. Since $e_L^{\text{ext}} \neq g_L^{\text{ext}}$ the endpoints of either p or q in the gadget remain unmatched by $M_{\bar{\circ}}^{\text{ext}}$. Then, if they exist, we add the edges $f_L^{\text{ext}} \in Q_L^{\text{ext}}$ and $f_R^{\text{ext}} \in Q_R^{\text{ext}}$ that are unmatched by $M_{\bar{\circ}}^{\text{ext}}$ to an initially empty set of non-edges $N_{\bar{\circ}}^{\text{ext}}$; hence, $M_{\bar{\circ}}^{\text{ext}}$ is maximal and the edges and non-edges learned are a subset of the first case gadget in $\tilde{H}_3^{\bar{\circ}}$. Otherwise, we have that $e_R = g_R^{\text{ext}}$ and let $N_{\bar{\circ}}^{\text{ext}} = \{f_L^{\text{ext}}, f_R^{\text{ext}}\}$ where $f_L^{\text{ext}} = e_L$ and $f_R^{\text{ext}} = e_R$. Any possible remaining query edges must be either only incident to or only not incident to $N_{\bar{\circ}}^{\text{ext}}$ since they may not form any 6-cycles. We pick at most one from Q_L^{ext} and one from Q_R^{ext} then add them to an initially empty matching $M_{\bar{\circ}}^{\text{ext}}$, which is thus maximal and, if both exist, forms a length-8 path that leaves the endpoints of either p or q in the gadget unmatched since $M_{\bar{\circ}}^{\text{ext}}$ does not contain g_L^{ext} or g_R^{ext}. If $M_{\bar{\circ}}^{\text{ext}}$ and $N_{\bar{\circ}}^{\text{ext}}$ are incident to each other, then the edges and non-edges learned are a subset of the second case gadget in $\tilde{H}_3^{\bar{\circ}}$. Otherwise, they are a subset of the first case gadget in $\tilde{H}_3^{\bar{\circ}}$. ◀