Bipartite Matching in Massive Graphs: A Tight Analysis of EDCS

Amir Azarmehr [*1] **Soheil Behnezhad** [*1] **Mohammad Roghani** [*2]

Abstract

Maximum matching is one of the most fundamental combinatorial optimization problems with applications in various contexts such as balanced clustering, data mining, resource allocation, and online advertisement. In many of these applications, the input graph is massive. The sheer size of these inputs makes it impossible to store the whole graph in the memory of a single machine and process it there. Graph sparsification has been an extremely powerful tool to alleviate this problem. In this paper, we study a highly successful and versatile sparsifier for the matching problem: the *edge-degree constrained subgraph (EDCS)* introduced first by Bernstein & Stein 2015. The EDCS has a parameter $\beta \geq 2$ which controls the density of the sparsifier. It has been shown through various proofs in the literature that by picking a subgraph with $O(n\beta)$ edges, the EDCS includes a matching of size at least $2/3 - O(1/\beta)$ times the maximum matching size. As such, by increasing β the approximation ratio of EDCS gets closer and closer to $2/3$. In this paper, we propose a new approach for analyzing the approximation ratio of EDCS. Our analysis is *tight* for any value of β. Namely, we pinpoint the precise approximation ratio of EDCS for any sparsity parameter β. Our analysis reveals that one does not necessarily need to increase β to improve approximation, as suggested by previous analysis. In particular, the best choice turns out to be $\beta = 6$, which achieves an approximation ratio of .677! This is arguably surprising as it is even better than $2/3 \sim .666$, the bound that was widely believed to be the limit for EDCS.

[*]Equal contribution [1]Khoury College of Computer Science, Northeastern University, Boston, USA [2]Department of Management Science and Engineering, Stanford University, Stanford, USA. Correspondence to: Amir Azarmehr <azarmehr.a@northeastern.edu>, Soheil Behnezhad <s.behnezhad@northeastern.edu>, Mohammad Roghani <roghani@stanford.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Maximum matching is one of the most fundamental combinatorial optimization problems. Recall that a *matching* in a graph is a collection of edges that do not share any vertices. A *maximum* matching is a matching of the largest possible size.

The matching problem finds applications in various contexts such as data mining, resource allocation, online advertisement, bioinformatics, and many others. For instance, maximum matching can improve the quality of data clustering (Assadi et al., 2019b), it can produce fair k-center clustering (Jones et al., 2020), or can be used to discover subgraphs for bioinformatics applications (Berger et al., 2008; Langmead & Donald, 2004). In most of these applications, the input graph is massive. The sheer size of these inputs makes it impossible to store the whole graph in the memory of a single machine and process it there. This has motivated a large and beautiful body of work over the past two decades on large-scale algorithms for this problem.

Graph Sparsification: Graph sparsification is a powerful tool to process massive graphs. A graph sparsifier receives an n-vertex graph that may have as many as $\Omega(n^2)$ edges and sparsifies it into a sparse subgraph, say with $O(n)$ edges, that preserves some property of it. Graph sparsifiers have been instrumental tools for various graph problems. Cut sparsifiers (Nagamochi & Ibaraki, 1992), spectral sparsifiers (Spielman & Teng, 2011), and spanners (Abboud & Bodwin, 2017) are some famous examples. Graph sparsifiers did not find many applications for matchings until nearly a decade ago when Bernstein & Stein 2015 introduced the *edge-degree constrained subgraph (EDCS)*. See in particular the nice paper of Assadi & Bernstein 2019 for an in-depth introduction to EDCS and an overview of some of its applications.

Over the years, the EDCS has been successfully applied to a variety of large-scale settings including the massively parallel computations (MPC) setting which is a common theoretical model of MapReduce-style computation (Assadi et al., 2019a), the dynamic setting (Bernstein & Stein, 2015; Behnezhad & Khanna, 2022; Roghani et al., 2022), the streaming setting (Bernstein, 2020; Assadi & Behnezhad, 2021), the sublinear time setting (Behnezhad et al., 2023;

Bhattacharya et al., 2023), communication complexity (Azarmehr & Behnezhad, 2023), and the stochastic matching setting (Assadi & Bernstein, 2019).

In this paper, we revisit the key property of EDCS: that it obtains a good approximation of maximum matching while at the same time being sparse. To put our results into perspective, we first need to provide some background and overview existing bounds.

Background

Let us start by stating the formal definition of EDCS.

Definition 1.1 (Bernstein & Stein 2015)**.** Given a graph G, a subgraph $H \subseteq G$ is an *edge-degree constrained subgraph* with parameters (β, β^-), or a (β, β^-)-EDCS, if the following conditions hold:

1. for all edges $(u, v) \in H$, $\deg_H(u) + \deg_H(v) \leq \beta$,
2. for all $(u, v) \in G \setminus H$, $\deg_H(u) + \deg_H(v) \geq \beta^-$.

The following proposition shows that a (β, β^-)-EDCS always exists for all integers $\beta > \beta^- \geq 1$.

Proposition 1.2 (Bernstein & Stein 2015)**.** *Any graph G contains a (β, β^-)-EDCS for any integers $\beta > \beta^- \geq 1$, and one can be found greedily in polynomial time.*

Moreover, since the edge-degrees in a (β, β^-)-EDCS are all upper bounded by β by the first property of Definition 1.1, so are the vertex degrees. Therefore, the EDCS has at most $O(n\beta)$ edges. This means that smaller values of β are more desirable as the subgraph picked will be a sparser.

It will be instructive to set $\beta = 2$ and $\beta^- = 1$. It can be easily confirmed that any $(2, 1)$-EDCS is a *maximal matching* (i.e., a matching that is not a subset of another matching) and that any maximal matching is a $(2, 1)$-EDCS. It is well-known that a maximal matching has at least half as many edges as a maximum matching and that this bound is tight.[1] The key property of EDCS is that by slightly increasing β and keeping β^- close to it, the approximation ratio improves to almost 2/3. Formally:

Proposition 1.3 (Bernstein & Stein 2015; 2016; Assadi & Bernstein 2019; Behnezhad 2021)**.** *Given a graph G, and parameters $\beta \geq 1/\varepsilon$ and $\beta^- \geq (1-\varepsilon)\beta$, any (β, β^-)-EDCS of G contains a $(2/3 - O(\varepsilon))$-approximate maximum matching of G.*

The approximation guarantee of Proposition 1.3 is close to optimal. In particular, for infinitely many choices of

$$1 \leq \beta^- < \beta,$$

[1]Take G to be a path with 3 edges and take H to be the subgraph only containing the middle edge.

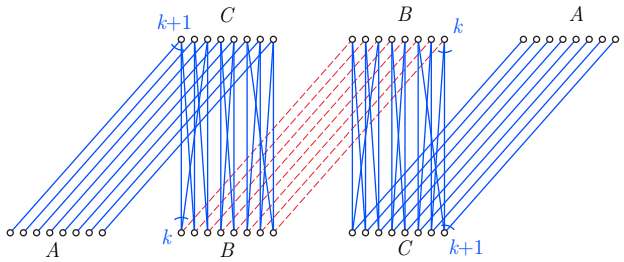

Figure 1. An example due to (Bernstein & Stein, 2015) where a $(\beta, \beta-1)$-EDCS does not obtain a better than 2/3-approximation for any odd $\beta = 2k+1$. Here we have a bipartite graph, each side having three equally sized vertex sets A, B, C. The solid blue lines denote the EDCS edges, and the dashed red lines denote the edges not in the EDCS. The vertices in B have degree k in the EDCS, the vertices in C have degree $k+1$ in the EDCS, and the vertices in A have degree one. Note that any edge in the EDCS has edge degree at most $\beta = 2k+1$. The only edges missing from the EDCS are the dashed B-B edges which all have edge degree exactly $\beta - 1 = 2k$. While the graph G has a perfect matching, the EDCS can only match $2/3$ of the vertices.

(particularly for all odd $\beta = 2k+1$ and all $\beta^- < \beta$) examples have been known since the original paper of (Bernstein & Stein, 2015) where a (β, β^-)-EDCS does not include a better than 2/3-approximation. See Figure 1.

Our Contribution

In most applications of EDCS, we would like to set β to be as small as possible to achieve sparser subgraphs.[2] On the other hand, making β smaller would make the approximation guarantee of Proposition 1.3 worse. It is therefore natural to study the trade-off between β and the approximation ratio. Unfortunately, known proofs are too loose, especially, for small values of β. In this paper, we propose a new approach to analyze the approximation ratio of EDCS. Our analysis is tight and precisely pinpoints the exact approximation ratio achieved for any given β. Table 1 states the approximation ratio for various values of β and β^-.

Our Results: Our analysis reveals that the approximation ratio of (β, β^-)-EDCS, when $\beta^- = \beta - 1$, behaves very differently from when $\beta^- < \beta - 1$. Note that this is still in the regime where a (β, β^-)-EDCS exists and can be found in polynomial time. For instance, for any odd value of $\beta \geq 7$, a $(\beta, \beta-1)$-EDCS obtains an exact $2/3$-approximation. The approximation turns out to be quite surprising when considering even values of $\beta \geq 6$. For instance, a $(6, 5)$-

[2]The only exception is the deterministic dynamic algorithm of (Bernstein & Stein, 2015) for maintaining EDCS where larger values of β help as they make the EDCS more "robust" to changes.

EDCS obtains a .677-approximation which is even better than $2/3 \sim .666$! While this may seem to contradict the example of Figure 1, it has to be noted that Figure 1 requires β to be odd and does not work when it is even. Moreover, as we increase β in the even case, the approximation ratio gets worse and approaches $2/3$ (see Figure 4). We note that when β is even, $\beta - 1$, which is the edge-degree lower bound for missed edges, is odd. This means that the two endpoints of such missed edges must have different degrees. Such imbalance between the degrees of the vertices with missed optimal edges is precisely the reason for a better than 2/3 approximation. When we increase β, this imbalance becomes less significant (as the ratio of degrees gets closer and closer to 1) and so the approximation becomes worse. This is in sharp contrast with previous analysis such as the one in Proposition 1.3 where increasing β improves the approximation.

We emphasize that going beyond 2/3-approximation for the maximum matching problem is considered a difficult task in many settings. We refer the interested reader to (Assadi & Behnezhad, 2021; Behnezhad et al., 2023) where the precise problem of beating 2/3-approximation is studied in various settings. We hope that our discovery that a $(6, 5)$-EDCS beats 2/3-approximation combined with the known bound of Proposition 1.2 that such EDCS's can be found via a simple greedy algorithm paves the way for future progress on this important question.

Our Analysis: While previous analysis of EDCS were analytical, our analysis is based on a new factor-revealing linear program (formalized as LP 1) that we show provides the exact approximation ratio of (β, β^-)-EDCS for any given parameters β, β^-.

We then provide the claimed approximation guarantees by solving this LP. We note that a factor revealing LP has also been used to analyze a hierarchical version of EDCS in (Behnezhad & Khanna, 2022). However, the factor revealing LP there is different from ours and, importantly, is not tight. For instance, the LP used by (Behnezhad & Khanna, 2022) only guarantees a 0.6-approximation for a $(6, 5)$-EDCS which is way smaller than the correct bound of 0.677 returned by our tight LP.

2. Preliminaries

In this section, we introduce the notations and definitions we use, and provide some background on matchings.

A graph G is *bipartite* if its vertices can be partitioned into two sets L and R, such that every edge has exactly one endpoint in L and one endpoint in R. We use $G(L, R)$ to denote a bipartite graph with partitions L and R.

Given a set of vertices A, we use $N_G(A)$ (or $N(A)$ when G is clear from the context) to denote the set of its neighbors. For a vertex u, we use $\deg_G(u)$ to denote its degree in G.

Given a graph G, a *matching* is a subset of edges such that no two edges share an endpoint. We say that a matching M covers a vertex u, or that u is matched in M, if there is an edge adjacent to u in M. A *maximum matching* is a matching with the largest possible number of edges. We use $\mu(G)$ to denote the size of a maximum matching in G.

The following is a well-known primal-dual result for bipartite maximum matching.

Proposition 2.1 (Extended Hall's Theorem)**.** *Given a bipartite graph $G(L, R)$, it holds that*

$$\mu(G) = \min_{A \subseteq L} |N(A)| + |L \setminus A| \,.$$

The vertex set that minimizes the right-hand side is referred to as a Hall's witness *for G. Furthermore, for every maximum matching M and Hall's witness A, every edge of M covers exactly one vertex in $N(A) \cup (L \setminus A)$.*

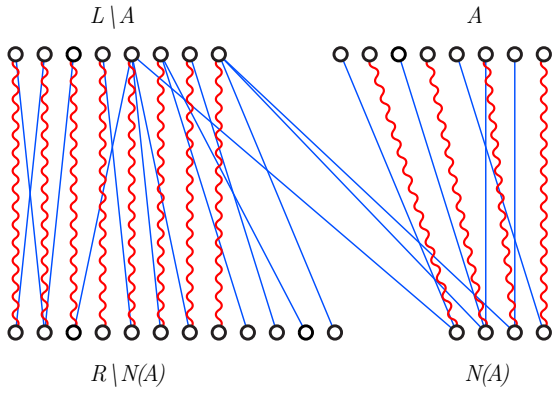

Figure 2. An example of Hall's witness. The curvy red edges denote a maximum matching, and A is a Hall's witness. Generally, considering a vertex set $A \subseteq L$ and any matching, the vertices in A are matched to a subset of $N(A)$. Therefore, even if all the vertices in $L \setminus A$ are somehow matched, the matching has size at most $|N(A)| + |L \setminus A|$, i.e. $\mu(G) \leq |N(A)| + |L \setminus A|$. Hall's theorem states that there exists a vertex set A, referred to as a Hall's witness, for which this inequality is tight.

3. A New Analysis of the EDCS via a Factor-Revealing LP

In this section, we present the linear program that "reveals" the approximation ratio of EDCS for fixed parameters (β, β^-). We prove that it is tight, i.e. any pair (G, H) consisting of a bipartite graph $G(L, R)$ and a (β, β^-)-EDCS $H \subseteq G$ can be converted to a feasible solution of the LP and vice versa.

The edges and the vertices of the graph are divided into groups based on their properties and their role in the graph (e.g. for a vertex this includes its degree, whether it is matched in the maximum matching, where it is in the Hall's witness, etc.) We refer to these properties as a *vertex profile* or an *edge profile*, and use VP and EP to denote the set of valid vertex profiles and edge profiles, respectively. Then, a variable is designated to each group of vertices or edges with the same profile. Its value is set (proportionally) to the number of vertices or edges in that group.

The easiest way to understand the LP is to see how a pair (G, H) is converted to a feasible solution, where G is a bipartite graph, and H is a (β, β^-)-EDCS for G. First, fix a maximum matching M^* of G and a maximum matching M of H, along with a Hall's witness A for H, i.e. $|M| = \mu(H) = |N_H(A)| + |L \setminus A|$. Without loss of generality, we can assume that G contains no edges other than $M^* \cup H$. Because those edges can be removed from the graph, in which case $\mu(G)$ and $\mu(H)$ remain unchanged while H is still a (β, β^-)-EDCS.

We divide the vertices and the edges into different groups. All the vertices or edges in a group have the same properties, a.k.a. profile, which will be defined shortly. Each variable of the LP then reflects how many vertices or edges with each profile there are in the graph. We scale all the numbers by $1/\mu(H)$, that is if there are k vertices with a certain profile, then the variable corresponding to that vertex profile holds the value $k/\mu(H)$. As a result, the variables corresponding to the edges in M sum to 1, the variables corresponding to the edges in M^* sum to the approximation ratio $\mu(G)/\mu(H)$ which we set as the objective function of our LP to be *maximized*.

Now, we define the vertex profiles. To do so, we consider all the possible cases of the following properties for a vertex:

- whether it is in A, $L \setminus A$, $N_H(A)$, or $R \setminus N_H(A)$,
- its degree in H, an integer between 0 and $\beta - 1$,
- whether it is matched in M, and
- whether it is matched in M^*.

Since we create a variable for each vertex profile, we make sure to use only valid vertex profiles, so that we do not create "extra" variables. That is, we have to confirm that the properties above in a profile make sense together. We have a total of ten validity conditions, two for the vertex profiles and (as explained later) eight for the edge profiles. These conditions are enforced when formulating the LP, and they are not a part of the LP itself. Specifically, for the vertex profiles, we assert the following:

1. if a vertex is in $N_H(A)$ or $L \setminus A$, then it must be matched in M, because M is a maximum matching of H and A is a Hall's witness for H (see Proposition 2.1), and
2. if a vertex has zero degree in H, then it must be unmatched in M, since having an edge in M would mean having degree at least 1.

After considering all the cases and discarding the vertex profiles that do not satisfy the aforementioned conditions, we create a variable for each vertex profile. Observe that any vertex in the graph has a valid profile, i.e. its properties satisfy the two conditions. Finally, to set the values in our feasible solution, if there are n_{v} vertices with profile v, we let the variable x_{v} corresponding to that vertex profile be equal to $n_{\mathsf{v}}/\mu(H)$.

To define an edge profile, we consider all the possible cases of the following properties for an edge:

- the vertex profiles of its endpoints in L and R,
- whether it is in H,
- whether it is in M, and
- whether it is in M^*.

Checking the validity of an edge profile takes more work. This is where we enforce the bulk of the properties that G, H, and M have. Note that if a profile is invalid, we do not create a variable for it at all. First, some simple consistency conditions:

3. if the edge is in H, then its endpoint vertex profiles must indicate a nonzero degree in H,
4. if the edge is in M (resp. M^*), its endpoint vertex profiles must indicate that they are matched in M (resp. M^*),
5. if an edge is in M, it must be (by definition) in H, and
6. there are no edges outside $H \cup M^*$ (see the beginning of Section 3).

Then, the conditions concerning the EDCS H (see Definition 1.1):

7. if an edge is in H, the degrees of its endpoints in H must sum to at most β, and
8. if an edge is not in H, the degrees of its endpoints in H must sum to at least β^-.

Finally, the conditions for the Hall's witness A (see Proposition 2.1):

9. there should be no edges of H between A and $R \setminus N_H(A)$ (by definition), and
10. for every edge of M, *exactly* one of the following holds: (i) it has an endpoint in $N(A)$, or (ii) it has an endpoint in $L \setminus A$.

Similar to the vertices, we create a variable for each valid edge profile (i.e. an edge profile that satisfies all the conditions), and set their values proportional to the number of edges with that profile. That is, if there are n_{e} edges with profile e, the variable x_{e} corresponding to that profile is set equal to $\frac{n_{\mathsf{e}}}{\mu(H)}$. This completes the explanation of how the variables are meant to correspond to a graph.

To tie this all together, we need to add the constraints. Most of the properties of G, H, and M are already encoded in the vertex/edge profiles. The purpose of the constraints is to link the number of vertices to the number of edges adjacent to them, and to ensure that the variables corresponding to M sum to 1, i.e. everything is scaled by $1/\mu(H)$.

Recall, VP and EP denote the set of valid vertex profiles and edge profiles, respectively. For every profile $\mathsf{v} \in VP$ (resp. $\mathsf{e} \in EP$), we create a variable, and denote it by x_{v} (resp. x_{e}). We use $H(\mathsf{v})$ (resp. $M(\mathsf{v})$, $M^*(\mathsf{v})$) to denote the set of edge profiles that are in H (resp. M, M^*) and one of their endpoints has profile v. We also use $\mathsf{e} \in M^*$ to denote that the edges with profile e are in M^*, and $\mathsf{v}_{\deg}$ to denote the degree of vertices with profile v in H. The LP is then as follows:

LP 1: The factor-revealing LP for the approximation ratio of (β, β^-)-EDCS

$$
\begin{aligned}
\text{maximize} \quad & \sum_{\substack{\mathsf{e} \in EP \\ \mathsf{e} \in M^*}} x_{\mathsf{e}} & \\
& \sum_{\mathsf{e} \in H(\mathsf{v})} x_{\mathsf{e}} = \mathsf{v}_{\deg} \cdot x_{\mathsf{v}} & \forall \mathsf{v} \in VP \\
& \sum_{\mathsf{e} \in M(\mathsf{v})} x_{\mathsf{e}} = x_{\mathsf{v}} & \forall \mathsf{v} \in VP \text{ that is matched in } M \\
& \sum_{\mathsf{e} \in M^*(\mathsf{v})} x_{\mathsf{e}} = x_{\mathsf{v}} & \forall \mathsf{v} \in VP \text{ that is matched in } M^* \\
& \sum_{\mathsf{e} \in M} x_{\mathsf{e}} = 1 & \\
& x_{\mathsf{e}} \geq 0 & \forall \mathsf{e} \in EP \\
& x_{\mathsf{v}} \geq 0 & \forall \mathsf{v} \in VP
\end{aligned}
$$

To see how the first three constraints tie the vertex variables to edge variables, take the first set of constraints as an example. These constraints state that for a vertex profile that has degree d in H, if there are k vertices with this profile, then there should be $d \cdot k$ edges of H adjacent to these vertices. The fourth constraint simply states that everything is scaled by $1/\mu(H)$, therefore the variables corresponding to the edges of M sum to 1 (recall $|M| = \mu(H)$). With that, we are ready to state the main theorem of this section.

Theorem 3.1. *The optimal objective value for LP 1 is equal to the approximation ratio of (β, β^-)-EDCS. That is, if the optimal value for LP 1 is r, then for every bipartite graph G and every (β, β^-)-EDCS $H \subseteq G$, it holds that $r \cdot \mu(H) \geq \mu(G)$. Furthermore, there exists an instance where this inequality is tight.*

We prove the theorem by showing the following two claims hold.

Claim 3.2. *The optimal value of LP 1 is an upper bound for the approximation ratio of (β, β^-)-EDCS. That is, any pair of a bipartite graph G and (β, β^-)-EDCS $H \subseteq G$ can be converted to a feasible solution of LP 1 such that the objective value is equal to $\mu(G)/\mu(H)$.*

Proof. The reduction has been partially explained. Given G and H, we fix a maximum matching M^* in G, a maximum matching M in H, and a Hall's witness A for H. Then for every vertex profile v (similarly for every edge profile e), if the number of vertices with that profile is n_{v}, we set the value of the corresponding variable x_{v} equal to $n_{\mathsf{v}}/\mu(H)$. Note that any vertex (similarly edge) of G corresponds to exactly one vertex profile and the constraints hold automatically. Therefore, x is feasible.

Now we calculate the objective value for x (denote it by r).

$$r = \sum_{\mathsf{e} \in M^*} x_{\mathsf{e}} = \frac{\sum_{\mathsf{e} \in M^*} n_{\mathsf{e}}}{\mu(H)} = \frac{\mu(G)}{\mu(H)},$$

which concludes the proof. □

Claim 3.3. *The optimal value of LP 1 is a lower bound for the approximation ratio of (β, β^-)-EDCS. That is, an optimal solution of LP 1 with objective value r can be converted to a bipartite graph G and a (β, β^-)-EDCS $H \subseteq G$ such that $\mu(G)/\mu(H) = r$.*

Proof. Take a rational optimal solution x (note that since the coefficients in the constraints are rational, there exists a rational optimal solution to the LP). Because x is rational, there exists a positive integer N such that $n_{\mathsf{v}} = N \cdot x_{\mathsf{v}}$ and $n_{\mathsf{e}} = N \cdot x_{\mathsf{e}}$ is an integer for all profile v and e.

We create a graph G with exactly n_{v} vertices with profile v, and n_{e} edges with profile e. To start, we create a group of n_{v}

vertices for each vertex profile v. To connect them, for each edge profile e we use n_{e} edges between the two endpoint groups of e (recall that each edge profile indicates the vertex profiles of its endpoints). This can be done because of the LP constraints that link the number of edges to the number of vertices. More specifically, one can start with the edge profiles of M^* and match unmatched vertices from the endpoint groups. Then, move on to the edge profiles of $M \setminus M^*$ and match unmatched vertices (w.r.t. M) from the endpoint groups. Finally, go over the edge profiles of $H \setminus (M \cup M^*)$ and connect the vertices from the endpoint groups to achieve their designated degree in H.

Now that we have G, we define H, M, M^*, and A simply by considering what the vertex/edge profiles indicate. H is a (β, β^-)-EDCS of G because of the EDCS validity conditions for edge profiles (conditions 7 and 8). M is a maximum matching of H since it is coupled with the Hall's witness A. That is, there is a vertex set $A \subseteq L$ such that each edge of M matches exactly one vertex from $(L \setminus A) \cup N(A)$ (conditions 1, 9, and 10). Finally, M^* is a maximum matching since otherwise we could choose a larger matching as M^* and derive a feasible solution with a larger objective value, which contradicts optimality.

Now we calculate the approximation ratio in this instance:

$$\frac{\mu(G)}{\mu(H)} = \frac{\sum_{\mathsf{e} \in M^*} n_{\mathsf{e}}}{\mu(H)} = \sum_{\mathsf{e} \in M^*} x_{\mathsf{e}}.$$

The right-hand side is the objective value for x, i.e. the optimal value of LP 1, which concludes the proof. □

Putting the two claims together, gives Theorem 3.1.

4. Numerical Solutions

4.1. Setup

All of our code[3] is written in Python (version 3.10.12) and is available in the supplementary material. For solving factor-revealing LP instances, we utilized the Gurobi optimization package (version 11.0.0). The experiments were conducted on a computing cluster equipped with 64 cores, each running at 2.30GHz on Intel(R) Xeon(R) processors, and with 756 GiB of main memory. The operating system used was Ubuntu 22.04.3 LTS.

4.2. Results

We implement the factor-revealing LP in Section 3 for all possible values of $\beta, \beta^- \in [1, 100]$ where $\beta > \beta^-$. The largest approximation ratio that we obtain using the factor-revealing LP is 0.6774 which is achieved for parameter $\beta = 6$ and $\beta^- = 5$ (see Table 1 and Figure 3).

[3]The implemented code can be found at the following link.

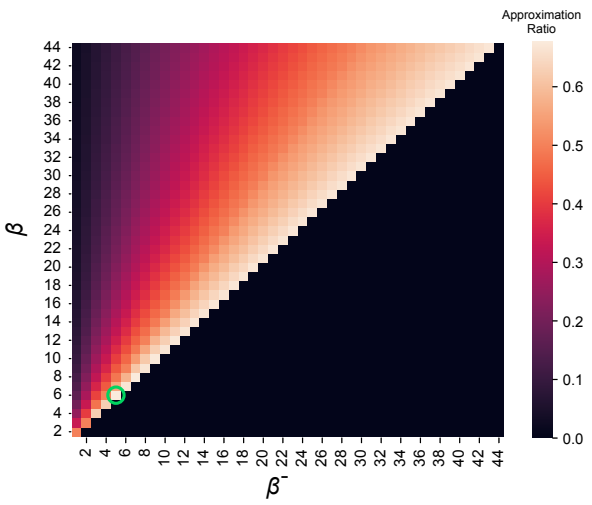

Figure 3. Heatmap shows the approximation ratio of (β, β^-)-EDCS for different values of β^- (x-axis) and β (y-axis). The best approximation ratio is shown in a green circle for $\beta = 6$ and $\beta^- = 5$.

The result of the factor-revealing LP shows that for $(\beta, \beta - 1)$-EDCS when β is an even number larger than 4, the approximation ratio of the EDCS is larger than 2/3 and as β grows, the approximation ratio converges to 2/3 (see Figure 4).

On the other hand, for all other values of (β, β^-), the approximation ratio is always below 2/3, and for a constant integer $c \geq 2$, the approximation ratio of $(\beta, \beta - c)$-EDCS converge to 2/3 as β goes to infinity (see Figure 5 and Table 2).

5. Conclusion

Over the recent years, EDCS has proven to be a successful matching sparsifier in various applications, including MPC model, stochastic matching model, sublinear time model, dynamic model, and streaming model. Many state-of-the-art results have been achieved by employing EDCS as a matching sparsifier in these settings.

It is well-known that for large values of β and β^-, the approximation ratio of EDCS is $2/3 - \varepsilon$. This paper provides a tight analysis of the approximation ratio of (β, β^-)-EDCS for small values of β and β^- using a factor-revealing linear program. Remarkably, we discover that when β is even, the $(\beta, \beta - 1)$-EDCS has an approximation ratio greater than 2/3, a previously unknown result. Our findings reveal that the maximum achievable approximation ratio is 0.6774 when $\beta = 6$ and $\beta^- = 5$. We hope the discovery that a (6, 5)-EDCS surpasses the 2/3 approximation, combined

$\beta \backslash \beta^-$	1	2	3	4	5	6	7	8	9	10	11
2	0.5	-	-	-	-	-	-	-	-	-	-
3	0.3333	0.5	-	-	-	-	-	-	-	-	-
4	0.25	0.4	0.625	-	-	-	-	-	-	-	-
5	0.2	0.3333	0.4782	0.6249	-	-	-	-	-	-	-
6	0.1666	0.2857	0.4117	0.5	0.6774	-	-	-	-	-	-
7	0.1428	0.25	0.3617	0.4444	0.5604	0.6666	-	-	-	-	-
8	0.125	0.2222	0.3225	0.4	0.4827	0.5783	0.6756	-	-	-	-
9	0.1111	0.2	0.2911	0.3636	0.4399	0.5	0.6097	0.6666	-	-	-
10	0.1	0.1818	0.2653	0.3333	0.4042	0.4615	0.539	0.6153	0.6721	-	-
11	0.0909	0.1666	0.2436	0.3076	0.3739	0.4285	0.4862	0.5569	0.625	0.6666	-
12	0.0833	0.1538	0.2253	0.2857	0.3478	0.3999	0.4545	0.5	0.5796	0.625	0.6703

Table 1. The approximation ratio achieved by EDCS obtained from the factor-revealing LP in Section 3. Rows correspond to β, and columns correspond to β^-. The table presents the approximation ratio for all possible (β, β^-)-EDCS when $1 \leq \beta^- < \beta \leq 12$. A hyphen in a cell of the table indicates that there is no EDCS with the corresponding parameters.

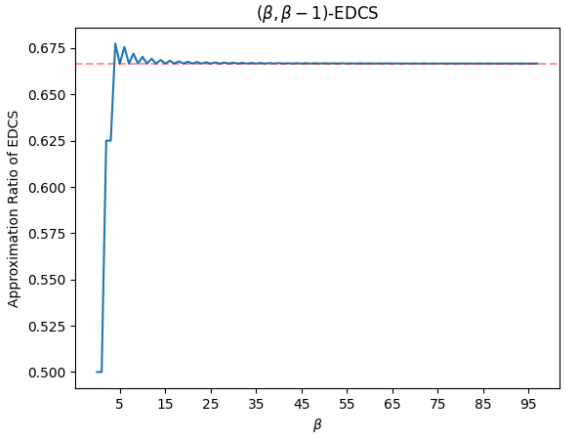

Figure 4. The approximation ratio of $(\beta, \beta - 1)$-EDCS is computed for all possible values of $\beta \leq 100$. The y-axis denotes the approximation ratio, while the x-axis corresponds to the values of β. The horizontal red dashed line represents the approximation ratio $2/3$, which was previously believed to be the best possible approximation ratio of EDCS. The largest approximation ratio occurs when $\beta = 6$.

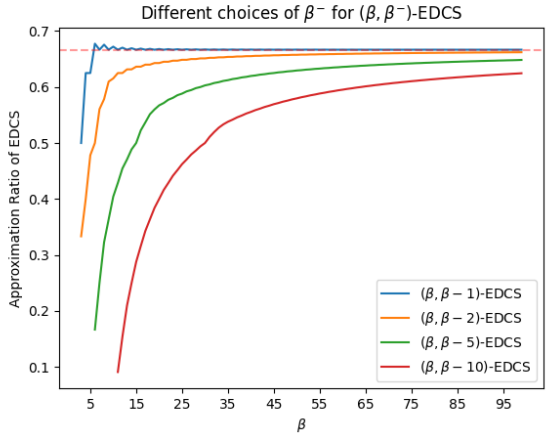

Figure 5. The approximation ratio of $(\beta, \beta - c)$-EDCS is calculated for various choices of $c \in [1, 2, 5, 10]$ across all feasible values of $\beta \leq 100$. The y-axis represents the approximation ratio, and the x-axis corresponds to the values of β. The horizontal red dashed line represents the approximation ratio $2/3$. As c increases, the approximation ratio worsens, and the convergence to 2/3 occurs at a slower rate.

with the known bound that EDCS can be obtained through a simple greedy algorithm, opens avenues for future advancements in solving the maximum matching problem in different settings.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abboud, A. and Bodwin, G. The 4/3 additive spanner exponent is tight. *J. ACM*, 64(4):28:1–28:20, 2017. doi: 10.1145/3088511. URL https://doi.org/10.1145/3088511.

Assadi, S. and Behnezhad, S. Beating two-thirds for random-order streaming matching. In Bansal, N., Merelli, E., and Worrell, J. (eds.), *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pp. 19:1–19:13. Schloss

β	$(\beta, \beta-1)$-EDCS	$(\beta, \beta-2)$-EDCS
5	0.6249	0.4782
6	0.6774	0.5
7	0.6666	0.5604
8	0.6756	0.5783
9	0.6666	0.6097
10	0.6721	0.6153
20	0.6678	0.6428
30	0.6671	0.6511
40	0.6669	0.6551
50	0.6668	0.6575
60	0.6667	0.659
70	0.6667	0.6601
80	0.6667	0.661
90	0.6667	0.6616
100	0.6667	0.6621

Table 2. The approximation ratio achieved by EDCS using the factor-revealing LP in Section 3. Rows correspond to β, and columns correspond to instances of the EDCS that we are using (either $(\beta, \beta-1)$-EDCS or $(\beta, \beta-2)$-EDCS). The table presents the approximation ratio for some of the values of $\beta \leq 100$. (β, β^-)-EDCS when $1 \leq \beta^- < \beta < 13$. As discussed before, for $(\beta, \beta-1)$-EDCS, the approximation ratio is larger than 2/3 when β is even and it converges to 2/3. On the other hand, the approximation ratio of $(\beta, \beta-2)$-EDCS is always smaller than 2/3 and converges to 2/3 as β grows.

Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPICS.ICALP.2021.19. URL https://doi.org/10.4230/LIPIcs.ICALP.2021.19.

Assadi, S. and Bernstein, A. Towards a unified theory of sparsification for matching problems. In Fineman, J. T. and Mitzenmacher, M. (eds.), *2nd Symposium on Simplicity in Algorithms, SOSA 2019, January 8-9, 2019, San Diego, CA, USA*, volume 69 of *OASIcs*, pp. 11:1–11:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi: 10.4230/OASICS.SOSA.2019.11.

Assadi, S., Bateni, M., Bernstein, A., Mirrokni, V. S., and Stein, C. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In Chan, T. M. (ed.), *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 1616–1635. SIAM, 2019a. doi: 10.1137/1.9781611975482.98.

Assadi, S., Bateni, M., and Mirrokni, V. S. Distributed weighted matching via randomized composable coresets. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 333–343. PMLR, 2019b. URL http://proceedings.mlr.press/v97/assadi19a.html.

Azarmehr, A. and Behnezhad, S. Robust communication complexity of matching: EDCS achieves 5/6 approximation. In Etessami, K., Feige, U., and Puppis, G. (eds.), *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pp. 14:1–14:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPICS.ICALP.2023.14. URL https://doi.org/10.4230/LIPIcs.ICALP.2023.14.

Behnezhad, S. Improved analysis of EDCS via gallai-edmonds decomposition. *CoRR*, abs/2110.05746, 2021.

Behnezhad, S. and Khanna, S. New trade-offs for fully dynamic matching via hierarchical EDCS. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pp. 3529–3566, 2022. doi: 10.1137/1.9781611977073.140. URL https://doi.org/10.1137/1.9781611977073.140.

Behnezhad, S., Roghani, M., and Rubinstein, A. Sublinear time algorithms and complexity of approximate maximum matching. In Saha, B. and Servedio, R. A. (eds.), *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pp. 267–280. ACM, 2023. doi: 10.1145/3564246.3585231. URL https://doi.org/10.1145/3564246.3585231.

Berger, B., Singh, R., and Xu, J. Graph algorithms for biological systems analysis. In Teng, S. (ed.), *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pp. 142–151. SIAM, 2008. URL http://dl.acm.org/citation.cfm?id=1347082.1347098.

Bernstein, A. Improved bounds for matching in random-order streams. In Czumaj, A., Dawar, A., and Merelli, E. (eds.), *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pp. 12:1–12:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPICS.ICALP.2020.12. URL https://doi.org/10.4230/LIPIcs.ICALP.2020.12.

Bernstein, A. and Stein, C. Fully dynamic matching in bipartite graphs. In Halldórsson, M. M., Iwama, K., Kobayashi, N., and Speckmann, B. (eds.), *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in*

Computer Science, pp. 167–179. Springer, 2015. doi: 10.1007/978-3-662-47672-7_14.

Bernstein, A. and Stein, C. Faster fully dynamic matchings with small approximation ratios. In Krauthgamer, R. (ed.), *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pp. 692–711. SIAM, 2016. doi: 10.1137/1.9781611974331.CH50.

Bhattacharya, S., Kiss, P., and Saranurak, T. Sublinear algorithms for (1.5+ε)-approximate matching. In Saha, B. and Servedio, R. A. (eds.), *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pp. 254–266. ACM, 2023. doi: 10.1145/3564246.3585252. URL https://doi.org/10.1145/3564246.3585252.

Jones, M., Nguyen, H. L., and Nguyen, T. D. Fair k-centers via maximum matching. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4940–4949. PMLR, 2020. URL http://proceedings.mlr.press/v119/jones20a.html.

Langmead, C. J. and Donald, B. R. High-throughput 3d structural homology detection via NMR resonance assignment. In *3rd International IEEE Computer Society Computational Systems Bioinformatics Conference, CSB 2004, Stanford, CA, USA, August 16-19, 2004*, pp. 278–289. IEEE Computer Society, 2004. doi: 10.1109/CSB.2004.1332441. URL https://doi.org/10.1109/CSB.2004.1332441.

Nagamochi, H. and Ibaraki, T. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5&6):583–596, 1992. doi: 10.1007/BF01758778. URL https://doi.org/10.1007/BF01758778.

Roghani, M., Saberi, A., and Wajc, D. Beating the folklore algorithm for dynamic matching. In Braverman, M. (ed.), *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pp. 111:1–111:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPICS.ITCS.2022.111. URL https://doi.org/10.4230/LIPIcs.ITCS.2022.111.

Spielman, D. A. and Teng, S. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011. doi: 10.1137/08074489X. URL https://doi.org/10.1137/08074489X.