Master Thesis

Theory of Computation Laboratory 4

Coresets for Graph Matching

Author:
Gilbert Maystre
gilbert@maystre.ch

Supervisors:
Prof. Michael Kapralov
Jakab Tardos

January 17, 2020

École Polytechnique Fédérale de Lausanne

Preface

Abstract

This master thesis focuses on randomized composable coresets as building blocks for memory-efficient algorithms in the context of massive graphs. Randomized composable coresets were initiated by [16] in the context of submodular function maximization and have later been successfully used for the graph matching problem [20].

We improve on the analysis of [20] for the maximum matching coreset with *regular* graphs, proving 1/2-approximation in the general case and 1-approximation in the bipartite case with a few other results for intermediate regimes (2/3 for triangle-free graphs and 3/4 for C_3/C_5-free graphs). We then propose a simple extension to the maximum matching coreset that works well on some hard instances.

We also propose a new coreset for bipartite graphs with size $\mathcal{O}(n)$ which is based on the notion of server flows of [21]. We develop several new structural results for server flows and prove the sanity of our coreset. Finally, we we give a lower bound of 1/2 (in expectation) and an upper bound of 2/3 on its approximation ratio.

Acknowledgments

I would like to thank Prof. M. Kapralov for guidance, hosting me in his lab and proposing this topic of research, especially the idea of using server flows to build efficient coresets. I would also like to thank J. Tardos for continuous help as well as Dr. L. Maystre for discussions on probability theory.

Thesis organization

The first chapter is devoted to briefly introducing and motivating the concepts that we will use all along. In particular, we present the different models used to devise sublinear algorithms and spend some time exploring the relatively new concept of randomized composable coresets. We also present the maximum matching problem and state a few important results. Finally, we describe the current landscape of the literature relating to maximum matching two models of interest.

Chapter two is dedicated to improving results for coresets which are based on maximum matching. Specifically, we prove several new bounds for the case of regular graph. We also propose a simple extension to the maximum matching coreset and show that it performs well on some graphs that are hard for the classical maximum matching coreset.

In the third chapter, we propose a new coreset named the *server flow coreset*. We first develop the theory of server flows by proving several new structural results. We then present how to use server flows as guides to build efficient coreset and prove a 1/2-approximation for the coreset in general bipartite graph. Finally, we raise the question of the limits of this new coreset and prove a 2/3 upper bound on its approximation ratio.

We conclude the thesis by stating some open questions and presenting omitted proofs.

Notation

For the whole thesis, we use standard algorithm and graph theory notation. We would like to grab the reader's attention to the following notational convention in particular.

- If G is a graph, we write $E(G)$ for its edge set and $V(G)$ for its vertex set if not previously named. Also, for a set of edge M, we let $V(M)$ be the set of endpoints of M. Finally, for any $S_1, S_2 \subseteq V$, we write $E[S_1, S_2]$ to denote the edges of E having an end in S_1 and the other in S_2.

- A bipartite graph is denoted by $G = (C, S, E)$ where $C \cup S$ is the bipartition. Elements of C are sometime denoted as *clients* and those of S as *servers*.

- The neighborhood of a vertex v is denoted by $\Gamma(v)$ and its degree by $\delta(v) = |\Gamma(v)|$. If the graph is not clear from context, we will specify it using a subscript.

- For any $n \in \mathbb{N}$, C_n denotes the cycle on n vertices.

- If M is a set and e some element, we use $M + e$ as a shortcut for the heavier $M \cup \{e\}$ and $M - e$ for $M \setminus \{e\}$.

- We use $\widetilde{\mathcal{O}}(f(n))$ to hide polylogarithmic factors. In standard big O notation, it would correspond to $\mathcal{O}(f(n) \cdot \text{polylog}(n))$.

- Let Ω be some set and $\mathbf{x} \in \mathbb{R}^{|\Omega|}$ be a vector indexed by Ω. The support of $\mathbf{x}$ is defined by $\text{support}(\mathbf{x}) = \{\omega \in \Omega : x_\omega \neq 0\}$.

Contents

Chapter 1

Introduction

1.1 Algorithms and large datasets

As the size of datasets increase (for instance, the Facebook graph has two billion nodes while Googles serves more than three billion queries each day) and their use becomes more popular, there is a dramatic need for algorithms that are capable of handling such sizes. Since the early 2010s, several new programming paradigms have emerged to responds to those exigencies and their theoretical counter-parts are developing in parallel. This thesis is about those.

In the classical computational model (Turing machines), the input is written before execution and in its totality in the computer memory. This poorly fits the needs of the massive datasets paradigm, where the memory use must be vastly smaller than the input. In light of this, many new models, more or less inspired by already existing technology were proposed. We introduce here briefly two of main interest that we tailor to the case of the input being a graph.

In the **Streaming Model**, the algorithm knows in advance the name of the n nodes and sees the edges of the graph as a stream. Each time an edge is presented, the algorithm can do some computation and modify its memory state. At any time, the memory size should be kept small. This model mimics what happens for instance at a router on the Internet's backbone: it sees a *flow* of packets and can keep little information. The typical metric to asses the quality of a streaming algorithm is its memory consumption as well as the update time[1]. In the context of graph streaming, *small* memory is usually thought of $\widetilde{\mathcal{O}}(n)$, i.e, vastly sublinear in the number of edge. Interested readers can read the survey of McGregor [14]

In the more recent **Massively Parallel Communication** (MPC) model of Karloff et al. [7] (as well as [8], [13] and [12]), there are several computing nodes that effectuate rounds of computation. Between each round, all the servers synchronize and they can exchange messages. Typically, each server only holds a tiny fraction of the whole input. This is a direct abstraction of the *MapReduce* and other *Hadoop*-like frameworks. The metrics of interest are the memory size of each server and the number of rounds. Another one is the total size of communication, which can usually be derived from the first two.

[1] The time the algorithm needs to process a new incoming element.

1.2 The maximum matching problem

Suppose that as a teacher you have prepared an assignment to be done in pairs. Naturally, the goal is to create as much pairs as possible within your class: that's less correction time at the end! On the other hand, you want to keep everyone happy and so, will only accept to pair two students that go along well. How to optimize the number of pairs under those constraints? This problem and many other fall into the realm of the **maximum matching** problem and the affinity constraints are best expressed using a graph.

In this section, we define formally what a matching is and then list a few basic properties of matchings that will be used throughout this thesis. No proof will be provided, but the interested reader can find them and other basic facts in the book of Diestel [22] or West [3].

Definition 1. Let $G = (V, E)$ be a graph. A **matching** is a subset of the edges $M \subseteq E$ such that each vertex $v \in V$ is adjacent to at most one edge $e \in M$. The **size** of the matching M is simply $|M|$. A matching is **maximum** if there exists no larger matching. A matching is **maximal** if it cannot be extended. More precisely, there exists no $e \in E \setminus M$ such that $M + e$ is a valid matching. We indicate the size of a maximum matching of G with $\mathsf{MM}(G)$.

As usual, any maximum matching is also maximal but the converse is not true. The maximum matching problem, i.e., given a graph find a maximum matching, has been studied for a long time as it has many real-life applications. Beside matching students, it can be interesting to match other things, like jobs with applicants or ads with internet users.

Jack Edmonds showed in 1961 that the maximum matching problem admits a poly-time algorithm [1]. His work used a central notion of matching theory, namely the one of augmenting path that we define next.

Definition 2. Let $G = (V, E)$ be a graph and $M \subseteq E$ a matching. We say that $P = (v_1, v_2, \ldots, u_{2\ell})$ is an **augmenting path with respect to** M if:

1. $\{v_{2i-1}, v_{2i}\} \in E \setminus M$ for all $i \in [\ell]$.
2. $\{v_{2i}, v_{2i+1}\} \in M$ for all $i \in [\ell - 1]$
3. v_1 and $v_{2\ell}$ are both unmatched by M

Note that an augmenting path has an odd number of edge and effectively alternates between edges of M and $\overline{M}$.

If such a path exists then flipping its edges[2] is good way to augment the size of M. See Figure 1.2 for an example. It turns out that not having any augmenting path is also a sufficient optimality condition, as the following says.

Lemma 1. [Berge (1957)] Let G be a graph and $M \subseteq E(G)$ a matching. M is maximum if and only if there exists no augmenting path with respect to M.

Most classical algorithms for finding maximum matching are actually based on the above observation, where a matching is incrementally improved by finding augmenting paths and flipping

[2] For each edge in the path, what was in M is now in $\overline{M}$ and vice-versa

them (e.g. Edmond's blossom algorithm). In the context of bipartite graph, there also exists another condition that guarantees optimality, namely Hall's matching condition.

Lemma 2. [Hall (1935)] A bipartite graph $G = (C, S, E)$ has a matching that matches all clients if and only if for any $T \subseteq C$, we have $|\Gamma(T)| \geq |T|$.

We now turn our attention to the dual of the maximum matching problem: the minimum vertex cover problem, where the goal is to find a cover of minimum size.

Definition 3. Let $G = (V, E)$ be a graph. A **vertex cover** is a subset $C \subseteq V$ such that each edge $e \in E$ has at least one end in C (so that C effectively *covers* each edge). The **size** of a vertex cover is simply $|C|$ and we indicate the size of the minimum vertex cover of G with $\mathsf{VC}(G)$.

Although the maximum matching problem is polynomial, the vertex cover one is NP-hard. The maximum matching problem and the vertex cover problem are linked via the duality of their respective linear programming formulation. In particular, we have the following relationships.

Lemma 3. For any graph G, we have $\mathsf{MM}(G) \leq \mathsf{VC}(G) \leq 2 \cdot \mathsf{MM}(G)$.

A direct corollary of the above fact is that it is easy to approximate the minimum vertex cover problem within a factor 2. Interestingly, it is impossible to go beyond this ratio provided that the unique game conjecture holds (Khot & Regev [6]). In the bipartite case however, things are simpler thanks to König's theorem:

Lemma 4. [König (1931)] For any bipartite graph G, we have $\mathsf{MM}(G) = \mathsf{VC}(G)$.

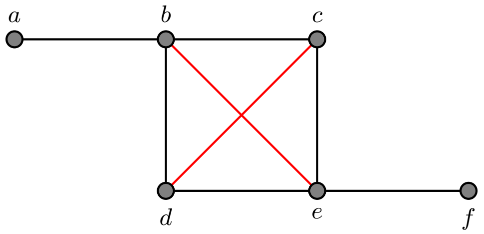

Figure 1.1: A graph together with a (maximal) matching M depicted in red. $P = (a,\ b,\ e,\ f)$ is an augmenting path with respect to M so in light of Lemma 1, M is not maximum and indeed flipping P yields a larger matching. A minimum vertex cover is $C = \{b,\ c,\ e\}$.

1.3 Randomized composable coresets

As explained in the introduction, several new computational models were proposed to better respond to the challenge of massive datasets. There is a large need to unify somewhat these models and in this section we present one of the proposed approach to produce good algorithms in a variety of models simultaneously. The approach that we will study in this thesis is a versatile building block, coined as *composable coreset*. The idea stems from the following divide and conquer heuristics for large datasets:

i Split the data into chunks.

ii Process each chunks separately, yielding many small-space sub-solutions.

iii Aggregate all small-space solution to make a final solution.

We call each of the small solutions a **composable coreset**. This idea was applied with success in several different optimization contexts (see [16] for a small summary), however it suffered many harsh impossibility results for a range of interesting problems including submodular function maximization and maximum matching[3]. Those disappointing hardness results usually stem from an adversarial initial split of the data.

Mirrokni & Zadimoghaddam proposed in [16] to use distributional assumptions on the initial splitting to circumvent those shortcomings. In particular, they coined the term ***randomized* composable coreset** (or simply *coreset* for this thesis) when the distribution is uniform. As we will see, asking for a uniform splitting is not that much and translates especially well in the streaming and MPC model.

1.3.1 Definitions

Let us now formaly define randomized composable coresets. We follow somewhat the definitions of Assadi & Khanna in [19] but tailor them to the graph matching setting.

Definition 4. Let $G = (V, E)$ be a graph and $k \in \mathbb{N}$. A **random k-partition of G** is a set of k random subgraphs $G^{(1)} = (V, E^{(1)})$, $G^{(2)} = (V, E^{(2)})$, $\cdots$, $G^{(k)} = (V, E^{(k)})$ of G where each edge $e \in E$ is sent independently and uniformly at random to exactly one of $\{E^{(1)}, E^{(2)}, \ldots, E^{(k)}\}$.

Definition 5. Let $\mathcal{A}$ be an algorithm that takes as input a graph H and returns a subgraph $\mathcal{A}(H) \subseteq H$. We say that $\mathcal{A}$ outputs an **α-approximate randomized composable coreset for the maximum matching problem** if given any graph $G = (V, E)$, any $k \in \mathbb{N}$ and a random k-partition of G, we have

$$\mathsf{MM}\Big(\mathcal{A}\Big(G^{(1)}\Big) \cup \ldots \cup \mathcal{A}\Big(G^{(k)}\Big)\Big) \geq \alpha \cdot \mathsf{MM}(G)$$

with high probability (over the random k-partition). The **size of the coreset** is defined as the number of edges returned by $\mathcal{A}$.

[3] For the maximum matching problem, if the splitting is abritrary, then the coreset must essentialy be the whole graph even to achieve only a ratio of $\mathsf{polylog}(n)$ (see [17] for details).

Note that in the above definition, α is naturally some number of $[0,1]$ and we keep this convention for the whole thesis. We also stress that the concept of randomized composable coreset is not to be restricted to the maximum matching problem. For instance, MM could be replaced by any other graph theoretic function f. It is also interesting to note that authors of [16] define the approximation ratio of a randomized composable coreset using expectation instead of "with high[4] probability". More precisely, in their definition, $\mathcal{A}$ is an α-approximate randomized composable coreset if the following holds:

$$\mathbb{E}\Big[\mathsf{MM}\Big(\mathcal{A}\Big(G^{(1)}\Big) \cup \ldots \cup \mathcal{A}\Big(G^{(k)}\Big)\Big)\Big] \geq \alpha \cdot \mathsf{MM}(G)$$

where the expectation is taken over the random k-partition. This is arguably weaker than Definition 5 but we will use it nevertheless in Section 2.3 and Chapter 3.

1.3.2 From coresets to classical models

We now argue that randomized composable coresets are good building blocks for the two different memory-conscious models discussed in introduction.

Lemma 5. [Section 1.1, [19]] Let G be a graph with n vertices and m edges and f a graph-theoretic function (e.g. $f = \mathsf{MM}$). If $\mathcal{A}$ produces an α-approximate randomized composable coreset of size s for the function f, then:

1. There exists an MPC algorithm that can approximate f within a factor α in two rounds using $\mathcal{O}\Big(\sqrt{m/s}\Big)$ machines each with $\mathcal{O}(\sqrt{ms}+n)$ memory with high probability.

2. There exists a single-pass streaming algorithm (where arrival is assumed to be *random*) that outputs an α-approximation of f with high probability using memory $\mathcal{O}(\sqrt{ms})$.

Proof. For the MPC model, we specify the two map/reduce rounds as follows:

1.M Each mapper receives their share of the input. If the load is well balanced, it represents roughly $m/\sqrt{m/s} = \sqrt{ms}$ memory per machine. They map each edge randomly to one of the $\sqrt{m/s}$ mappers. This steps mimics the construction of a random k-partition of G.

1.R Each reducer gets a small graph (with high probability it has $\mathcal{O}(\sqrt{ms})$ edges), runs $\mathcal{A}$ on it and sends the result with size s.

2.M Each mapper receives s edges and route them to a single designated mapper without modification

2.R The chosen mapper receives $\sqrt{m/s} \times s = \sqrt{ms}$ data and outputs a solution to f given its graph.

Note that each machine actually needs $\mathcal{O}(\sqrt{ms}+n)$ space. The additional n, is necessary to store the vertices. The answer that the last mapper yields is an α-approximation by definition of $\mathcal{A}$.

For the streaming model with random arrival, the algorithm stores $\sqrt{ms}$ edges, applies $\mathcal{A}$ to this first block and store the solution which has size s. Each of the subsequent blocks is processed in the same way. When the $\sqrt{m/s}$ blocks are treated (the algorithm has had a chance to

[4] i.e, inverse polynomial

see the $\sqrt{ms}\sqrt{m/s} = m$ edges), the memory is actually filled with $\sqrt{m/s}$ individual solution of size s (so the overall memory constraint of $\mathcal{O}(\sqrt{ms})$ is fulfilled). Those solutions are then unioned, a global solution is computed using any classical tool and the output is α-approximate by definition of coreset again. □

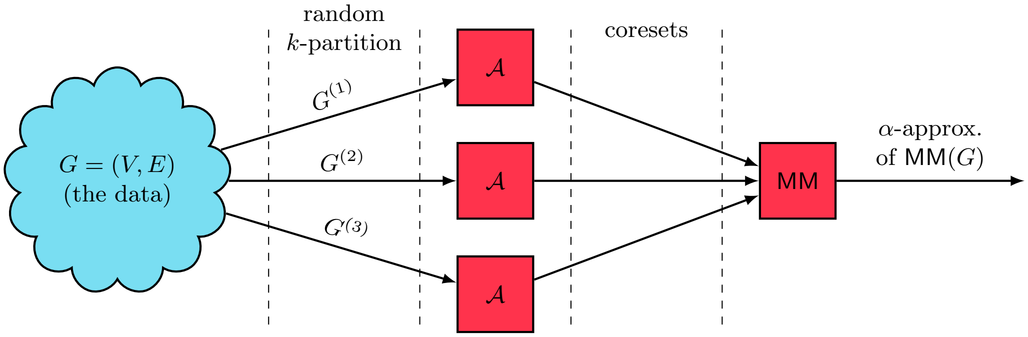

Figure 1.2: Illustration of the concept of randomized composable coresets in a pipeline-like fashion. Here, three coresets are used and the function to optimize is the maximum matching one.

1.4 Landscape of known results

Even though the complexity of the maximum matching problem is quite well understood in the classical model, there is still a lot of work to be done in the sublinear setting. We give here a quick overview of the current literature for the two models mentioned in introduction.

In the semi-streaming model[5] with one pass, the greedy algorithm proposed by Feigenbaum et al. in their seminal article [4] remains unbeaten. A small improvement with expected approximation ratio 0.503 for general graphs was proposed a decade later by Konrad et al. in [11]. Their model has the further assumption that edges arrive uniformly at random in the stream.

Michael Kapralov was able to prove many lower bounds and particularly that no single-pass semi-streaming algorithm with arbitrary arrival can achieve an approximation ratio better than $1 - 1/e \approx 0.632$ [10].

In the MPC model, a first algorithm using $\Theta(n)$ memory and $\mathcal{O}(\log(n))$ rounds (or $\Theta(n^{1+c})$ memory and $\mathcal{O}(1)$ round for any constant $c > 0$) achieving an approximation ratio of 1/2 was proposed by Lattanzi et al. in [9]. In 2018, Czumaj et al. used a different approach to get an approximation ratio of 1/2 with $\mathcal{O}(n)$ of memory per computing node but only $\mathcal{O}\big((\log\log(n))^2\big)$ communication rounds [23].

Even though randomized composable coresets are not tailored precisely for one model, considering their use for the maximum matching problem was able to improve results in both of the model. In particular, Assadi et al. designed a coreset based on edge degree constraint subgraph with size $\widetilde{\mathcal{O}}(n)$ and approximation ratio 2/3 [20]. This directly yields an MPC algorithm using $\mathcal{O}\big(n^{1.5}\big)$ memory per computer, two rounds and an approximation ratio 2/3. For the streaming setting with random arrival, we get the same approximation ratio with $\mathcal{O}\big(n^{1.5}\big)$ memory. Note that this doesn't fulfil the requirements of the semi-streaming model but it is nevertheless impressive given the hardness results for that setting.

Finally, let us mention another line of work on fixed-parameter[6] maximum matching. Chitnis et al. [18] proposed an elegant distribution which gives random graph that have the same matching size as the original one with high probability. They demonstrated how to sample from it both in the dynamic[7] streaming model and in the classical MPC model. In the streaming model, a space of $\widetilde{\mathcal{O}}(k\log(k))$ is needed where k is the size of a maximum matching of the graph. This result is especially important because in the $\mathcal{SF}$ coreset that we introduce later, we will need the assumption that the input graph has a large matching. We we let their primitive handle the case of small matching.

1.4.1 Goal of this thesis

The goal of this thesis is to further investigate randomized composable coresets for the maximum matching problem, i.e. try to improve the analysis of existing coreset or propose new ones.

[5] In the semi-streaming model, edges arrive in arbitrary order and the memory is limited to $\widetilde{O}(n)$ (this limitation instead of plain $\mathcal{O}(n)$ justifies the *semi* prefix)

[6] By fixed parameter, we mean that the size k of a maximum matching is treated as a parameter. This is especially useful if we know *a priori* that k will be small in which case we can afford a complexity described by a large function of k.

[7] Edge can be both inserted and deleted

Chapter 2

Coresets based on maximum matching

In this chapter, we explore two coresets that are based on picking a maximum matching. They are by nature easy to implement but not very robust against adversarial graphs.

2.1 The maximum matching ($\mathcal{MM}$) coreset

Simply *returning any maximum matching as coreset* was first proposed by Assadi & Khanna in [19]. Their goal was to motivate the use of randomized composable coresets (as opposed to the classical *non-randomized* ones). This was done by showing that the following coreset gives a constant-factor approximation. Formally, the $\mathcal{MM}$ coreset[1] is defined as follows:

```
$\mathcal{MM}$ on input $G$:
    1. **Return** a maximum matching $M$ of $G$.
```

Recall from the introduction that there exists efficient primitives to compute a maximum matching (e.g. Edmond's blossom algorithm), so this coreset is easily implementable in practice. Assadi & Khanna initially proved in [19] that the approximation ratio of $\mathcal{MM}$ is $\Omega(1)$ where the hidden constant was lower-bounded by $1/9$. The analysis was later refined by Assadi et al. in [20] improving the lower bound to $1/3$. In the same article, they also proved an upper bound of $1/2$ on the ratio by coming up with the family[2] of Z-graphs (see Figure 2.1).

One of the goal of this thesis was to further tighten the gap between the lower and upper bound of the approximation ratio. We were able to improve our understanding for the class of *regular* graphs as the following section demonstrates.

[1] $\mathcal{MM}$ is not the original typography, but we use it to match the rest of the thesis.

[2] The graph bears no name in the original paper but we gave it one to refer to it more easily.

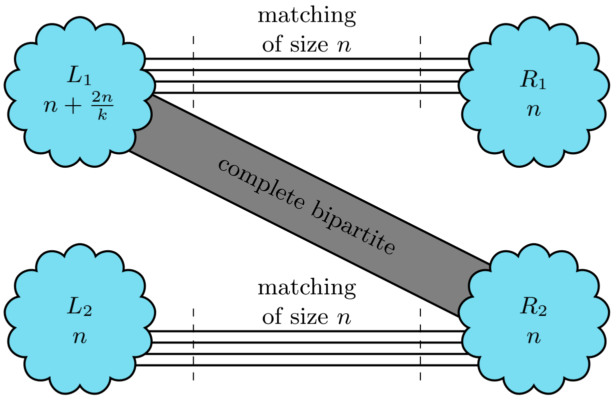

Figure 2.1: The graph $Z(n,k)$ of [20] is bipartite with bipartitions $L_1 \cup L_2$ and $R_1 \cup R_2$ where $|L_1| = n + 2n/k$ and $|L_2| = |R_1| = |R_2| = n$. There is a perfect matching of size n between L_1 and R_1, a complete bipartite graph between L_1 and R_2 and finally a perfect matching of size n between L_2 and R_2.

It is possible to show that with high probability, each $G^{(i)}$ in a random k-partition of $Z(n,k)$ has a maximum matching that contains no edge of $E[L_2, R_2]$. Thus, if somehow each coreset is chosen to be one of those without edges of $E[L_2, R_2]$, then the union of the coresets has a maximum matching of size at most n, i.e. half of $\mathsf{MM}(Z(n,k))$.

2.2 The $\mathcal{MM}$ coreset and regular graphs

The class of regular graphs is known to have good properties with respect to maximum matchings and the results that follow could be common folklore but for sake of completeness, we give detailed arguments. It is also the opportunity to present a new proof strategy that could help reduce the gap in the general case. The results of this section are summarised in the following theorem.

Theorem 1. Let G be some d-regular graph and $G^{(1)}, \dots, G^{(k)}$ a random k-partition. Suppose further that $d \in \omega(k \log(n))$. For any fixed constant $\varepsilon > 0$ and n (the number of vertices of G) large enough, we have:

$$\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{MM}\left(G^{(i)}\right)\right) \geq (\alpha - \mathcal{O}(\varepsilon)) \cdot \mathsf{MM}(G)$$

with probability at least $1 - 1/n$ where α is a constant that we can lower bound depending on the structure of G (see Table 2.1).

Proof. This result follow directly (in order) from Lemmas 7, 8, 9 and 10. Note in particular that those lemmas are stronger than needed. Using a straightforward union bound, we have that *all* of the $G^{(i)}$ have a matching large enough to provide a good approximation on their own. □

Special structure of G	Approximation ratio α
None	1/2
C_3-free	2/3
C_3/C_5-free	3/4
Bipartite	1

Table 2.1: Approximation guarantees for the maximum matching coreset on d-regular graphs

In the above theorem, we assume that the graph G is d-regular with $d \in \omega(k \log(n))$. To justify this, notice that the case $d \in \mathcal{O}(k \log(n))$ is not really interesting *per se*. The random k-partition will be such that each $G^{(i)}$ has size $\widetilde{\mathcal{O}}(n)$ with high probability. In that case, we can afford to return the whole graph as coreset. In practice, this amounts to augmenting $\mathcal{MM}$ with a clause that checks if the input graph is very small and if it is, $\mathcal{MM}$ just returns it in its entirety. Of course, this yields an *exact* coreset for the case $d \in \mathcal{O}(k \log(n))$.

Finally, let us state that the above results can be slightly generalized to graphs that are not perfectly regular but for which the ratio between the maximum and the minimum degree is not too large (e.g. at most two).

2.2.1 Proofs

As this section could be interesting on its own, we unify the notation and keep it for the whole section. Let $G = (V, E)$ be a d-regular graph with $n = |V|$, $k : \mathbb{N} \to \mathbb{N}$ some function and G' a random sub-graph of G where each edge is sampled independently at random with probability

$1/k(n)$. We further assume that $d \in \omega(k \log(n))$.

The common ingredient to all proofs is that under this regime of d, G' is quite likely to look regular. We formalize this intuition in the following lemma.

Lemma 6. [regularity-conservation] For any fixed constant $\varepsilon > 0$ and n large enough:

$$\delta_{G'}(v) \in (1 \pm \varepsilon) \cdot \frac{d}{k} \quad \text{for all } v \in V$$

with probability at least $1 - 1/n$.

Proof. Let $v \in V$ be any vertex. Let $N \subseteq E$ be the set of edges adjacent to v in G (since G is d-regular, we have $|N| = n$). For each $e \in E$, let X_e be a random indicator variable set to one if $e \in E(G')$ and zero else. Note that this collection of random variables is fully independent and that $\delta_{G'}(v) = \sum_{e \in N} X_e$. By linearity of expectation, we have $\mathbb{E}[\delta_{G'}(v)] = d/k$. Using a Chernoff bound (Theorem 10 in appendix), we get that:

$$\Pr\left(\delta_{G'}(v) \notin (1 \pm \varepsilon) \cdot \frac{d}{k}\right) \leq 2e^{-\frac{\varepsilon^2 d}{3k}} = e^{-\omega(\log(n))} = n^{-\omega(1)}$$

Now, we use the union bound to get the probability that there exists a vertex whose degree is out of the desired range:

$$\Pr\left(\exists v \in V : \delta_{G'}(v) \notin (1 \pm \varepsilon) \cdot \frac{d}{k}\right) \leq \sum_{v \in V} \Pr\left(\delta_{G'}(v) \notin (1 \pm \varepsilon) \cdot \frac{d}{k}\right) \leq n \cdot n^{-\omega(1)} \in n^{-\omega(1)}$$

Therefore, with probability at least $1 - 1/n$, all vertex degrees are concentrated around their expectation d/k. □

With this tool in hand, we are ready to prove separately each approximation ratio from Theorem 1. The proof structure will be quite similar between each result whereas the details will increase in difficulty (except for the bipartite case). This is why we recommend reading them in order.

Lemma 7. For any fixed constant $\varepsilon > 0$ and n large enough:

$$\mathsf{MM}(G') \geq \left(\frac{1}{2} - \mathcal{O}(\varepsilon)\right) \cdot \mathsf{MM}(G)$$

with probability at least $1 - 1/n$.

Proof. Fix $M^\star$ a maximum matching of G and put $\mu = |M^\star|$. Let $W \subseteq V$ be all the vertices touched by $M^\star$, i.e. $W = V(M^\star)$. Note that $|W| = 2\mu$ because $M^\star$ is a proper matching. Using Lemma 6, we have that with probability at least $1 - 1/n$:

$$\delta_{G'}(v) \in (1 \pm \varepsilon) \cdot \frac{d}{k} \quad \text{for all } v \in V$$

Let us now condition on that event. Let $M \subseteq E(G')$ be any maximum matching and let $F \subseteq W$ be all the vertices of W that are not touched by M. Let $\delta \in [0, 1]$ be the proportion of vertices of W that are in F, i.e. $|F| = \delta|W|$. We now bound the size of M in two different ways.

For δ small: If $|F|$ is small, then most of W is covered by M and thus M has to be large. More precisely, M covers $(1-\delta)\cdot|W|$ vertices of W. Since $|W|=2\mu$, this makes for $2(1-\delta)\mu$ vertices. As each edge of M can touch at most two such vertices, we have $|M|\geq(1-\delta)\mu$.

For δ large: Any neighbor of $v\in F$ in G' must lie in $V(M)$, else M is not maximum. Under the degree concentration assumption, the number of edges F generates is at least:

$$|F|\cdot(1-\varepsilon)\cdot\frac{d}{k}=2\delta\mu\cdot(1-\varepsilon)\cdot\frac{d}{k}$$

On the other hand, the maximum number of edges coming from F that $V(M)$ can host is at most:

$$2|M|\cdot\left((1+\varepsilon)\cdot\frac{d}{k}-1\right)$$

This is indeed an upper bound as it would imply that each endpoint of M only holds edge from F. There is a minus 1 in the above expression because each $v\in V(M)$ has one edge of its degree taken by M itself. We can safely drop it however, giving only a looser upper-bound. Using the above two observations, we get that the following must hold:

$$2|M|\cdot(1+\varepsilon)\cdot\frac{d}{k}\geq 2\delta\mu\cdot(1-\varepsilon)\cdot\frac{d}{k}\implies|M|\geq\left(\frac{1-\varepsilon}{1+\varepsilon}\right)\cdot\delta\mu\geq(1-2\varepsilon)\cdot\delta\mu$$

Hence, combining the case for δ small and δ large, we find that $|M|\geq\max\{1-\delta,\,(1-2\varepsilon)\cdot\delta\}\cdot\mu$. This guarantees the desired lower bound on $|M|$:

$$|M|\geq\left(\frac{1}{2-2\varepsilon}\right)\cdot\mu\geq\left(\frac{1}{2}-\mathcal{O}(\varepsilon)\right)\cdot\mu$$

□

Another (perhaps simpler) way to prove the above is to follow the arguments of Lemma 10 and use the fact that a maximum matching has at least half the size of a minimum vertex cover for general graphs (see Lemma 3 in the introduction).

Lemma 8. If G is further C_3-free, for any fixed constant $\varepsilon>0$ and n large enough:

$$\mathsf{MM}(G')\geq\left(\frac{2}{3}-\mathcal{O}(\varepsilon)\right)\cdot\mathsf{MM}(G)$$

with probability at least $1-1/n$.

Proof. The proof of this is identical to the one of Lemma 7 except that we improve on the case of δ being large. We reuse exactly the same notation.

For δ large: Any neighbor of $v\in F$ in G' must again be an endpoint of M. This time however, note that for each $e\in M$ at most one endpoint of e can have a neighbor in V.

Suppose toward contradiction that the two endpoints a and b of e have neighbors in F. Let x be such a neighbor of a and y such a neighbor of b. Because G is triangle free (and thus G' too), we have that $x\neq y$. This implies that (x,a,b,y) is a valid augmenting path with respect to M. A contradiction with the fact that M is maximum.

This observation mean that it takes even more of M to handle the edges generated by F. In particular, we need:

$$|M| \cdot \left((1+\varepsilon) \cdot \frac{d}{k} - 1 \right) \geq |F| \cdot (1-\varepsilon) \cdot \frac{d}{k} \implies |M| \geq \left(\frac{1-\varepsilon}{1+\varepsilon} \right) \cdot 2\delta\mu \geq (1-2\varepsilon) \cdot 2\delta\mu$$

Combining the case for δ small (see Lemma 7) and δ large, we have $|M| \geq \max\{1-\delta,\, (1-2\varepsilon) \cdot 2\delta\}\mu$, hence:

$$|M| \geq \left(\frac{2-4\varepsilon}{3-4\varepsilon} \right) \cdot \mu \geq \left(\frac{2}{3} - \mathcal{O}(\varepsilon) \right) \cdot \mu$$

□

The above proof show that actually only half the endpoints of M have neighbor in F. The remaining endpoints also have degree about d/k. Where do those edge go? The following lemma brings an answer.

Lemma 9. If G is further C_3/C_5-free, for any fixed constant $\varepsilon > 0$ and n large enough:

$$\mathsf{MM}(G') \geq \left(\frac{3}{4} - \mathcal{O}(\varepsilon) \right) \cdot \mathsf{MM}(G)$$

with probability at least $1 - 1/n$.

Proof. Let us first condition on the vertices having their degree close to the expected value:

$$\delta_{G'}(v) \in (1 \pm \varepsilon) \cdot \frac{d}{k} \quad \text{for all } v \in V$$

From Lemma 6, this happens with probability $1-1/n$ for any fixed $\varepsilon > 0$. Let us now partition $V(M)$ into three disjoints sets H, J and N. Let $v \in V(M)$ be any vertex and u its unique neighbor in M, v is placed in one of the three set as follows.

$$v \in \begin{cases} H & \text{if } v \text{ has a neighbor in } F \\ J & \text{if } v \text{ has no neighbor in } F \text{ but } u \text{ does} \\ N & \text{if } v \text{ and } u \text{ have no neighbor in } F \end{cases}$$

Observe that this is indeed a partition of $V(M)$. An edge of M either has one end in H and the other in J or two ends in N because M is maximum and G' is triangle-free. This partitioning allow us to give a precise description of the possibles edges in G'. The following table summarises the possibilities and should be read as e.g. "there cannot be an edge between two vertices of J in G'".

Possible?	F	H	J	N	other vertex
other vertex	no(1)	yes	no(1)	yes	no(1)
N	no(2)	yes	yes	yes	
J	no(3)	yes	no(4)		
H	yes	yes			
F	no(1)				

Claims of type 1 are due to the fact that M is supposed maximum in G' and claim 2 holds by definition of N. If claim 3 doesn't hold, then either G' has a triangle or M is not maximum similarly to Lemma 8. Finally, suppose for sake of contradiction that claim 4 doesn't hold. It means that there exists an edge $e = \{u,\, v\}$ between two vertex of J. Let a, respectively b, be the other end of u, respectively v in M. By definition of J, we have $a, b \in H$. This further implies that there exists $x, y \in F$ such that x is a neighbor of a and y a neighbor of b. Because G' is C_5-free, $x \neq y$ and thus $(x,\, a,\, u,\, v,\, b,\, y)$ is an M-augmenting path: a contradiction.

Let us now put $|M| = \alpha\mu$ with the goal of proving that $\alpha \geq 3/4 - \mathcal{O}(\varepsilon)$. As before, we let $\delta \in [0,1]$ be a variable describing the fraction of $V(M^\star)$ that lies in F, i.e. $|F| = 2\delta\mu$. Analogously to Lemma 7, we have $\alpha \geq 1 - \delta$.

Note that any edge of G' with an end in F must have the other in H (see the above table). Also, any edge of G' with an end in J must have the other either in H or in N. The fact that each vertex has about the same degree in G' allow us to devise some inequalities relating the size of F, H and N. To make this precise, we let $\beta \in [0,1]$ be yet another parameter describing the fraction of edge with an end in H that have the other end in F. With this parameter in hand, we can write the following constraints:

$$|F| \cdot (1-\varepsilon) \cdot \frac{d}{k} \leq \beta \cdot |H| \cdot \left((1+\varepsilon) \cdot \frac{d}{k} - 1\right)$$

$$|J| \cdot \left((1-\varepsilon) \cdot \frac{d}{k} - 1\right) \leq |N| \cdot \left((1+\varepsilon) \cdot \frac{d}{k} - 1\right) + (1-\beta) \cdot |H| \cdot \left((1+\varepsilon) \cdot \frac{d}{k} - 1\right)$$

Observe that in the above formulation, an edge is removed from the degree of each $V(M)$ because it corresponds to the one in M. We can get rid of those subtleties using Lemma 28 in the appendix. In particular, the above constraints are fulfilled if the following are:

$$\begin{aligned} |F| &\leq (1+3\varepsilon) \cdot \beta \cdot |H| \\ |J| &\leq (1+3\varepsilon) \cdot \Big(|N| + (1-\beta) \cdot |H|\Big) \end{aligned}$$

At last, we introduce a final variable $\rho \in [0,1]$ that indicates the fraction of edges in M that have an end in H and the other in J (as opposed to having both in N). This allows us to write:

$$|J| = |H| = \rho\alpha\mu \quad \text{and} \quad |N| = 2(1-\rho)\alpha\mu$$

With this, we can re-write the above constraints with:

$$\begin{aligned} 2\delta\mu &\leq (1+3\varepsilon) \cdot \beta\rho\alpha\mu \\ \rho\alpha\mu &\leq (1+3\varepsilon) \cdot \Big(2(1-\rho)\alpha\mu + (1-\beta) \cdot \rho\alpha\mu\Big) \end{aligned}$$

Now that we have collected all those constraints we need to look for the worst possible parameters of β, δ and ρ that will make α the smallest. This is equivalent to solving the following optimization program:

$$\begin{aligned} \text{min.} \quad & \alpha \\ \text{s.t.} \quad & \alpha \geq 1 - \delta \\ & 2\delta \leq (1+3\varepsilon) \cdot \beta\rho\alpha \\ & \rho \leq (1+3\varepsilon) \cdot (2 - \rho - \beta\rho) \\ & \alpha, \beta, \delta, \rho \in [0,1] \end{aligned}$$

The first constraint corresponds to the case of δ small whereas constraints 2 and 3 are the one developed above and correspond to δ large. Lemma 29 of the appendix asserts that the optimal α for this above program is such that $\alpha \geq 3/4 - \mathcal{O}(\varepsilon)$ and we're done. □

In light of the above three results, it is only natural to wonder what kind of guarantees we can get if we know that G has no odd cycle up to order 7, 9 or higher. We bring a first answer by showing that in the extreme case for which G is bipartite (has no odd cycle at all), the approximation ratio is indeed about 1.

Lemma 10. If G is bipartite, for any fixed constant $\varepsilon > 0$ and n large enough:

$$\mathsf{MM}(G') \geq (1 - 2\varepsilon) \cdot \mathsf{MM}(G)$$

with probability at least $1 - 1/n$.

Proof. Note first that since G is bipartite and regular, its two partitions both have $n/2$ vertices and G admits a perfect matching (of size $n/2$). This is due, for instance, to Hall's condition (Lemma 2). We are going to show that G' also has an (almost) perfect matching. Using Lemma 6, G' is almost d/k regular with probability at least $1 - 1/n$:

$$\delta_{G'}(v) \in (1 \pm \varepsilon) \cdot \frac{d}{k} \quad \text{for all } v \in V$$

Let us condition on that and let C be any minimum vertex cover of G'. Note that C can cover at most $|C| \cdot (1+\varepsilon) \cdot d/k$ edges of G', because each edge of C has degree at most $(1+\varepsilon) \cdot d/k$ under our assumption. On the other hand, there is at least $n/2 \cdot (1-\varepsilon) \cdot d/k$ edges in G', therefore we need:

$$|C| \cdot (1+\varepsilon) \cdot \frac{d}{k} \geq \frac{n}{2} \cdot (1-\varepsilon) \cdot \frac{d}{k}$$

This directly implies the following lower bound on $|C|$:

$$|C| \geq \left(\frac{1-\varepsilon}{1+\varepsilon}\right) \cdot \frac{n}{2} \geq (1-2\varepsilon) \cdot \frac{n}{2} = (1-2\varepsilon) \cdot \mathsf{MM}(G)$$

Since G is bipartite we can use König's theorem (Lemma 4) which asserts that $\mathsf{MM}(G') = |C|$ and we're done. □

2.3 The extended maximum matching ($\mathcal{EMM}$) coreset

As the Z-graph illustrates, it turns out that even if the initial partition is uniform, there may still have many possible maximum matchings to choose from and especially matchings that do not go well together. In addition, especially focusing on the Z-graph to identify the shortcomings of the $\mathcal{MM}$ coreset could be a good idea because as last section shows, it seems that $\mathcal{MM}$ already performs quite well when the graph is about regular.

To make $\mathcal{MM}$ more robust, we propose a simple extension. Instead of returning a maximum matching only, we *augment* it by connecting isolated vertices if possible. Formally, let us define the $\mathcal{EMM}$ coreset as follows:

$\mathcal{EMM}$ on input G:

1. compute M a maximum matching of G.
2. connect any isolated vertex of G with M if possible.

A first observation to make is that the $\mathcal{EMM}$ coreset has size $\mathcal{O}(n)$. The coreset is a forest because it starts with a proper matching and then augment it without creating any cycle. The approximation ratio is at least $1/3$ because the coreset is a superset of some maximum matching and the $\mathcal{MM}$ coreset has approx ratio $1/3$ (see [20]). We now show that $\mathcal{EMM}$ outperforms $\mathcal{MM}$ in some instances and especially for the Z-graph, as desired.

2.3.1 Light maximum matchings

Definition 6. Let $G = (V, E)$ be a graph, $M \subseteq E$ a matching and $\beta \in \mathbb{N}$. We say that M is β-light if for any $e = \{u, v\} \in M$:

$$\min\{\delta(u), \delta(v)\} \leq \beta$$

Note in particular that the maximum matching of the Z-graph is 1-light.

Lemma 11. Let $G = (V, E)$ be a graph, $e = \{u, v\} \in E$ an edge and $G' \subseteq G$ any subgraph of G. If u or v has degree 1 in G', then $e \in \mathcal{EMM}(G')$.

Proof. Let M be the maximum matching initially selected by $\mathcal{EMM}$ and let $F \subseteq E(G') \setminus M$ be the edges added by $\mathcal{EMM}$ in the second step. Without loss of generality, suppose that $\delta_{G'}(u) = 1$. If $e \notin M$, this means that u is not covered by M. Since e can connect u to M (and is the only one capable of doing so), we must have $e \in F$. This implies that in any case $e \in \mathcal{EMM}(G')$. □

We are now ready to show that the $\mathcal{EMM}$ coreset fixes some flaws of $\mathcal{MM}$.

Theorem 2. *Let G be a graph. If G admits a maximum matching $M^\star$ which is 1-light, then the $\mathcal{EMM}$ coreset is exact (has approximation ratio 1).*

Proof. Let $\{G^{(i)}\}_{i\in[k]}$ be any k-partition of G (not necessarily random). Fix any edge $e \in M^\star$ and let G' be the subgraph in which e was sent. Because $M^\star$ is 1-light, e has an endpoint of degree one in G'. Using Lemma 11, we conclude that $e \in \mathcal{EMM}(G')$. Therefore:

$$M^\star \subseteq \bigcup_{i\in[k]} \mathcal{EMM}\left(G^{(i)}\right)$$

This directly implies that the approximation ratio is 1. □

We are now going to generalize the above to larger value of β, and especially value of β that depends on k, the size of the random partition. The idea is that $M^\star$ might not be exactly 1-light, but if k is sufficiently large it can make it *look* so.

Theorem 3. *Let G be a graph. If G admits a maximum matching $M^\star$ which is $(\rho \cdot k)$-light for some $\rho \in \mathbb{N}$, then the $\mathcal{EMM}$ coreset has expected approximation ratio $e^{-\rho}$:*

$$\mathbb{E}\left[\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{EMM}\left(G^{(i)}\right)\right)\right] \geq e^{-\rho} \cdot \mathsf{MM}(G)$$

Where the expectation is over the random k-partition $\left\{G^{(i)}\right\}_{i\in[k]}$

Proof. Fix some $e \in M^\star$, call this edge *good* if its lowest degree endpoint is 1 in the sampled graph. We now compute the probability that e is good. Let $e = \{u, v\}$ and without loss of generality, let u be the endpoint with $\delta_G(u) \leq \rho k$ (it exists because $M^\star$ is $(\rho \cdot k)$-light). The probability that u has no neighbors outside of v is p:

$$p = \left(1 - \frac{1}{k}\right)^{\delta(u)-1} \geq \left(1 - \frac{1}{k}\right)^{\rho k} \approx e^{-\frac{\rho k}{k}} = e^{-\rho}$$

Let X be the number of good edges in $M^\star$. By linearity of expectation, we have $\mathbb{E}[X] = p \cdot |M^\star|$. Notice that any good edge is also selected in its coreset via Lemma 11. Therefore:

$$\mathbb{E}\left[\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{EMM}\left(G^{(i)}\right)\right)\right] \geq \mathbb{E}[X] = p \cdot |M^\star| \geq e^{-\rho} \cdot |M^\star|$$

□

It is not trivial to get rid of the expectation and get some "with high probability" kind of statement instead. Indeed, an edge being good depends somewhat on its neighbors in G being good and this is not negatively correlated.

As a last note, observe that the $\mathcal{EMM}$ coreset has an upper-bound of 2/3 on its approximation ratio. This can be seen for instance by taking the ZZ-graph introduced in Section 3.5. Indeed, the matching provided by Lemma 26 is such that no edge of $E[T_2, B_2]$ can be considered for an extension.

Chapter 3

The server flow coreset

In this chapter, we propose a new coreset tailored for bipartite graphs. We will first define our main tool, namely *server flows.* We then move on to proving several new structural results of server flows and show how to use them to guide the computation of a coreset. Finally, we will prove a first lower-bound of 1/2 and an upper bound of 2/3 on the approximation ratio of our coreset.

The concept of server flow was proposed by Bernstein et al. in [21] as a tool to solve bipartite matching in the online setting. The proof structure of Section 3.4 is inspired from [20].

3.1 Introduction & preliminary results

3.1.1 Definitions

Definition 7. [Section 3.1 in [21]] Let $G = (C, S, E)$ be a bipartite graph. We say that $\alpha : S \to \mathbb{R}_{\geq 0}$ is a **server flow** if there exists non-negative numbers $(x_e)_{e \in E}$ such that:

$$\forall c \in C : \sum_{s \in \Gamma(c)} x_{cs} = 1 \quad \text{and} \quad \forall s \in S : \sum_{c \in \Gamma(s)} x_{cs} = \alpha(s)$$

We call $\mathbf{x}$ a **realisation of the server flow** α. The first set of constraint are named **client constraints** while the second set are named **server constraints**.

Intuitively, a realisation is a fractional matching between C and S where each client must be fully matched. The server flow is then a function that tells how much load must be put on each server to hold this fractional matching.

Definition 8. [Section 3.1 in [21]] Let $G = (C, S, E)$ be a bipartite graph and $\alpha : S \to \mathbb{R}_{\geq 0}$ a server flow of G together with a realisation $\mathbf{x}$. For each $v \in V(G)$, we let $A(v)$ be the set of **active neighbors** of v:

$$A(v) = \{u \in \Gamma(v) : x_{uv} > 0\}$$

We say that α is **balanced** if for any client $c \in C$, all of its active neighbors $s \in A(c)$ are such that:

$$\alpha(s) = \min_{s' \in \Gamma(c)} \{\alpha(s')\}$$

The balancedness condition means that clients only put load on their least loaded neighbors. We now recall a result on the existence and uniqueness of balanced server flows.

Lemma 12. [Lemma 14 in [21]] A bipartite graph has a unique balanced server flow if and only if there are no isolated clients.

Their proof showcases an elegant interpretation of the server loads. They construct explicitly a balanced server flow by recursively finding the highest server loads and removing them. They also provide a more straightforward approach using results of convex optimization.

3.1.2 Server flows, maximum matchings and vertex covers

The concept of active neighbors actually yields a natural way to partition the graph G into regions, as we define next.

Definition 9. Let $G = (C, S, E)$ be a bipartite graph and $\alpha : S \to \mathbb{R}_{\geq 0}$ its balanced server flow. Suppose that α takes d different values $\{\alpha_p\}_{p \in [d]}$. For each $p \in [d]$, we define the **region** $R_p = (C_p, S_p, E_p) \subseteq G$ as having all the servers S_p with load exactly α_p, all clients C_p that have active neighbors in S_p and any edge of G with an end in C_p and the other in S_p.

Lemma 13. The regions of G define a partition of the vertex set of G.

Proof. The servers are trivially partitioned according to their server flow value. Now, suppose that a client $c \in C$ happens to belong to region R_p as well as R_q and suppose without loss of generality that $\alpha_p > \alpha_q$. By definition of region, it means that c has active servers in both R_p and R_q, a contradiction with the fact that the server flow that induced the regions is balanced. □

Note that any edge can only appear in at most one region but some edges are removed. More precisely, any edge crossing two regions is absent. See Figure 3.1 for a visual summary of the definitions introduced so far.

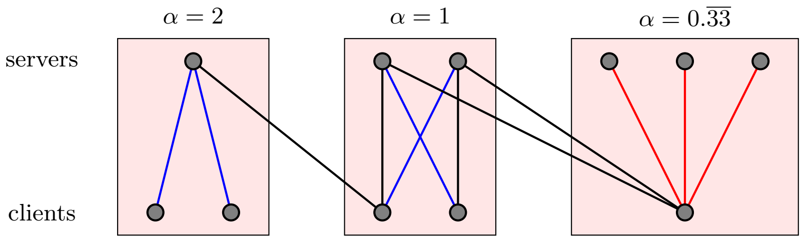

Figure 3.1: A graph together with a balanced server flow. The three induced regions are delimited with rectangles. A possible realisation is to give blue edges a weight of 1, red edges a weight of 1/3 and black edges zero.

Relation with maximum matchings

It turns out that regions behaves well with respect to the maximum matching as this section shows.

Theorem 4. Let $G = (C, S, E)$ be a bipartite graph with no isolated vertices and $\alpha : S \to \mathbb{R}_{\geq 0}$ its balanced server flow together with any realisation $\mathbf{x}$. There exists a maximum matching M of G with $M \subseteq \text{support}(\mathbf{x})$.

We will prove this by constructing M directly. Let $\{R_p\}_{p \in [d]}$ be the regions associated with the d distinct server flow values of α.

Lemma 14. If R_p is a region associated with a server load $\alpha_p \geq 1$, there exists $M_p \subseteq E_p$ that matches all servers $s \in S_p$ and is such that $M_p \subseteq \text{support}(\mathbf{x})$.

Proof. Let $R = (C_p, S_p, E')$ with $E' = E_p \cap \text{support}(\mathbf{x})$. We use Hall's condition (Lemma 2) to show that there exists a matching within R that matches all of S_p. Let us define $(y_e)_{e \in E'}$ with $y_e = x_e/\alpha_p$. For any $s \in S_p$, we have:

$$\sum_{c \in \Gamma_R(s)} y_{cs} = \sum_{c \in \Gamma_R(s)} \frac{x_{cs}}{\alpha_p} = \sum_{c \in \Gamma_G(s)} \frac{x_{cs}}{\alpha_p} = \frac{\alpha_p}{\alpha_p} = 1$$

The second equality comes by definition of R and balancedness of the server flow. Let $A \subseteq S_p$ be any subset of servers. Using the above, we have:

$$|A| = \sum_{s \in A} \sum_{c \in \Gamma_R(s)} y_{cs} = \sum_{c \in \Gamma_R(A)} \sum_{s \in \Gamma_R(c)} y_{cs} = \sum_{c \in \Gamma_R(A)} \sum_{s \in \Gamma_R(c)} \frac{x_{cs}}{\alpha_p} = \sum_{c \in \Gamma_R(A)} \sum_{s \in \Gamma_G(c)} \frac{x_{cs}}{\alpha_p}$$

The last equality comes from the way R is defined, in particular $x_{cs} = 0$ for any $s \notin R$. Using the fact that $\alpha_p \geq 1$ and the client constraints, we get:

$$|A| = \sum_{c \in \Gamma_R(A)} \sum_{s \in \Gamma(c)} \frac{x_{cs}}{\alpha_p} = \sum_{c \in \Gamma_R(A)} \frac{1}{\alpha_p} = \frac{|\Gamma_R(A)|}{\alpha_p} \leq |\Gamma_R(A)|$$

Therefore, by Hall's condition, there exists a matching within R that covers all of S_p. Furthermore, this matching is by construction on the support of $\mathbf{x}$. □

Lemma 15. If R_p is a region associated with a server load $\alpha_p < 1$, there exists $M_p \subseteq E_p$ that matches all clients $c \in C_p$ and is such that $M_p \subseteq \text{support}(\mathbf{x})$.

Proof. Let $R = (C_p, S_p, E')$ with $E' = E_p \cap \text{support}(\mathbf{x})$. Let $B \subseteq C_p$ be any subset, we have:

$$|B| = \sum_{c \in B} 1 = \sum_{c \in B} \sum_{s \in \Gamma_G(c)} x_{cs} = \sum_{c \in B} \sum_{s \in \Gamma_R(c)} x_{cs} = \sum_{s \in \Gamma_R(B)} \sum_{c \in \Gamma_G(s)} x_{cs} = \sum_{s \in \Gamma_R(B)} \alpha_p$$

The last equality comes from the server constraints. Now, recall that $\alpha_p \leq 1$, hence:

$$|B| = \alpha_p |\Gamma_R(B)| \leq |\Gamma_R(B)|$$

Therefore, by Hall's condition (Lemma 2), there exists a matching in R that covers all of C_p. Furthermore, this matching lies by construction on the support of $\mathbf{x}$. □

We now have the tools to prove that the support of a balanced realisation always contains a maximum matching.

Proof of theorem 4. Let us consider the matching $M = \bigcup_{p \in [d]} M_p$ where each M_p is provided either by Lemma 14 or 15 depending on the value of α_p. By construction, this matching only uses the support of $\mathbf{x}$. Suppose now that M is not maximum. This further implies that there exists an augmenting path P in G with respect to M (see Lemma 1). Suppose that it starts in one of the vertices of R_i and ends in one of the vertices of R_j. We split the analysis in three cases.

Case $\alpha_i, \alpha_j \geq 1$: Since all servers are matched by M in R_i and R_j, the augmenting path P starts and ends at a client but then P has an even length and thus cannot be augmenting.

Case $\alpha_i, \alpha_j < 1$: The same logic as in the above case applies here.

Case $\alpha_i \geq 1$ and $\alpha_j < 1$: Since all servers are matched by M for R_i and all clients are matched for R_j, the path starts at some client $c \in C_i$ and ends at some server $s \in S_j$. Let us now say that P is oriented from c to s. Since P is augmenting, edges from servers to clients along P are always part of M. On the other hand, since P goes from R_i to R_j, the augmenting path must use an edge $e = \{u, v\}$ from $u \in R_p$ with $\alpha_p \geq 1$ to $v \in R_q$ with $\alpha_q < 1$. The vertex u cannot be a client because the server flow is assumed to be balanced. This implies that e is from a server to a client. By the observation that P is alternating, it means that $e \in M$. This is impossible by construction of M which uses no edge crossing two regions. □

Note that it is possible for a graph to have a maximum matching which is **not** on the support of $\mathbf{x}$ (see Figure 3.2). The above directly yields an interesting corollary.

Corollary 1. Let $G = (C, S, E)$ be a graph without isolated vertices and $\alpha : S \to \mathbb{R}_{\geq 0}$ a balanced server flow. Suppose that α takes d distinct values and induces regions $R_p = (C_p, S_p, E_p)$ for each $p \in [d]$. We have:

$$\mathsf{MM}(G) = \sum_{p=1}^{d} |S_p| \cdot \mathbf{1}_{[\alpha_p \geq 1]} + |C_p| \cdot \mathbf{1}_{[\alpha_p < 1]}$$

Proof. Note that this is the size of M in theorem 4 and we proved that it is maximum. □

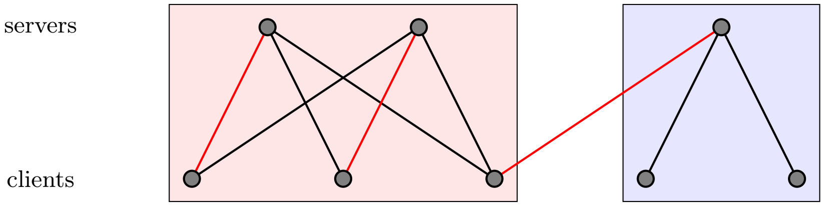

Figure 3.2: A graph together with a maximum matching in red and the balanced server flow decomposition. The red area has server flow 3/2 while the blue one has 2. Note that the matching cannot lie on the support of a realisation of the server flow because any realisation of the crossing edge has value zero.

Canonical vertex covers

We now investigate on the relation between server flows and vertex covers. Once again, the notion of region is crucial and let us derive the existence of a vertex cover with interesting structural properties.

Definition 10. Let $G = (C, S, E)$ be a bipartite graph and $\alpha : S \to \mathbb{R}_{\geq 0}$ its balanced server flow. Let $\{R_p\}_{p\in[d]}$ be the regions associated with the d distinct values of α. For each $p \in [d]$, define $\Phi_p \subseteq V(R_p)$ with:

$$\Phi_p = \begin{cases} S_p & \text{if } \alpha_p \geq 1 \\ C_p & \text{else} \end{cases}$$

We call $\Phi = \bigcup_{p\in[d]} \Phi_p$ the **canonical vertex cover** of G. Its existence and uniqueness is guaranteed by Lemma 14 in [21] (in this thesis, Lemma 12).

Theorem 5. The canonical vertex cover Φ of $G = (C, S, E)$ is a valid minimum vertex cover.

Proof. Observe that each $e \in E$ contained in a region is covered by Φ. Indeed, for each region, Φ is taking either all of the client or all of the servers, hence any edge completely in a region is covered. Now, suppose an edge $e = \{c,\, s\}$ crossing two regions is not covered by Φ. This means that its server s is not in Φ and hence in a region with server flow < 1. On the other hand, its client c is also not in Φ and thus lies in a region with flow ≥ 1. By balancedness such an edge cannot exist.

Now that we know Φ is a valid vertex cover, it remains to argue that it is one of minimum size. Notice that the size of Φ is:

$$|\Phi| = \sum_{p=1}^{d} |S_p| \cdot \mathbf{1}_{[\alpha_p \geq 1]} + |C_p| \cdot \mathbf{1}_{[\alpha_p < 1]}$$

Which is exactly the size of a maximum matching in G by Corollary 1. Since for any graph we have $\mathsf{MM}(G) \leq \mathsf{VC}(G)$ (see Lemma 3), Φ is indeed minimum. □

3.2 The server flow ($\mathcal{SF}$) coreset

The server flow coreset will simply return the support of a balanced realisation. As realisations can have large[1] support, we first argue that there always exists one of small space and show how to construct it. We then move on to give an actual description of the $\mathcal{SF}$ coreset and prove a few preliminary results.

3.2.1 Small space realisations

Theorem 6. Let $G = (C, S, E)$ be a bipartite graph and $\alpha : S \to \mathbb{R}_{\geq 0}$ its balanced server flow. There exists a realisation of α whose support is a forest in G.

We prove this claim by providing an algorithm $\mathcal{F}$ (for *forestify*) that given an initial realisation, reduces its support incrementally without violating the client and server constraints until the realisation is small enough. The algorithm works as follows:

```
$\mathcal{F}$ on input $G = (C, S, E)$, together with an initial valid realisation $(x_e)_{e \in E}$:
  1. let $t = 0$ and $\mathbf{x}^0 = \mathbf{x}$
  2. while there exists a cycle $\Gamma$ in $(C, S, \text{support}(\mathbf{x}^t))$:
     (a) partition the edge of $\Gamma$ in two perfect matchings $A$ and $B$
     (b) compute $\varepsilon$ as follows:
             $\varepsilon = \min_{e \in E(\Gamma)} \{x_e\}$
     (c) suppose $\varepsilon$ is attained for $e \in A$ (if $e \in B$, do something analogous) and let
         $\mathbf{x}^{t+1}$ be defined as:
             $x_e^{t+1} = \begin{cases} x_e^t - \varepsilon & \text{if } e \in A \\ x_e^t + \varepsilon & \text{if } e \in B \\ x_e^t & \text{else} \end{cases}$
     (d) let $t = t + 1$
  3. return $\mathbf{x}^t$
```

Lemma 16. Suppose $\mathcal{F}$ terminates in τ iterations. Then, $\mathbf{x}^\tau$ is a valid realisation of α.

Proof. We prove this by induction on $t \in \{0, \ldots, \tau\}$. Notice that $\mathbf{x}^0$ is by assumption a valid realisation. Suppose now that $\mathbf{x}^t$ is a valid realisation and that there is one more iteration. Let Γ be the cycle found, A and B be the two perfect matchings of Γ (they do exist because Γ is even) and ε be the smallest value, attained for some edge of A.

[1] For instance, in a complete bipartite graph with n and n vertices, a valid balanced realisation is to give every edge the value $1/n$. The support of such a realisation has size $\mathcal{O}(n^2)$!

Non-negativity of $\mathbf{x}^{t+1}$: Fix any edge $e \in A$. By construction, $\varepsilon \leq x_a^t$ and thus, $x_a^{t+1} = x_a^t - \varepsilon \geq 0$. For the other edges, the weight doesn't decrease. Hence $\mathbf{x}^{t+1} \geq \mathbf{0}$.

Client constraints: Let $c \in V(\Gamma)$ be any client on the cycle of Γ. The vertex c has two adjacent edges in Γ: $e \in A$ and $f \in B$. We have:

$$\sum_{s \in N(c)} x_{cs}^{t+1} = \left(\sum_{s \in \Gamma(c) \setminus \{e, f\}} x_{cs}^t \right) + x_e^t - \varepsilon + x_f^t + \varepsilon = \sum_{s \in \Gamma(c)} x_{cs}^t = 1$$

Server contraints: Let $s \in V(\Gamma)$ be any vertex on the cycle of Γ. The vertex s has two adjacent edges in Γ: $e \in A$ and $f \in B$. We have:

$$\sum_{c \in N(s)} x_{cs}^{t+1} = \left(\sum_{c \in \Gamma(s) \setminus \{e, f\}} x_{cs}^t \right) + x_e^t - \varepsilon + x_f^t + \varepsilon = \sum_{c \in \Gamma(s)} x_{cs}^t = \alpha(s)$$

□

Lemma 17. $\mathcal{F}$ runs in time poly-time.

Proof. Notice that each iteration of the algorithm removes at least one edge from the support (the one with minimal weight). Since the graph has $\mathcal{O}(n^2)$ edges, there are at most $\mathcal{O}(n^2)$ iterations. Finally, a cycle or lack thereof can be detected in poly-time using standard DFS. □

We are now all set to prove Theorem 6.

Proof of Theorem 6. Let $\mathbf{x}$ be some realisation of α. Notice that if we feed G and $\mathbf{x}$ to $\mathcal{F}$, the algorithm finishes because of Lemma 17 and the returned realisation contains no cycle on its support by construction. Therefore, the output is a valid realisation (Lemma 16) whose support is a forest. □

This result is encouraging because it means that the size of the $\mathcal{SF}$ coreset is $\mathcal{O}(n)$ (the size of a forest). This is a slight improvement over the $\widetilde{\mathcal{O}}(n)$ size of the EDCS coreset of Assadi et al. [20].

3.2.2 The server flow coreset

We formally describe the **server flow coreset** ($\mathcal{SF}$) as the following algorithm:

$\mathcal{SF}$ on input G:

1. Remove any isolated vertex in G.
2. Compute a server flow α together with a realisation $\mathbf{x}$
3. Modify $\mathbf{x}$ as to have a forest support using $\mathcal{F}$.
4. **Return** the support of $\mathbf{x}$.

Note that the second step amounts to solving a convex quadratic program (see section 3.2.1 in [21]) and can be done in poly-time. The third step is also executable in poly-time because of Lemma 17. As previously noted, this coreset has size[2] $\mathcal{O}(n)$.

Corollary 2. Let α be the approximation ratio of the $\mathcal{MM}$ coreset on the class of graph $\mathcal{G}$ and β be the approximation ratio of the $\mathcal{SF}$ coreset on the same class. Then $\beta \geq \alpha$. In particular, $\mathcal{SF}$ is a 1/3-approximation coreset for the maximum matching problem.

Proof. As seen in theorem 4, the support of $\mathbf{x}$ at the end of the $\mathcal{SF}$ coreset contains a maximum matching of G. Hence, $\mathcal{SF}$ returns at least a maximum matching of G and hence has at least the approximation ratio of the maximum matching coreset. □

[2] Recall that by definition, the size of a coreset is the number of edge.

3.3 The theory of necessary edges

In this section, we develop sufficient conditions for an edge of the graph to *have* to be taken in its respective coreset. We let $G = (C, S, E)$ be a bipartite graph and $M^\star$ a maximum matching of G. To simplify, we analyze this section with a deterministic setting that mimics the one of the server flow coreset, we thus use the following two:

1. G^- is any sub-graph of $G \setminus M^\star$
2. $\widetilde{M}$ is a (potentially empty) subset of $M^\star$

To avoid subtleties, we remove implicitly any isolated client from the graph before asking for a server flow. For convenience, let us put $H = G^- \cup \widetilde{M}$.

Definition 11. $e \in M^\star \setminus \widetilde{M}$ is a $(G^-, \widetilde{M})$**-necessary edge** if its value in any realisation of the balanced server flow of $H + e$ is non-zero.

Lemma 18. Let $e = \{c, s\} \in M^\star \setminus \widetilde{M}$ be some edge. If c or s has degree zero in H, then e is a $(G^-, \widetilde{M})$-necessary edge.

Proof. Let $\mathbf{x}$ be any realisation of the balanced server flow of $H + e$. If $\delta_H(c) = 0$, then by the client constraints we have $x_e = 1$. Else, if $\delta_H(c) > 0$ and $\delta_H(s) = 0$, we have $\alpha_{H+e}(s) = x_{cs}$. Suppose now toward contradiction that $x_{cs} = 0$. It means that $\alpha_{H+e}(s') = 0$ for all $s' \in \Gamma_H(c)$, else α is unbalanced. In particular, we have:

$$\sum_{s' \in \Gamma_{H+e}(c)} x_{cs'} \leq \sum_{s' \in \Gamma_{H+e}(c)} \alpha_{H+e}(s') = 0$$

A contradiction with the client constraints, hence $x_{cs} > 0$ and e is $(G^-, \widetilde{M})$-necessary, as desired. □

Lemma 18 resonates well with the theory developed for light matchings (see the $\mathcal{EMM}$ coreset, section 2.3.1). In Particular, Lemma 11 holds for the $\mathcal{SF}$ coreset as well and the approximation ratio of $\mathcal{SF}$ on the Z-graph is 1.

Lemma 19. Let $e = \{c, s\} \in M^\star \setminus \widetilde{M}$ be some edge. Denote by α_H the balanced server flow of H. If the two following conditions hold, then e must be a $(G^-, \widetilde{M})$-necessary edge:

1. $\alpha_H(s) < 1$
2. $\alpha_H(s') \geq 1$ for all $s' \in \Gamma_H(c)$

Proof. Let α_{H+e} be the the unique balanced server flow of $H + e$ together with any realisation $\mathbf{x}$. Suppose toward contradiction that $x_e = 0$: this implies that $\mathbf{x}$ (without the zero entry for e) is also a valid realisation of α_H and in particular that $\alpha_H = \alpha_{H+e}$. Notice that for all active neighbors $s' \in A_{H+e}(c)$:

$$\alpha_{H+e}(s') = \alpha_H(s') \geq 1$$

Whereas s (which is not in $A_{H+e}(c)$) has:

$$\alpha_{H+e}(s) = \alpha_H(s) < 1$$

A contradiction with the hypothesis that α_{H+e} is balanced. □

Lemma 20. Let $e = \{c, s\} \in M^\star \setminus \widetilde{M}$ be some edge and Φ be the canonical vertex cover of H. If $e \cap \Phi = \emptyset$, then e is a $(G^-, \widetilde{M})$-necessary edge.

Proof. If $e = \{c, s\}$ has an isolated endpoint in H, then it is included in the coreset by Lemma 18. If not, then c is included in the server flow of H. By definition of the canonical vertex cover Φ, we have:

1. $\alpha_H(s) < 1$, because any server not in Φ has load less than 1.
2. $\alpha_H(s') \geq 1$ for all $s' \in \Gamma_H(c)$. Indeed, since $c \notin \Phi$, all of its neighbors must belong to Φ (else Φ is not a valid vertex cover) and hence by definition of Φ, they have load at least 1.

We thus conclude using Lemma 19. □

Note that for the above lemma to work, it is crucial that Φ is the canonical vertex cover. Other covers might not have this property.

Theorem 7. Let $\mu = \mathsf{MM}(G)$, $\mu^- = \mathsf{MM}(G^-)$ and $\nu = |\widetilde{M}|$. There are at least $\mu - \mu^- - 2\nu$ edges in $M^\star \setminus \widetilde{M}$ that are $(G^-, \widetilde{M})$-necessary.

Proof. Let Φ be the canonical vertex cover of H. By Lemma 20, any $e \in M^\star \setminus \widetilde{M}$ is necessary if $e \cap \Phi = \emptyset$. We thus have that the number of necessary edges is at least:

$$|M^\star \setminus \widetilde{M}| - |\Phi| = \mu - \nu - |\Phi|$$

This corresponds to the case in which each vertex of Φ spoils one element of $M^\star \setminus \widetilde{M}$ and this is indeed a worst case as a vertex cannot spoil more than one ($M^\star$ is a proper matching). On the other hand, since G is a bipartite graph and Φ is minimum (see Theorem 5), we have $|\Phi| = \mathsf{MM}(H) \leq \mu^- + \nu$ (recall $H = G' \cup \widetilde{M}$) and the claim follows. □

Let us state that a tighter analysis would yield $\mu - \mu^- - \nu$, but this is not necessary for what follows.

3.4 The $\mathcal{SF}$ coreset provides a $1/2$-approximation

In this section, we finally prove a non-trivial lower bound on the approximation ratio of the $\mathcal{SF}$ coreset. Formally, our claim is the following.

Theorem 8. Let $G = (C, S, E)$ be a bipartite graph with $\mathsf{MM}(G) \in \omega(k \log(n))$ and $\{G^{(i)}\}_{i \in [k]}$ be a random k-partition of G. The server flow coreset $\mathcal{SF}$ is an expected $(1/2 - \varepsilon)$-approximation randomized composable coreset of size $\mathcal{O}(n)$ for the maximum matching problem. More precisely:

$$\mathbb{E}\left[\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{SF}\left(G^{(i)}\right)\right)\right] \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \mathsf{MM}(G)$$

for any fixed $\varepsilon > 0$ and n (the number of vertices in G) large enough.

As discussed in Section 1.4 and motivated by Assadi et al. in [20], the assumption $\mathsf{MM}(G) \in \omega(k \log(n))$ is not a problem since for this regime, we can essentially get an exact coreset of small space using techniques of Chitnis et al. [18].

3.4.1 Proof skeleton

The proof follows somewhat the strategy of [20]. We thus start by recalling one of their main ingredient. We let $M^\star \subseteq E$ be a fixed maximum matching of G of size $\mu = \mathsf{MM}(G)$ and put $G^- = G \setminus M^\star$ and $G_i^- = G^{(i)} \setminus M^\star$ for each $i \in [k]$. The following says that with high probability, all G_i^- have about the same maximum matching size.

Lemma 21. [Claim 3.4 in [20]] Let $\varepsilon \in (0, 1)$ and suppose that $\mu \geq 4\varepsilon^{-2} k \log(n)$. Then, there exists some $\mu^- \in \mathbb{R}$ such that for all $i \in [k]$:

$$\mathsf{MM}\left(G_i^-\right) \in \mu^- \pm \varepsilon\mu$$

with probability $1 - \frac{1}{n}$.

The proof relies on an exposure martingale and the concentration bound is provided by Azuma's inequality. For the remainder of the argument, let us call $\mathcal{A}$ the event of Lemma 21, i.e.

$$\mathcal{A} = \forall i \in [k] : \mathsf{MM}\left(G_i^-\right) \in \mu^- \pm \varepsilon\mu$$

We condition on $\mathcal{A}$ happening (this is with high probability for any fixed $\varepsilon > 0$ for n large enough since we assumed $\mu \in \omega(k \log(n))$). We further split the analysis in two cases, depending on the actual value of μ^-. The most involved will be the second one.

1. If $\mu^- \geq \mu/2$, then any coreset is already good enough on its own.

2. if $\mu^- < \mu/2$, then many edges of $M^\star$ must appear in the union of the coresets.

3.4.2 The case $\mu^- \geq \mu/2$

Lemma 22. Conditioned on $\mathcal{A}$ happening and with $\mu^- \geq \mu/2$, we have:

$$\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{SF}\Big(G^{(i)}\Big)\right) \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \mathsf{MM}(G)$$

Proof. Fix some $i \in [k]$ and let $\widetilde{C} = \mathcal{SF}\big(G^{(i)}\big)$ be the coreset of $G^{(i)}$. We have $G_i^- \subseteq G^{(i)}$ and thus $\mathsf{MM}\big(G^{(i)}\big) \geq \mathsf{MM}\big(G_i^-\big)$. Also, from Theorem 4, we know that $\widetilde{C}$ contains a maximum matching of $G^{(i)}$. This implies that:

$$\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{SF}\Big(G^{(i)}\Big)\right) \geq \mathsf{MM}\Big(\widetilde{C}\Big) = \mathsf{MM}\Big(G^{(i)}\Big) \geq \mu^- - \varepsilon\mu \geq \mu/2 - \varepsilon\mu$$

□

3.4.3 The case $\mu^- < 1/2$

To make the analysis tractable, we will use a trick and compute the probability that a *random* edge of $M^\star$ (where the randomness is independent from the one that generated the random k-partition) is taken in its coreset. Having a random edge rather than a fixed one allow us to leverage results of necessary edges (see section 3.3). We then argue that a good fraction of $M^\star$ must actually be taken.

Random edge analysis

Lemma 23. Let us suppose that $\mathcal{A}$ happens with $\mu^- < \mu/2$, and let $e \in M^\star$ be an edge selected uniformly at random and independently from the random k-partition. Then:

$$\Pr(e \text{ is selected in its coreset}) \geq \frac{1}{2} - 2\varepsilon$$

Proof. Notice crucially that conditioning on $\mathcal{A}$ does not affect how the edges of $M^\star$ are partitioned. We let $\Gamma = \{N \subseteq M^\star : e \notin N\}$ be the set of potential groups of edge in $M^\star$ that could be together with e in its partition. To lighten the notation we denote by $\mathcal{S}$ the event "e is selected in its coreset" and for any $N \in \Gamma$, $\mathcal{T}_N$ the event defined with:

$$\mathcal{T}_N = \text{"The set of edges of } M^\star \text{ that are with } e \text{ in } e\text{'s partition is } N\text{"}$$

Now, we have:

$$\Pr(\mathcal{S}) = \sum_{N\in\Gamma} \Pr(\mathcal{S} \text{ and } \mathcal{T}_N) = \sum_{N\in\Gamma} \Pr(\mathcal{S} \,|\, \mathcal{T}_N) \cdot \Pr(\mathcal{T}_N)$$

Let $\Gamma^+ = \{N \in \Gamma : |N| \leq (1+\delta)\cdot \mu/k\}$ for a constant $\delta > 0$ that we will choose later. We can bound the probability as follows:

$$\begin{aligned}\Pr(\mathcal{S}) &= \sum_{N\in\Gamma} \Pr(\mathcal{S} \mid \mathcal{T}_N)\cdot \Pr(\mathcal{T}_N)\\ &= \sum_{N\in\Gamma^+} \Pr(\mathcal{S} \mid \mathcal{T}_N)\cdot \Pr(\mathcal{T}_N) + \sum_{N\in\Gamma\setminus\Gamma^+} \Pr(\mathcal{S} \mid \mathcal{T}_N)\cdot \Pr(\mathcal{T}_N)\\ &\geq \sum_{N\in\Gamma^+} \Pr(\mathcal{S} \mid \mathcal{T}_N)\cdot \Pr(\mathcal{T}_N)\end{aligned}$$

We now focus on computing $\Pr(\mathcal{S} \mid \mathcal{T}_N)$ with $N \in \Gamma^+$. This means that we have $|N| \leq (1+\delta)\cdot\mu/k$. let G^- be the base sub-graph to which N and e are added. A sufficient condition for e to be part of the final coreset is that it is a (G^-, N)-necessary edge.

Observe that $\mathcal{A}$ only fixes the edges of $G \setminus M^\star$ and that how the edges of $M^\star$ are distributed in the k-partition is independent. Also note that e conditioned on knowing N can be seen as being selected uniformly at random from $M^\star \setminus N$ (see Lemma 30 of the appendix for details).

Now, using Theorem 7, we have that there are at least $\mu - \mathsf{MM}(G^-) - 2|N|$ edges of $M^\star \setminus N$ that are (G^-, N)-necessary. Therefore, we have:

$$\begin{aligned}\Pr(\mathcal{S} \mid \mathcal{T}_N) &\geq \Pr(e \text{ is necessary} \mid \mathcal{T}_N)\\ &\geq \frac{\mu - \mathsf{MM}(G^-) - 2|N|}{\mu - |N|}\\ &\geq \frac{\mu - (\mu/2 + \varepsilon\mu) - 2(1+\delta)\cdot\mu/k}{\mu - |N|}\\ &\geq \frac{\mu - \mu/2 - \varepsilon\mu - 2(1+\delta)\cdot\mu/k}{\mu}\\ &= \frac{1}{2} - \varepsilon - \frac{2(1+\delta)}{k}\end{aligned}$$

Where the third inequality comes from the assumption that $\mathcal{A}$ happens and because $|N| \leq (1+\delta)\cdot\mu/k$ for all $N \in \Gamma^+$. This directly implies that:

$$\Pr(\mathcal{S}) \geq \sum_{N\in\Gamma^+} \Pr(\mathcal{S} \mid \mathcal{T}_N)\cdot \Pr(\mathcal{T}_N) \geq \left(\frac{1}{2} - \varepsilon - \frac{2(1+\delta)}{k}\right)\cdot \underbrace{\sum_{N\in\Gamma^+} \Pr(\mathcal{T}_N)}_{\beta}$$

We now give a lower bound for β. Note that $\beta = \Pr(N \in \Gamma^+)$. The process that creates N is simple, it takes any edge from $M^\star - e$ independently with probability $1/k$. We let $X = |N|$ and for each $f \in M^\star - e$ we let X_f be an indicator random variable set to 1 if $f \in N$ and zero else. We have $X = \sum X_f$ and $\mathbb{E}[X] = (\mu - 1)/k$ and as each indicator variable is independent, we can use a Chernoff bound (Theorem 10 in appendix) to get:

$$\Pr\left(X > (1+\delta)\cdot\frac{(\mu-1)}{k}\right) \leq e^{-\frac{\delta^2(\mu-1)}{3k}}$$

We can thus lower bound β with:

$$\beta = \Pr\left(|N| \leq (1+\delta)\cdot\frac{\mu}{k}\right) \geq \Pr\left(|N| \leq (1+\delta)\cdot\frac{(\mu-1)}{k}\right) \geq 1 - e^{-\frac{\delta^2(\mu-1)}{3k}}$$

Recall that we have $\mu \in \omega(k\log(n))$ and therefore, $\beta \geq 1 - o(1)$ for any fixed $\delta > 0$. If we set $\delta = 1$, we finally get:

$$\Pr(\mathcal{S}) \geq \left(\frac{1}{2} - \varepsilon - \frac{4}{k}\right) \cdot (1 - o(1)) \geq \frac{1}{2} - \varepsilon - \frac{4}{k} - o(1)$$

If we suppose that $k \in \omega(1)$, then we have $\Pr(\mathcal{S}) \geq 1/2 - 2\varepsilon$ for n large enough. □

Now that we know that a random edge is likely to be taken, we argue that a large fraction of $M^\star$ actually gets taken in expectation.

Lemma 24. Conditioned on $\mathcal{A}$ happening and with $\mu^- < \mu/2$, we have:

$$\mathbb{E}\left[\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{SF}\left(G^{(i)}\right)\right)\right] \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \mathsf{MM}(G)$$

Proof. Let e be an edge selected uniformly at random from $M^\star$ and $F \subseteq M^\star$ be the edges of $M^\star$ that are selected in their respective coreset. We have:

$$\begin{aligned}\Pr(e \text{ is selected in its coreset}) &= \sum_{f \in M^\star} \Pr(e \text{ is selected in its coreset} \,|\, e = f) \cdot \Pr(e = f) \\ &= \frac{1}{|M^\star|} \cdot \sum_{f \in M^\star} \Pr(f \in F) \\ &= \frac{\mathbb{E}[|F|]}{|M^\star|}\end{aligned}$$

Conditioned on $\mathcal{A}$ happening, we have $\Pr(e \text{ is selected in its coreset}) \geq \frac{1}{2} - 2\varepsilon$ (see Lemma 23) and therefore:

$$\mathbb{E}[|F|] \geq \frac{|M^\star|}{2} \cdot (1 - 2\varepsilon)$$

Now, we have:

$$\mathbb{E}\left[\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{SF}\left(G^{(i)}\right)\right)\right] \geq \mathbb{E}[\mathsf{MM}(F)] = \mathbb{E}[|F|] \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \mu$$

□

3.4.4 Wrapping it up

Proof of Theorem 8. From theorem 21, we know that $\mathcal{A}$ happens with probability $1 - 1/n$ for any fixed $\varepsilon > 0$. Lemmas 22 and 24 allow us to conclude. □

3.5 A 2/3-approximation upper-bound for the $\mathcal{SF}$ coreset

We are now going to give an upper bound of 2/3 for the approximation ratio of the $\mathcal{SF}$ coreset. Our counter-example graph can be seen as an extension of the Z-graph of [20] and is an original idea of Jakab Tardos. To do so, we will first prove another structural property of server flows.

The following Lemma can be seen as an extension to Lemma 23 of [21], but tailored at servers instead of clients.

Lemma 25. Let $G = (C, S, E)$ be a bipartite graph with no isolated client and such that there exists a matching that covers all of S. Let $\alpha : S \to \mathbb{R}_{\geq 0}$ be the unique balanced server flow. Then, for any server $s \in S$, we have $\alpha(s) \geq 1$.

Proof. Let $\alpha^\star = \min_{s \in S}\{\alpha(s)\}$. We are going to show that $\alpha^\star \geq 1$. In order to do so, let us put $S^\star = \{s \in S : \alpha(s) = \alpha^\star\}$ and $C^\star = \{c \in C : A(c) \cap S^\star \neq \emptyset\}$. Note that since $\alpha^\star$ is the smallest load, no client in $C^\star$ can have an active edge with a server outside of $S^\star$. Finally, there is no edge between $c \in C \setminus C^\star$ and $S^\star$, as this would break balancedness (again, $\alpha^\star$ is minimal).

Given the two previous observations, we have that $\alpha^\star = |C^\star|/|S^\star|$ (all of the load of $C^\star$ is for $S^\star$ and $S^\star$ has no other source of load) and $\Gamma(S^\star) = C^\star$. This further implies that:

$$|S^\star| \cdot \alpha^\star = |C^\star| = |\Gamma(S^\star)|$$

By the converse of Hall's condition (Lemma 2 in the introduction) (which holds since G has a matching of size $|S|$), we have $|S^\star| \leq \Gamma(S^\star)|$ so we must have $\alpha^\star \geq 1$, as desired. □

In comparison, Lemma 23 of [21] states that if there exists a matching that covers each client, then the load is at most one for every server.

3.5.1 The ZZ-graph and a lower bound

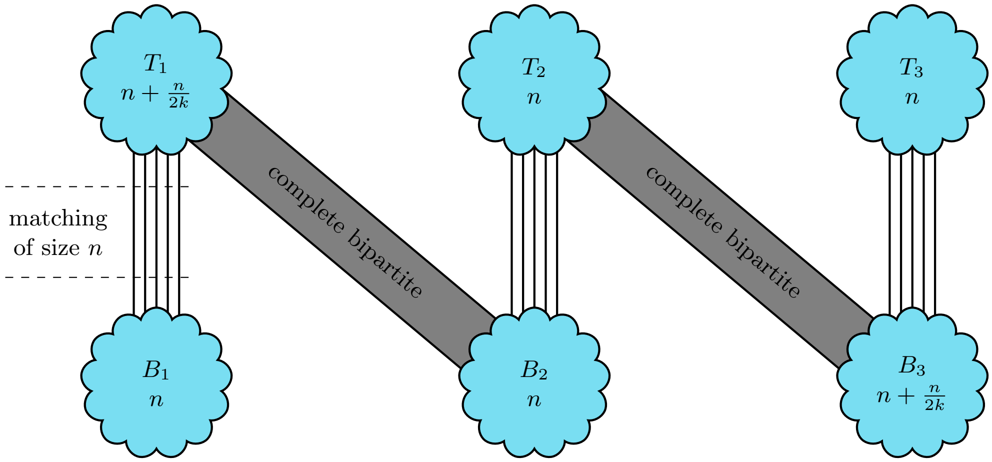

Figure 3.3: Illustration of $ZZ(n, k)$. Note that the perfect matchings are thin compared to the complete bipartite components

Definition 12. Let $n \in \mathbb{N}$ and $k \leq n$. The graph $ZZ(n,k)$ is bipartite with bipartitions $T_1 \cup T_2 \cup T_3$ and $B_1 \cup B_2 \cup B_3$ whose sizes are:

$$|T_1| = |B_3| = n + \frac{2n}{k} \quad \text{and} \quad |B_1| = |B_2| = |T_2| = |T_3| = n$$

$ZZ(n,k)$ contains a perfect matching between T_1 and B_1, T_2 and B_2 and finally T_3 and B_3. Also, it has a complete bipartite graph between T_1 and B_2 and another one between T_2 and B_3. See Figure 3.3 for a visual representation of $ZZ(n,k)$.

We are now going to prove a few properties of the ZZ-graph.

Lemma 26. Let $H(n,k)$ be the $ZZ(n,k)$ graph without the edges between T_2 and B_2. If $H' \subseteq H$ is an edge-sampled subgraph of H (each edge is sampled independently with probability $1/k$). Then H has a matching that matches all the (non-isolated) vertices of B_1, B_2, T_2 and T_3 with probability at least $1 - 1/n^2$ for $k \in o(n/\log(n))$ and n large enough.

Proof. Note that H is actually a graph with two disconnected components. Let us denote by $\mathcal{K}_1$ the one with vertices in T_1, B_1 and B_2 and $\mathcal{K}_2$ the other. If we can prove that $\mathcal{K}_1$ admits w.h.p a matching that covers all of B_1 and B_2, then by symmetry we can apply the same principle to argue that there is one that matches all of T_2 and T_3 within $\mathcal{K}_2$. Indeed, $\mathcal{K}_2$ is simply $\mathcal{K}_1$ but with server and clients flipped. Let us therefore focus on $\mathcal{K}_1$.

Let X be a random variable describing the number of edge between T_1 and B_1 in H'. Using linearity of expectation, we have $\mathbb{E}[X] = n/k$ and using a Chernoff bound (see Theorem 10 in the appendix) we get the following concentration result for X:

$$\Pr\left(X \leq \frac{2n}{k}\right) \leq e^{-\frac{n}{3k}} \leq n^{-\omega(1)}$$

We now condition on $X \leq 2n/k$ (this happens with probability at least $1 - 1/n^2$ for n large enough). Let M_L be the perfect matching between T_1 and non isolated vertices of B_1 and let T_1^- be the vertices of T_1 that are not touched by M_L. Notice that under our assumption, we have $|T_1^-| \geq n$. Since $k \in o(n/\log(n))$, we have $p \in \omega(\log(n)/n)$ and hence can use Lemma 31 (in appendix) that asserts the existence w.h.p of a matching M_R between T_1^- and B_2 that covers all vertices of B_2.

If we let $M_1 = M_L \cup M_R$, we get a matching in $\mathcal{K}_1$ that matches all of the (non-isolated) clients of B_1 and B_2 as desired. Furthermore, this matching exists with probability at least $1 - 1/n^2$. □

Lemma 27. Let G' be an edge-sampled subgraph of $ZZ(n,k)$ (each edge is sampled independently with probability $1/k$). Then, there exists a balanced realisation $\mathbf{y}$ of G' such that:

1. The support of $\mathbf{y}$ contains no cycle.
2. $\text{support}(\mathbf{y}) \cap E[T_2, B_2] = \emptyset$.

with probability at least $1 - 1/n^2$ for $k \in o(n \log(n))$ and n large enough.

Proof. Denote by H' the graph G' without its edges from $E[T_2, B_2]$. Let $\mathcal{A}$ be the event that H' has a matching that covers all the vertices of B_1, B_2, T_2 and T_3. As H' and k fit the setting of Lemma 26, we have that $\mathcal{A}$ happens with probability at least $1-1/n^2$. We now condition on this.

Let $\alpha : S \to \mathbb{R}_{\geq 0}$ be the unique balanced server flow of H'. Using our conditioning, Lemma 25 and Lemma 23 of [21], observe first that we have:

1. $\alpha(s) \leq 1$ for all $s \in T_1$.
2. $\alpha(s) \geq 1$ for all $s \in T_2$.

We are now going to prove that α is not only the balanced server flow of H' but also the one of G'. Let $(x_e)_{e \in E(H')}$ be an arbitrary realisation of α in H'. Define $(y_e)_{e \in E(G')}$ as follows:

$$y_e = \begin{cases} x_e & \text{if } e \in H' \\ 0 & \text{else} \end{cases}$$

Note that the server and client constraints are trivially fulfilled by $\mathbf{y}$. The only thing left to prove is that $\mathbf{y}$ is indeed balanced. The situation of any client not touching anything from $G' \cap E[T_2, B_2]$ remains unchanged (we didn't modify the server flow). So let c be any client of B_2 that has an edge in $E[T_2, B_2]$ and let s be its unique neighbor within T_2. Recall that we have $\alpha(s) \geq 1$ and $\alpha(s') \leq 1$ for all $s' \in \Gamma(c) \setminus s$. Therefore, the active set of c is still valid for balancedness.

In conclusion, $\mathbf{y}$ is a valid realisation of the balanced server flow of G' and by construction it has no edge from $E[T_2, B_2]$ on its support. If we further require that $\mathbf{y}$ doesn't contain any cycle, we can simply ask for the initial $\mathbf{x}$ to have none too. This is certainly possible following the theory developed in Section 3.2. □

3.5.2 Wrapping up

We are now ready to prove the main result of this section.

Theorem 9. For any $n \in \mathbb{N}$ large enough and $k \in o(n/\log(n))$, there exists a graph G for which $\mathcal{SF}$ is such that:

$$\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{SF}\left(G^{(i)}\right)\right) \leq \left(\frac{2}{3} + \frac{\mathcal{O}(1)}{k}\right) \cdot \mathsf{MM}(G)$$

with probability at least $1 - 1/n$ over the random k-partition.

Proof. We take $G = ZZ(n,k)$ and let $G^{(1)}, \ldots, G^{(k)}$ be a random k-partition of G. Using a union bound and lemma 27, we have that with probability $1 - 1/n$, each $G^{(i)}$ admits a balanced realisation $\mathbf{x}^{(i)}$ with no edge of $E[T_2, B_2]$ on its support. Furthermore, those realisations could be the output of $\mathcal{SF}$ as they contain no cycle (i.e., for each $G^{(i)}$, $\mathcal{SF}$ would *unluckily* compute $\mathbf{x}^{(i)}$ on the first go and then skip the cycle detection part because there are none). Let F be the union of all the coreset, i.e.

$$F = \bigcup_{i=1}^{k} \mathcal{SF}\left(G^{(i)}\right) = \bigcup_{i=1}^{k} \text{support}\left(\mathbf{x}^{(i)}\right)$$

Under our adversarial case, we have have $F \cap E[T_2, B_2] = \emptyset$. This means that $\mathsf{MM}(F) \leq 2n + 4n/k$, whereas $\mathsf{MM}(G) = 3n$, further implying:

$$\mathsf{MM}\left(\bigcup_{i=1}^{k} \mathcal{SF}\left(G^{(i)}\right)\right) = \mathsf{MM}(F) \leq 2n + 4n/k = \left(\frac{2}{3} + \frac{\mathcal{O}(1)}{k}\right) \cdot \mathsf{MM}(G)$$

□

Chapter 4

Conclusion

4.1 Summary of results

As as a summary, we provide here a table of all known maximum matching coresets as well as their guarantees and special cases.

coreset	G	lower-bound	upper-bound	reference
EDCS	general	2/3	—	[20]
$\mathcal{MM}$	general	1/3	1/2	[20]
	regular	1/2	—	Section 2.2
	regular and C_3-free	2/3	—	Section 2.2
	regular and C_3/C_5-free	3/4	—	Section 2.2
	regular bipartite	1	—	Section 2.2
$\mathcal{EMM}$	general	1/3	2/3	Section 2.3
	1-light	1	—	Section 2.3
	$(\rho \cdot k)$-light	$e^{-\rho}$	—	Section 2.3
$\mathcal{SF}$	bipartite	1/2	2/3	Chapter 3

Table 4.1: Summary of all maximum matching coresets and their known approximation guarantees.

4.2 Future directions

4.2.1 Maximum matching coreset

For the maximum matching coreset, the 1/3 lower-bound of Assadi et al. [20] still holds for *general* graphs. This leaves us with a few questions:

1. Can we improve this bound for general graph to match the 1/2 worst case of the Z-graph family?

2. Can we improve the ratio for other special classes of graphs? One promising example could be the class of graphs with high girth and the analysis could be done with respect to fractional matchings.
3. We can see the Z-graph as an extension of the path of length 3, a graph know for having the worst ratio between a maximal matching and a maximum one. Could we base a bad example on some other observation, to improve the upper-bound?

For the case of regular graph, can we extend our results for graphs that have larger odd cycle? More precisely, can we say that if G has its smallest odd-length cycle of size $2\ell + 1$, then the $\mathcal{MM}$ coreset has approximation ratio at most $\ell/(\ell + 1)$? An elegant way to provide such a statement would be to prove the following conjecture:

Conjecture 1. Let G be a d-regular graph. If the smallest odd-cycle in G has size $2\ell + 1$, then:

$$\frac{\mathsf{VC}(G)}{\mathsf{MM}(G)} \leq \frac{\ell + 1}{\ell}$$

One way to investigate this would be to study this relationship under the prism of the linear programming relaxation of both problems and strong duality.

On the other hand, can we find other ideas than the one of the $\mathcal{EMM}$ coreset to make the $\mathcal{MM}$ coreset more robust?

4.2.2 Server flow coreset

We believe that the approximation ratio of the $\mathcal{SF}$ is actually 2/3. In order to prove this, it might be necessary to change strategy and instead of looking at $M^\star$ only or not at all (respectively the case $\mu^- \leq \mu/2$ and $\mu^- \geq \mu/2$) try to understand better how coresets interact with each other, much like in the analysis of the EDCS coreset of [20]. This could come from a new concentration result for server flows.

Bibliography

[1] Jack Edmonds. "Paths, Trees, and Flowers". In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467. DOI: 10.4153/cjm-1965-045-4. URL: https://doi.org/10.4153/cjm-1965-045-4.

[2] Béla Bollobás and Alan M. Frieze. "On Matchings and Hamiltonian Cycles in Random Graphs". In: *Random Graphs '83*. Ed. by Michał Karoński and Andrzej Ruciński. Vol. 118. North-Holland Mathematics Studies. North-Holland, 1985, pp. 23–46. DOI: https://doi.org/10.1016/S0304-0208(08)73611-9. URL: http://www.sciencedirect.com/science/article/pii/S0304020808736119.

[3] Douglas B. West. *Introduction to Graph Theory (2nd Edition)*. Pearson, Sept. 2000. ISBN: 0130144002.

[4] Joan Feigenbaum et al. "On graph problems in a semi-streaming model". In: *Theoretical Computer Science* 348.2-3 (Dec. 2005), pp. 207–216. DOI: 10.1016/j.tcs.2005.09.013. URL: https://doi.org/10.1016/j.tcs.2005.09.013.

[5] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. John Wiley & Sons, Inc., July 2008. DOI: 10.1002/9780470277331. URL: https://doi.org/10.1002/9780470277331.

[6] Subhash Khot and Oded Regev. "Vertex cover might be hard to approximate to within $2 - \varepsilon$". In: *Journal of Computer and System Sciences* 74.3 (May 2008), pp. 335–349. DOI: 10.1016/j.jcss.2007.06.019. URL: https://doi.org/10.1016/j.jcss.2007.06.019.

[7] Howard Karloff, Siddharth Suri, and Sergei Vassilvitskii. "A Model of Computation for MapReduce". In: *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Jan. 2010. DOI: 10.1137/1.9781611973075.76. URL: https://doi.org/10.1137/1.9781611973075.76.

[8] Michael T. Goodrich, Nodari Sitchinava, and Qin Zhang. *Sorting, Searching, and Simulation in the MapReduce Framework*. 2011. arXiv: 1101.1902.

[9] Silvio Lattanzi et al. "Filtering". In: *Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures - SPAA '11*. ACM Press, 2011. DOI: 10.1145/1989493.1989505. URL: https://doi.org/10.1145/1989493.1989505.

[10] Michael Kapralov. *Better bounds for matchings in the streaming model*. 2012. eprint: arXiv:1206.2269.

[11] Christian Konrad, Frédéric Magniez, and Claire Mathieu. "Maximum Matching in Semi-streaming with Few Passes". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer Berlin Heidelberg, 2012, pp. 231–242. DOI: 10.1007/978-3-642-32512-0_20. URL: https://doi.org/10.1007/978-3-642-32512-0_20.

[12] Alexandr Andoni et al. *Parallel Algorithms for Geometric Graph Problems.* 2013. arXiv: `1401.0042`.

[13] Paul Beame, Paraschos Koutris, and Dan Suciu. "Communication Steps for Parallel Query Processing". In: *Proceedings of the 32Nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems.* PODS '13. New York, New York, USA: ACM, 2013, pp. 273–284. ISBN: 978-1-4503-2066-5. DOI: `10.1145/2463664.2465224`. URL: `http://doi.acm.org/10.1145/2463664.2465224`.

[14] Andrew McGregor. "Graph stream algorithms". In: *ACM SIGMOD Record* 43.1 (May 2014), pp. 9–20. DOI: `10.1145/2627692.2627694`. URL: `https://doi.org/10.1145/2627692.2627694`.

[15] Alan Frieze and Michal Karonski. *Introduction to Random Graphs.* Cambridge University Press, 2015. DOI: `10.1017/cbo9781316339831`. URL: `https://doi.org/10.1017/cbo9781316339831`.

[16] Vahab Mirrokni and Morteza Zadimoghaddam. *Randomized Composable Core-sets for Distributed Submodular Maximization.* 2015. eprint: `arXiv:1506.06715`.

[17] Sepher Assadi et al. "Maximum matchings in dynamic graph streams and the simultaneous communication model". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM symposium on Discrete Algorithms, SODA 2016.* 2016, pp. 1345–1364.

[18] Rajesh Chitnis et al. "Kernelization via Sampling with Applications to Finding Matchings and Related Problems in Dynamic Graph Streams". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms.* SODA '16. Arlington, Virginia: Society for Industrial and Applied Mathematics, 2016, pp. 1326–1344. ISBN: 9781611974331.

[19] Sepehr Assadi and Sanjeev Khanna. *Randomized Composable Coresets for Matching and Vertex Cover.* 2017. eprint: `arXiv:1705.08242`.

[20] Sepehr Assadi et al. *Coresets Meet EDCS: Algorithms for Matching and Vertex Cover on Massive Graphs.* 2017. eprint: `arXiv:1711.03076`.

[21] Aaron Bernstein, Jacob Holm, and Eva Rotenberg. *Online Bipartite Matching with Amortized $O(\log^2 n)$ Replacements.* 2017. eprint: `arXiv:1707.06063`.

[22] Reinhard Diestel. *Graph Theory.* Springer Berlin Heidelberg, 2017. DOI: `10.1007/978-3-662-53622-3`. URL: `https://doi.org/10.1007/978-3-662-53622-3`.

[23] Artur Czumaj et al. "Round compression for parallel matching algorithms". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing - STOC 2018.* ACM Press, 2018. DOI: `10.1145/3188745.3188764`. URL: `https://doi.org/10.1145/3188745.3188764`.

Chapter 5

Appendix

5.1 Omitted proofs for the $\mathcal{MM}$ coreset and regular graphs

Lemma 28. Let $k, d : \mathbb{N} \to \mathbb{N}$ be two functions such that $d \in \omega(k\log(n))$. Then, for any $\varepsilon \in (0, 1/3)$ and n large enough, we have:

$$\frac{(1+\varepsilon)\cdot\frac{d}{k}-1}{(1-\varepsilon)\cdot\frac{d}{k}-1} \leq (1+3\varepsilon)$$

Proof. Re-arranging and simplifying, we get that the above is equivalent to:

$$\frac{d}{k} \geq 3\varepsilon\cdot\frac{d}{k}+3$$

Which is true, because $d \in \omega(k\log(n))$ and $\varepsilon \in (0, 1/3)$. □

Lemma 29. Consider the following optimization program with four continuous variables, where $\varepsilon > 0$ is an arbitrary non-zero parameter:

$$\begin{aligned}
\text{min.}\quad & \alpha \\
\text{s.t.}\quad & \alpha \geq 1-\delta && \text{(I)}\\
& 2\delta \leq (1+3\varepsilon)\cdot\beta\rho\alpha && \text{(II)}\\
& \rho \leq (1+3\varepsilon)\cdot(2-\rho-\beta\rho) && \text{(III)}\\
& \alpha, \beta, \delta, \rho \in [0,1]
\end{aligned}$$

The optimal value $\alpha^\star$ is such that $\alpha^\star \geq 3/4 - \mathcal{O}(\varepsilon)$.

Proof. For $\delta \leq 1/4$, constraint (I) already implies the result. We can therefore focus on the case $\delta \geq 1/4$ and drop the first constraint. This yields the following program:

$$\begin{aligned}
\text{min.}\quad & \alpha \\
\text{s.t.}\quad & 2\delta \leq (1+3\varepsilon)\cdot\beta\rho\alpha \\
& \rho \leq (1+3\varepsilon)\cdot(2-\rho-\beta\rho) \\
& \alpha, \beta, \rho \in [0,1] \text{ and } \delta \in [1/4,\ 1]
\end{aligned}$$

From the first constraint, we have that:

$$\alpha \geq \frac{2\delta}{(1+3\varepsilon)\cdot\beta\rho}$$

Since $\delta > 0$, this is minimum if the product $\beta\rho$ is maximum. This observation allows us to reduce the optimization program to the following:

$$\begin{aligned} \text{max.} \quad & \beta\rho \\ \text{s.t.} \quad & \rho \leq (1+3\varepsilon)\cdot(2-\rho-\beta\rho) \\ & \beta, \rho \in [0,1] \end{aligned}$$

This program can be seen as the above one where δ is treated as a parameter rather than a variable. Its only constraint can be re-written as:

$$\beta\rho \leq 2 - 2\rho \cdot \left(\frac{1+\frac{3}{2}\varepsilon}{1+3\varepsilon}\right)$$

It is clear that to maximize the product, this constraint should be met with equality. This yields in particular that:

$$\rho = \frac{2}{\beta + 2\cdot\left(\frac{1+\frac{3}{2}\varepsilon}{1+3\varepsilon}\right)}$$

We therefore need to maximize for $\beta \in [0,1]$ the following expression:

$$\frac{2\beta}{\beta + 2\cdot\left(\frac{1+\frac{3}{2}\varepsilon}{1+3\varepsilon}\right)}$$

The function attains its maximum for $\beta^\star = 1$ as it is strictly increasing over $\mathbb{R}$. Note that this makes intuitive sense in the original problem; It says that all of H should be busy with handling F. From $\beta^\star$, we can directly find $\rho^\star$:

$$\rho^\star = \frac{2}{1 + 2\cdot\left(\frac{1+\frac{3}{2}\varepsilon}{1+3\varepsilon}\right)}$$

Using constraint (III) and because $\delta \geq 1/4$, we can derive the following lower bound on α:

$$\alpha^\star \geq \frac{2\delta}{\beta^\star\rho^\star} \geq \frac{1}{2\beta^\star\rho^\star} = \frac{1}{2\rho^\star} = \frac{1+2\cdot\left(\frac{1+\frac{3}{2}\varepsilon}{1+3\varepsilon}\right)}{4} \geq \frac{1+2\cdot(1-2\varepsilon)}{4} = 3/4 - \mathcal{O}(\varepsilon)$$

□

5.2 Omitted proofs for the $\mathcal{SF}$ coreset

Lemma 30. Let $\Omega = \{\omega_1, \dots, \omega_n\}$ be some ground set and $\widetilde{e}$ and $\widetilde{N}$ two random objects drawn as follows:

1. $\widetilde{e}$ is taken uniformly at random from Ω
2. For each $f \in \Omega - \widetilde{e}$, f appears in $\widetilde{N}$ independently with probability $p \in (0, 1)$.

Then, for any $f \in \Omega$ and $M \subseteq \Omega - \widetilde{e}$:

$$\Pr\left(\widetilde{e} = f \mid \widetilde{N} = M\right) = \begin{cases} \frac{1}{n-|M|} & \text{if } f \notin M \\ 0 & \text{else} \end{cases}$$

i.e. conditioned on $\widetilde{N} = M$, the variable $\widetilde{e}$ can be seen as a uniformly random element of $\Omega \setminus M$.

Proof. Using Baye's theorem, we have:

$$\Pr\left(\widetilde{e} = f \mid \widetilde{N} = M\right) = \frac{\Pr\left(\widetilde{N} = M \mid \widetilde{e} = f\right) \cdot \Pr(\widetilde{e} = f)}{\Pr\left(\widetilde{N} = M\right)}$$

We first investigate the value of $\Pr\left(\widetilde{N} = M \mid \widetilde{e} = f\right)$. Notice that if $f \in M$, then this probability is zero, because $f = \widetilde{e}$ and $\widetilde{e} \notin \widetilde{N}$ by definition of the process. Therefore:

$$\Pr\left(\widetilde{N} = M \mid \widetilde{e} = f\right) = \begin{cases} 0 & \text{if } f \in M \\ p^{|M|} \cdot (1-p)^{n-1-|M|} & \text{else} \end{cases}$$

Now, notice that $\Pr(\widetilde{e} = f) = 1/n$. Using these two observations, it is easy to derive $\Pr\left(\widetilde{N} = M\right)$:

$$\begin{aligned} \Pr\left(\widetilde{N} = M\right) &= \sum_{j \in \Omega} \Pr\left(\widetilde{N} = M \mid \widetilde{e} = j\right) \cdot \Pr(\widetilde{e} = j) \\ &= \sum_{j \in \Omega} \Pr\left(\widetilde{N} = M \mid \widetilde{e} = j\right) \cdot \frac{1}{n} \\ &= \frac{1}{n} \sum_{j \in \Omega \setminus M} p^{|M|} \cdot (1-p)^{n-1-|M|} \\ &= \frac{(n - |M|) \cdot p^{|M|} \cdot (1-p)^{n-1-|M|}}{n} \end{aligned}$$

Combining above observations yields the desired result. □

5.3 Some concentration results

5.3.1 Chernoff bounds

Chernoff bounds are essential results for bounding the tails of some distributions. In this thesis, we mainly use the following classical statement.

Theorem 10. Let $Y = \sum_{i=1}^{n} X_i$ the sum of n independent binary random variables each having $\Pr(X_i = 1) = p_i$. Let $\mu_Y = \mathbb{E}[Y] = \sum_{i=1}^{n} p_i$. Then, for any $\varepsilon \in (0, 1)$, we have:

$$\Pr(Y \notin (1 \pm \varepsilon)\mu_Y) \leq 2e^{-\frac{\varepsilon^2 \mu_Y}{3}}$$

A proof can be found in Alon & Spencer [5] Corollary A.1.14.

5.3.2 Perfect matchings in random bipartite graphs

The following says that under suitable assumptions, a random subgraph of the complete bipartite graph contains a maximum matching. The result we mention here is a straightforward extension to the results developped in [2] chapter 4 or in [15] section 6.1.

Lemma 31. Let $G = (A, B, E)$ be a random bipartite graph where $|A| = |B| = n$ and each edge $e \in A \times B$ is present independently with probability $p \in \omega(\log(n)/n)$. Then, for any fixed $\xi \geq 0$ and n large enough:

$$\Pr(G \text{ has a perfect matching}) \geq 1 - 1/n^{\xi}$$