Title
Matrix Balancing in Lp Norms

Author
Yousefi, Arman

Publication Date
2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Matrix Balancing in L_p Norms

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Arman Yousefi

2017

ABSTRACT OF THE DISSERTATION

Matrix Balancing in L_p Norms

by

Arman Yousefi
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2017
Professor Rafail Ostrovsky, Chair

Matrix balancing is a preprocessing step in linear algebra computations such as the computation of eigenvalues of a matrix. Such computations are known to be numerically unstable if the matrix is *unbalanced*, that is the L_2 norm of some rows and their corresponding columns are different by orders of magnitude. Given an unbalanced matrix A, the goal of matrix balancing is to find an invertible diagonal matrix D such that DAD^{-1} is balanced or approximately balanced in the sense that every row and its corresponding column have the same norm. In thesis, we study a classic iterative algorithm for matrix balancing due to Osborne (1960). The original algorithm was proposed for balancing rows and columns in the L_2 norm, and it works by iterating over balancing a row-column pair in fixed round-robin order. Variants of the algorithm for other norms have been heavily studied and are implemented as standard preconditioners in many numerical linear algebra packages. Despite the popularity of Osborne's algorithm in practice and extensive research on it there were no polynomial-time upper bound on the running time of this algorithm to explain the excellent performance of this algorithm in practice. Recently (in 2015), Schulman and Sinclair, in a first result of its kind for any norm, analyzed the rate of convergence of a variant of Osborne's algorithm that uses the L_∞ norm and a different order of choosing row-column pairs. In this

thesis we study matrix balancing in the L_1 norm and other L_p norms. We consider two notions of approximately balancing matrices and refer to them as *ϵ-balancing* and *strict ϵ-balancing*. As the names suggest strict ϵ-balancing implies ϵ-balancing. These notions will be defined in the body of the thesis.

We prove upper bounds on the convergence rate of Osborne's algorithm and some of its variants. We prove fast convergence of different variants of the algorithm to an ϵ-balanced matrix, and propose a variant that converges to a strictly ϵ-balanced matrix in polynomial time. These results resolve a problem that has been open since Osborne proposed his algorithm in 1960. The following is a summary of our results for any real matrix $A = (a_{ij})_{i,j=1}^n$:

1. We propose a simple greedy variant of Osborne's algorithm and show that it converges to an ϵ-balanced matrix in $K = O(\min\{\epsilon^{-2}\log w, \epsilon^{-1}n^{3/2}\log(w/\epsilon)\})$ iterations that cost a total of $O(m + Kn\log n)$ arithmetic operations over $O(n\log(w/\epsilon))$-bit numbers. Here m is the number of non-zero entries of A, and $w = \sum_{i,j}|a_{ij}|/a_{\min}$ with $a_{\min} = \min\{|a_{ij}| : a_{ij} \neq 0\}$.

2. We show that the original round-robin implementation of Osborne's algorithm converges to an ϵ-balanced matrix in $O(\epsilon^{-2}n^2\log w)$ iterations totaling $O(\epsilon^{-2}mn\log w)$ arithmetic operations over $O(n\log(w/\epsilon))$-bit numbers.

3. We devise a random implementation of the iteration and show that it converges to an ϵ-balanced matrix in $O(\epsilon^{-2}\log w)$ iterations using $O(m + \epsilon^{-2}n\log w)$ arithmetic operations over $O(\log(wn/\epsilon))$-bit numbers.

4. We propose a variant of Osborne's algorithm and prove that it converges to a strictly ϵ-balanced matrix in $O\left(\epsilon^{-2}n^9\log(wn/\epsilon)\log w/\log n\right)$ iterations that require $O\left(\epsilon^{-2}n^{10}\log(wn/\epsilon)\log w/\log n\right)$ arithmetic operations over $O(n\log w/\epsilon)$-bit numbers.

5. We demonstrate a lower bound of $\Omega(1/\sqrt{\epsilon})$ on the convergence rate of any implementation of the iteration. Thus, the dependence of our upper bounds on $1/\epsilon$ is in the right ballpark.

All our results are proved for balancing in L_1 norm, but we observe, through a known trivial reduction, that our results for L_1 balancing apply to any L_p norm for all finite p, at the cost of increasing the number of iterations by only a factor of p (or p^2 in some cases).

The dissertation of Arman Yousefi is approved.

Adnan Youssef Darwiche

Igor Pak

Neal E. Young

Rafail Ostrovsky, Committee Chair

University of California, Los Angeles

2017

To my parents

TABLE OF CONTENTS

1 Introduction 1

1.1 Our results 4

1.2 Organization 5

1.3 Preliminaries 5

2 Convergence to an ϵ-balanced matrix **15**

2.1 Greedy Balancing 15

2.2 Round-Robin Balancing (the original algorithm) 20

2.3 Randomized Balancing 25

3 Convergence to a strictly ϵ-balanced matrix **37**

3.1 Preliminaries 39

3.2 Strict Balancing 40

4 A Lower Bound on the Rate of Convergence **47**

References **51**

Acknowledgments

I would like to thank my advisor, Rafail Ostrovsky, for his continual support of me during my studies. Rafail's positive attitude and encouragements were uplifting, and I learnt a lot from his wisdom. Also his flexibility to allow me to work on the problems of my choice gave me a lot of freedom and taught me how to be more independent. I am really grateful to my M.Sc. advisor, Neal Young. I was introduced to theoretical computer science by Neal. Over the years he has been a great mentor to me and many times he has gone beyond his duty as an academic mentor and provided help and friendship that I will never forget. I am also grateful to Adnane Darwiche and Igor Pak, my dissertation committee members, for their support and encouragement.

I would like thank Alexander Sherstov for his mentorship when I first arrived in UCLA. I was inspired by him and his passion for research, and learned from him to invest more in strengthening my foundations in mathematics. I looked up to him for good research practices and learned a lot from him about how to do research. I had the great pleasure to work with my coauthors Yuval Rabani, and Claire Mathieu. I am grateful for their ideas and hard work.

I want to thank my family for their love and support through the ups and downs of graduate school. Finally, I would like to thank God for his faithfulness, blessings, the opportunities that I had, and all the wonderful people that I got to know during this time of my life.

VITA

2004–2007	B.Sc., Computer Engineering, Azad University, Tehran, Iran.
2008–2011	M.Sc., Computer Science, University of California, Riverside, USA.
2012–2017	Ph.D. Candidate, Computer Science, University of California, Los Angeles, USA.

PUBLICATIONS

Arman Yousefi and Neal Young, *On a Linear Program for Minimum-Weight Triangulation,* In ACM-SIAM Symposium on Discrete Algorithms (SODA 2012), 811–823.

Arman Yousefi and Neal Young, *On a Linear Program for Minimum-Weight Triangulation,* In SIAM Journal on Computing, 43 (2014), no. 1, 25–51.

Rafail Ostrovsky, Yuval Rabani and Arman Yousefi, *Matrix Balancing in L_p Norms: Bounding the Convergence Rate of Osborne's Iteration,* In ACM-SIAM Symposium on Discrete Algorithms (SODA 2017), 154–169.

CHAPTER 1

Introduction

Let $A = (a_{ij})_{n\times n}$ be a square matrix with real entries, and let $\|\cdot\|$ be a given norm. For an index $i \in [n]$, let $\|a_{i,.}\|$ and $\|a_{.,i}\|$, respectively, denote the norms of the ith row and the ith column of A, respectively. The matrix A is *balanced* in $\|\cdot\|$ iff $\|a_{.,i}\| = \|a_{i,.}\|$ for all i. An invertible diagonal matrix $D = \mathtt{diag}(d_1, \ldots, d_n)$ is said to *balance* a matrix A iff DAD^{-1} is balanced. A matrix A is *balanceable* in $\|\cdot\|$ iff there exists a diagonal matrix D that balances it.

Osborne [8] studied the above problem in the L_2 norm and considered its application in preconditioning a given matrix in order to increase the accuracy of the computation of its eigenvalues. The motivation is that standard linear algebra algorithms that are used to compute eigenvalues are numerically unstable for unbalanced matrices; diagonal balancing addresses this issue by obtaining a balanced matrix that has the same eigenvalues as the original matrix, as DAD^{-1} and A have the same eigenvalues. Osborne suggested an iterative algorithm for finding a diagonal matrix D that balances a matrix A in the L_2 norm, and also proved that his algorithm converges in the limit. He also observed that if a diagonal matrix $D = \mathtt{diag}(d_1, \ldots, d_n)$ balances a matrix A in the L_2 norm, then the diagonal vector $\mathbf{d} = (d_1, \ldots, d_n)$ minimizes the Frobenius norm of the matrix DAD^{-1}. Osborne's classic algorithm (see Algorithm 1) is an iteration that at each step balances a row and its corresponding column by scaling them appropriately. More specifically the algorithm balances row-column pairs in a fixed cyclic order. In order to balance row and column i, the algorithm scales the ith row by $\sqrt{\|a_{.,i}\|/\|a_{i,.}\|}$ and the ith column by $\sqrt{\|a_{i,.}\|/\|a_{.,i}\|}$. Osborne's algorithm converges to a unique balanced matrix, but there have been no upper bounds on the convergence rate of Osborne's algorithm for the L_2 norm prior to our work.

Parlett and Reinsch [9] generalized Osborne's algorithm to other norms without proving con-

Algorithm 1 Osborne's Balancing Algorithm

Input: Matrix $A \in \mathbb{R}^{n \times n}$, ϵ

Output: An ϵ-balanced matrix

$i = 0$

repeat

$i = i + 1 \mod n$

for $j = 1$ to n **do**

$a_{ij} = \sqrt{\frac{\|a_{.,i}\|}{\|a_{i,.}\|}} \cdot a_{ij}$

$a_{ji} = \sqrt{\frac{\|a_{i,.}\|}{\|a_{.,i}\|}} \cdot a_{ji}$

end for

until Sufficiently balanced

return the resulting matrix

vergence. The L_1 norm version of the algorithm has been studied extensively. The convergence in the limit of the L_1 version was proved by Grad [4], uniqueness of the balanced matrix by Hartfiel [5], and a characterization of balanceable matrices was given by Eaves et al. [3]. Again, there have been no upper bounds on the running time of the L_1 version of the iteration.

The first polynomial time algorithm for balancing a matrix in the L_1 norm was given by Kalantari, Khachiyan, and Shokoufandeh [6]. Their approach is different from the iterative algorithm of Osborne. They reduce the balancing problem to a convex optimization problem and then solve that problem approximately using the ellipsoid algorithm. Their algorithm runs in $O(n^4 \log(\epsilon^{-1} n \log w))$ arithmetic operations where $w = \sum_{i,j} |a_{i,j}|/a_{\min}$ for $a_{\min} = \min\{|a_{ij}| : a_{ij} \neq 0\}$ and ϵ is the relative imbalance of the output matrix (see Definition 1).

For matrix balancing in the L_∞ norm, Schneider and Schneider [11] gave an $O(n^4)$-time non-iterative algorithm. This running time was improved to $O(mn + n^2 \log n)$ by Young, Tarjan, and Orlin [14]. Despite the existence of polynomial time algorithms for balancing in the L_1 and L_∞ norms, and the lack of any theoretical bounds on the running time of the Osborne's iterative algorithm, the latter is favored in practice, and the Parlett and Reinsch variant [9] is implemented as a standard in almost all linear algebra packages (see Chen [2, Section 3.1], also the book [10, Chapter 11] and the code in [1]). One reason is that iterative methods usually perform well in practice and run for far fewer iterations than are needed in the worst case. Another advantage of iterative algorithms is that they are simple, they provide steady partial progress, and they can always generate a matrix that is sufficiently balanced for the subsequent linear algebra computation.

Motivated by the impact of the Osborne's algorithm and the lack of any theoretical bounds on its running time, Schulman and Sinclair [12] recently showed the first bound on the convergence rate of a modified version of this algorithm in the L_∞ norm. They prove that their modified algorithm converges in $O(n^3 \log(\rho n/\epsilon))$ balancing steps where ρ measures the initial imbalance of A and ϵ is the target imbalance of the output matrix. Their algorithm differs from the original algorithm only in the order of choosing row-column pairs to balance (we will use the term *variant* to indicate a deviation from the original round-robin order). Schulman and Sinclair do not prove any bounds on the running time of the algorithm for other L_p norms; this was explicitly mentioned as an open problem. In this work, we resolve this open question, and upper bound the convergence rate of the Osborne's iteration in any L_p norm.

It should be noted that the literature on matrix balancing uses two different definitions for a matrix that is balanced to the relative error of ϵ, with one being stronger than the other. We consider both definitions for balancing a matrix to the relative error of ϵ, and use the terms ϵ-balanced matrix and *strictly* ϵ-balanced matrix. These notions are defined exactly in Definition 1. As the names suggest obtaining an ϵ-balanced matrix is easier than a strictly ϵ-balanced matrix. [1]

We bound the convergence rate of the algorithm and its variants to both an ϵ-balanced matrix and a strictly ϵ-balanced matrix. We first show that Osborne's algorithm and some greedy and randomized variants of it converge to an ϵ-balanced matrix relatively quickly. Then, we propose a variant of Osborne's algorithm and prove that it converges to a strictly ϵ-balanced matrix in polynomial time. Naturally, proving convergence to a strictly ϵ-balanced requires more work and the time to converge to a strictly ϵ-balanced matrix, although polynomial, is much more than the

[1] because a strictly ϵ-balanced matrix is 2ϵ-balanced.

time to converge to an ϵ-balanced matrix.

It is easy to see that Osborne's iteration in L_p norm for any p reduces to balancing in L_1 norm because applying Osborne's iteration to matrix $A = (a_{ij})_{n\times n}$ in any L_p norm is equivalent to applying the iteration to matrix $A = (a_{ij}^p)_{n\times n}$ in L_1 norm. Thus, any time bounds for L_1 norm imply the same bounds with an extra factor of p (or sometimes p^2) for the L_p norm, by using them on the matrix with entries raised to the power of p. For this reason, in thesis we only consider Osborne's iteration in L_1 norm, and specifically show the following bounds. Below, the $\tilde{O}(\cdot)$ notation hides factors that are logarithmic in various parameters of the problem. Exact bounds await the statements of the theorems in the following sections. We also sometimes refer to an iteration of Osborne's algorithm as a *balancing step*.

1.1 Our results

We show that the original algorithm (with no modification) converges to an ϵ-balanced matrix in $\tilde{O}(n^2/\epsilon^2)$ balancing steps, using $\tilde{O}(mn/\epsilon^2)$ arithmetic operations. We also show that a greedy variant converges in $\tilde{O}(1/\epsilon^2)$ balancing steps, using $O(m) + \tilde{O}(n/\epsilon^2)$ arithmetic operations; or alternatively in $\tilde{O}(n^{3/2}/\epsilon)$ iterations, using $\tilde{O}(n^{5/2}/\epsilon)$ arithmetic operations. Thus, the number of arithmetic operations needed by our greedy variant is nearly linear in m or nearly linear in $1/\epsilon$. The near linear dependence on m is significantly better than the Kalantari-Khachiyan-Shokoufandeh algorithm that uses $O(n^4 \log(\epsilon^{-1} n \log w))$ arithmetic operations (and also the Schulman and Sinclair version with a stricter, yet L_∞, guarantee). For an accurate comparison we should note that we may need to maintain $\tilde{O}(n)$ bits of precision, so the running time is actually $O(m + n^2 \log n \log w/\epsilon^2)$ (the Kalantari et al. algorithm maintains $O(\log(wn/\epsilon))$-bit numbers). We improve this with yet another, randomized, variant that has similar convergence rate (nearly linear in m), but needs only $O(\log(wn/\epsilon))$ bits of precision.

For convergence to a strictly ϵ-balanced matrix, we propose a variant of Osborne's iteration and prove that it converges to a strictly ϵ-balanced matrix in $O\left(\epsilon^{-2} n^9 \log(wn/\epsilon) \log w/ \log n\right)$ iterations that require $O\left(\epsilon^{-2} n^{10} \log(wn/\epsilon) \log w/ \log n\right)$ arithmetic operations over $O(n \log(w/\epsilon))$-bit numbers. This polynomial but very large upper bound of $\tilde{O}(n^9)$ on the number of iterations should be viewed in the light of a $O(n^3)$ lower bound. Chen [2] shows that Osborne's algorithm on L_∞ needs at least $O(n^3)$ iterations to converge to a strictly ϵ-balanced matrix. She proves this lower bound for cycles of length n, so this lower bound naturally extends to other L_p norms. However, our work still leaves a huge gap between lower and upper bound of the convergence rate of Osborne's iteration to a strictly ϵ-balanced matrix.

Finally, we show that the dependence on ϵ given by our analyses is within the right ballpark by proving a lower bound of $\Omega(1/\sqrt{\epsilon})$ on the convergence rate of any variant of the algorithm to an ϵ-balanced matrix. This lower bound naturally applies to convergence rate to a strictly ϵ-balanced matrix. Also, notice the contrast with the Schulman-Sinclair upper bound for balancing in the L_∞ norm that has $O(\log(1/\epsilon))$ dependence on ϵ

1.2 Organization

The remainder of this thesis is organized as follows. In the next section we formally define the balancing problem and give definitions and some lemmas that are needed in future chapters. In Chapter 2 we discuss original Osborne's algroritihm and its greedy and randomized variants and prove upper bounds on their rate of convergence to an ϵ-balanced matrix. In Chapter 3 we propose a variant of Osborne's algorithm and bound its convergence rate to a strictly ϵ-balanced matrix. Finally in Chapter 4, we prove our lower bound of $\Omega(1/\sqrt{\epsilon})$ on the convergence rate of any variant of the algorithm to an ϵ-balanced matrix.

1.3 Preliminaries

In this section we introduce notation and definitions, we discuss some previously known facts and results, and we prove a couple of useful lemmas.

The problem. Let $A = (a_{ij})_{n\times n}$ be a square real matrix, and let $\|\cdot\|$ be a norm on $\mathbb{R}^n$. For an index $i \in [n]$, let $\|a_{i,.}\|$ and $\|a_{.,i}\|$, respectively, denote the norms of the ith row and the ith column of A, respectively. A matrix A is *balanced* in $\|\cdot\|$ iff $\|a_{.,i}\| = \|a_{i,.}\|$ for all i. An invertible diagonal matrix $D = \mathtt{diag}(d_1, \ldots, d_n)$ is said to *balance* a matrix A iff DAD^{-1} is balanced. A matrix A is *balanceable* in $\|\cdot\|$ iff there exists a diagonal matrix D that balances it.

For balancing a matrix A in the L_p norm only the absolute values of the entries of A matter, so we may assume without loss of generality that A is non-negative. Furthermore, balancing a matrix does not change its diagonal entries, so if a diagonal matrix D balances A with its diagonal entries replaced by zeroes, then D balances A too. Thus, for the rest of the paper, we assume without loss of generality that the given $n \times n$ matrix $A = (a_{ij})$ is non-negative and its diagonal entries are all 0.

A diagonal matrix $D = \mathtt{diag}(d_1, \ldots, d_n)$ balances $A = (a_{ij})$ in the L_p norm if and only if $D^p = \mathtt{diag}({d_1}^p, \ldots, {d_n}^p)$ balances the matrix $A' = ({a_{ij}}^p)$ in the L_1 norm. Thus, the problem of balancing matrices in the L_p norm (for any finite p) reduces to the problem of balancing matrices in the L_1 norm; for the rest of this thesis we focus on balancing matrices in the L_1 norm.

For an $n \times n$ matrix A, we use $G_A = (V, E, w)$ to denote the weighted directed graph whose adjacency matrix is A. More formally, G_A is defined as follows. Put $V = \{1, \ldots, n\}$, put $E = \{(i, j) : a_{ij} \neq 0\}$, and put $w(i, j) = a_{ij}$ for every $(i, j) \in E$. We use an index $i \in [n]$ to refer to both the ith row or column of A, and to the node i of the digraph G_A. Thus, the non-zero entries of the ith column (the ith row, respectively) correspond to the arcs into (out of, respectively) node i. In the L_1 norm it is useful to think of the weight of an arc as a flow being carried by that arc. Thus, $\|a_{.,i}\|_1$ is the total flow into vertex i and $\|a_{i,.}\|_1$ is the total flow out of it. Note that if a matrix A is not balanced then for some nodes i, $\|a_{.,i}\|_1 \neq \|a_{i,.}\|_1$, and thus the flow on the arcs does not constitute a valid circulation because flow conservation is not maintained. Thus, the goal of balancing in the L_1 norm can be stated as applying diagonal scaling to find a flow function on the arcs of the graph G_A that forms a valid circulation. We use both views of the graph (with arc weights or flow), and also the matrix terminology, throughout this thesis, as convenient.

Without loss of generality we may assume that the undirected graph underlying G_A is connected. Otherwise, after permuting $V = \{1, \ldots, n\}$, the given matrix A can be replaced by $\mathtt{diag}(A_1, \ldots, A_r)$ where each of $A_1, \ldots, A_r$ is a square matrix whose corresponding directed graph is connected. Thus, balancing A is equivalent to balancing each of $A_1, \ldots, A_r$.

The goal of the iterative algorithm is to balance approximately a matrix A, up to an error term ϵ. We define the error here.

Definition 1 (approximate balancing)**.** *Let $\epsilon > 0$.*

1. *A matrix A is* ϵ-balanced *iff*

$$\frac{\sqrt{\sum_{i=1}^{n}(\|a_{.,i}\|_1 - \|a_{i,.}\|_1)^2}}{\sum_{i,j} a_{i,j}} \le \epsilon.$$

2. *A matrix A is* strictly ϵ-balanced *iff for every index i of A (where $i \in [n]$)*

$$\frac{\max\{\|a_{.,i}\|_1, \|a_{i,.}\|_1\}}{\min\{\|a_{.,i}\|_1, \|a_{i,.}\|_1\}} \le 1 + \epsilon.$$

   We also say that every index i that satisfies the above condition is ϵ-balanced.

3. *A diagonal matrix D with positive diagonal entries is said to* (strictly) ϵ-balance A *iff DAD^{-1} is (strictly) ϵ-balanced.*

We use both notions of an ϵ-balanced matrix and a strictly ϵ-balanced matrix. Note that the literature on matrix balancing does not distinguish the two notions and uses ϵ-balanced matrix to refer to both. However, we distinguish between these two mainly because we can show very good convergence rate to an ϵ-balanced matrix with nearly linear rate, but the convergence to a strictly ϵ-balanced matrix is much slower (takes at least $O(n^3)$ as mentioned before) and the analysis is far more complicated. Most of the literature uses the notion of a strictly ϵ-balanced (without using the word "strictly"), and to the best of our knowledge the weaker notion of balancing was first introduced by Kalantari et al. in [6]. However, this weaker notion of an ϵ-balanced matrix is also very useful; its use is justified by the fact that the numerical stability of eigenvalue calculations depends on the Frobenius norm of the balanced matrix (see [9]). As we show in Corollary 2, ϵ-balancing in the L_2 norm approximates the minimum Frobenius norm that can be achieved by balancing.

Our technique Osborne observed that a diagonal matrix $D = \texttt{diag}(d_1, \ldots, d_n)$ that balances a matrix A in the L_2 norm also minimizes the Frobenius norm of the matrix DAD^{-1}. Thus, the balancing problem can be reduced to minimizing a function. Kalantari et al. [6] gave a convex program for balancing in the L_1 norm. Our analysis is based on their convex program. We relate the Osborne's balancing step to the coordinate descent method in convex programming. We show that each step reduces the value of the objective function. Our various bounds are derived through analyzing the progress made in each step. In particular, one of the main tools in our analysis is an upper bound on the distance to optimality (measured by the convex objective function) in terms of the L_1 norm of the gradient, which we prove using network flow arguments.

We prove upper and lower bounds on the convergence rate of the Osborne's balancing algorithm. The Osborne's iterative algorithm balances indices in a fixed round-robin order. Schulman and Sinclair considered a variant that uses a different rule to choose the next index to balance. We consider in this paper several alternative implementations of Osborne's balancing (including the original round-robin implementation) that differ only in the rule by which an index to balance is chosen at each step. For all rules that we consider, the iteration generates a sequence $A = A^{(1)}, A^{(2)}, \ldots, A^{(t)}, \ldots$ of $n \times n$ matrices that converges to a unique balanced matrix A^* (see Grad [4] and Hartfiel [5]). The matrix $A^{(t+1)}$ is obtained by balancing an index of $A^{(t)}$. If the ith index of $A^{(t)}$ is chosen, we get that $A^{(t+1)} = D^{(t)} A^{(t)} {D^{(t)}}^{-1}$ where $D^{(t)}$ is a diagonal matrix with $d_{ii}^{(t)} = \sqrt{\|a_{.,i}^{(t)}\|_1 / \|a_{i,.}^{(t)}\|_1}$ and $d_{jj}^{(t)} = 1$ for $j \neq i$. Note that $a_{i,.}^{(t)}$ ($a_{.,i}^{(t)}$, respectively) denotes the ith row (ith column, respectively) of $A^{(t)}$. Also, putting $\bar{D}^{(1)} = I_{n\times n}$ and $\bar{D}^{(t)} = D^{(t-1)} \cdots D^{(1)}$ for $t > 1$, we get that $A^{(t)} = \bar{D}^{(t)} A (\bar{D}^{(t)})^{-1}$.

The following lemma shows that each balancing step reduces the sum of entries of the matrix[2].

Lemma 1. *Balancing the ith index of a non-negative matrix $B = (b_{ij})_{n\times n}$ (with $b_{ii} = 0$) decreases the total sum of the entries of B by $\left(\sqrt{\|b_{.,i}\|_1} - \sqrt{\|b_{i,.}\|_1}\right)^2$.*

Proof. Before balancing, the total sum of entries in the ith row and in the ith column is $\|b_{i,.}\|_1 + \|b_{.,i}\|_1$. Balancing scales the entries of the ith column by $\sqrt{\|b_{i,.}\|_1 / \|b_{.,i}\|_1}$ and entries of the ith row

[2]This observation was known to Osborne and other researchers that worked on this problem. We prove it here for completeness.

by $\sqrt{\|b_{.,i}\|_1/\|b_{i,.}\|_1}$. Thus, after balancing the sum of entries in the ith column, which equals the sum of entries in the ith row, is equal to $\sqrt{\|b_{i,.}\|_1 \cdot \|b_{.,i}\|_1}$. The entries that are not in the balanced row and column are not changed. Therefore, balancing decreases $\sum_{i,j} b_{ij}$ by

$$\|b_{.,i}\|_1 + \|b_{i,.}\|_1 - 2\sqrt{\|b_{i,.}\|_1 \cdot \|b_{.,i}\|_1} = \left(\sqrt{\|b_{.,i}\|_1} - \sqrt{\|b_{i,.}\|_1}\right)^2 .$$

□

A reduction to convex optimization. Kalantari et al. [6], as part of their algorithm, reduce matrix balancing to a convex optimization problem. We overview their reduction here. Our starting point is Osborne's observation that if a diagonal matrix $D = \mathtt{diag}(d_1, \ldots, d_n)$ balances a matrix A in the L_2 norm, then the diagonal vector $\mathbf{d} = (d_1, \ldots, d_n)$ minimizes the Frobenius norm of the matrix DAD^{-1}. The analogous claim for the L_1 norm is that if a diagonal matrix $D = \mathtt{diag}(d_1, \ldots, d_n)$ balances a matrix A in the L_1 norm, then the diagonal vector $\mathbf{d} = (d_1, \ldots, d_n)$ minimizes the function $F(\mathbf{d}) = \sum_{i,j} a_{ij} \frac{d_i}{d_j}$. On the other hand, Eaves et al. [3] observed that a matrix A can be balanced if and only if the digraph G_A is strongly connected. The following theorem [6, Theorem 1] summarizes the above discussion.

Theorem 1 (Kalantari et al.)**.** *Let $A = (a_{ij})_{n\times n}$ be a real non-negative matrix, $a_{ii} = 0$, for all $i = 1, \ldots n$, such that the undirected graph underlying G_A is connected. Then, the following statements are equivalent.*

(i) A is balanceable in L_1 norm (i.e., there exists a diagonal matrix D such that DAD^{-1} is balanced).

(ii) G_A is strongly connected.

(iii) Let $F(\mathbf{d}) = \sum_{(i,j)\in E} a_{ij} \frac{d_i}{d_j}$. There is a point $\mathbf{d}^* \in \Omega = \{\mathbf{d} \in \mathbb{R}^n : \ d_i > 0, i = 1, \ldots, n\}$ such that $F(\mathbf{d}^*) = \inf\{F(\mathbf{d}) : \ \mathbf{d} \in \Omega\}$.

We refer the reader to [6, Theorem 1] for a proof. We have the following corollary.

Corollary 1. $\mathbf{d}^*$ *minimizes F over Ω if and only if $D^* = \mathtt{diag}(d_1^*, \ldots, d_n^*)$ balances A.*

Proof. As F attains its infimum at $\mathbf{d}^* \in \Omega$, its gradient ∇F satisfies $\nabla F(\mathbf{d}^*) = 0$. Also, $\frac{\partial F(\mathbf{d}^*)}{\partial d_i} = 0$ if and only if $\sum_{j=1}^n a_{ij} \cdot (d_i^*/d_j^*) = \sum_{j=1}^n a_{ji} \cdot (d_j^*/d_i^*)$ for all $i \in [n]$. In other words, $\nabla F(\mathbf{d}^*) = 0$ if and only if the matrix $D^* A D^{*-1}$ is balanced where $D^* = \mathtt{diag}(d_1^*, \ldots, d_n^*)$. Thus, $\mathbf{d}^*$ minimizes F over Ω if and only if $D^* = \mathtt{diag}(d_1^*, \ldots, d_n^*)$ balances A. □

It can also be shown that under the assumption of Theorem 1, the balancing matrix D^* is unique up to a scalar factor (see Osborne [8] and Eaves et al. [3]). Therefore, the problem of balancing matrix A can be reduced to optimizing the function F. Since we are optimizing over the set Ω of strictly positive vectors, we can apply a change of variables $\mathbf{d} = (e^{x_1}, \ldots, e^{x_n}) \in \mathbb{R}^n$ to obtain a convex objective function:

$$f(\mathbf{x}) = f_A(\mathbf{x}) = \sum_{i,j=1}^{n} a_{ij} e^{x_i - x_j}. \tag{1.1}$$

Kalantari et al. [6] use the convex function f because it can be minimized using the ellipsoid algorithm. We do not need the convexity of f, and use f instead of F only because it is more convenient to work with, and it adds some intuition. Notice that the partial derivative of f with respect to x_i is

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = \sum_{j=1}^{n} a_{ij} \cdot e^{x_i - x_j} - \sum_{j=1}^{n} a_{ji} \cdot e^{x_j - x_i}, \tag{1.2}$$

which is precisely the difference between the L_1 norms of the ith row and the ith column of the matrix DAD^{-1}, where $D = \mathtt{diag}(e^{x_1}, \ldots, e^{x_n})$. Also, by definition, the diagonal matrix $\mathtt{diag}(e^{x_1}, \ldots, e^{x_n})$ ϵ-balances A iff

$$\frac{\|\nabla f(\mathbf{x})\|_2}{f(\mathbf{x})} = \frac{\sqrt{\sum_{i=1}^n \left(\sum_{j=1}^n a_{ij} e^{x_i - x_j} - \sum_{j=1}^n a_{ji} e^{x_j - x_i} \right)^2}}{\sum_{i,j=1}^n a_{ij} e^{x_i - x_j}} \leq \epsilon. \tag{1.3}$$

We now state and prove a key lemma that our analysis uses. The lemma uses combinatorial flow and circulation arguments to measure progress by bounding $f(\mathbf{x}) - f(\mathbf{x}^*)$ in terms of $\|\nabla f(\mathbf{x})\|_1$ which is a global measure of imbalances of all vertices.

Lemma 2. *Let f be the function defined in Equation 1.1, and let $\mathbf{x}^*$ be a global minimum of f. Then, for all $\mathbf{x} \in \mathbb{R}^n$, $f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{n}{2} \cdot \|\nabla f(\mathbf{x})\|_1$.*

Proof. Recall that $f(\mathbf{x}) = f_A(\mathbf{x})$ is the sum of entries of a matrix $B = (b_{ij})$ defined by $b_{ij} = a_{ij} \cdot e^{x_i - x_j}$. Notice that $f(\mathbf{x}) = f_B(\vec{\mathbf{0}})$, and $f(\mathbf{x}^*) = f_B(\mathbf{x}^{**})$, where $\mathbf{x}^{**} = \mathbf{x}^* - \mathbf{x}$. Alternatively, $f(\mathbf{x})$ is the sum of flows (or weights) of the arcs of G_B, and $f(\mathbf{x}^*)$ is the sum of flows of the arcs of a graph G^* (an arc ij of G^* carries a flow of $a_{ij} \cdot e^{x_i^* - x_j^*}$). Notice that G_B and G^* have the same set of arcs, but with different weights. By Equation 1.2, $\|\nabla f_A(\mathbf{x})\|_1 = \sum_{i=1}^n \left| \|b_{.,i}\|_1 - \|b_{i,.}\|_1 \right|$, i.e., it is the sum over all the nodes of G_B of the difference between the flow into the node and flow out of it. Also notice that G_B is unbalanced (else the statement of the lemma is trivial), however G^* is balanced. Therefore, the arc flows in G^*, but not those in G_B, form a valid circulation.

Our proof now proceeds in two main steps. In the first step we show a way of reducing the flow on some arcs of G_B, such that the revised flows make every node balanced (and thus form a valid circulation). We also make sure that the total flow reduction is at most $\frac{n}{2} \cdot \|\nabla f_A(\mathbf{x})\|_1$. In the second step we show that sum of revised flows of all the arcs is a lower bound on $f(\mathbf{x}^*)$. These two steps together prove the lemma.

We start with the first step. The nodes of G_B are not balanced. Let S and T be a partition of the unbalanced nodes of G_B, with $S = \{i \in [n] : \ \|b_{.,i}\|_1 > \|b_{i,.}\|_1\}$ and $T = \{i \in [n] : \ \|b_{.,i}\|_1 < \|b_{i,.}\|_1\}$. That is, the flow into a node in S exceeds the flow out of it, and the flow into a node in T is less than the flow out of it. We have that

$$\sum_{i \in S} (\|b_{.,i}\|_1 - \|b_{i,.}\|_1) - \sum_{i \in T} (\|b_{i,.}\|_1 - \|b_{.,i}\|_1)$$
$$= \sum_{i \in [n]} (\|b_{.,i}\|_1 - \|b_{i,.}\|_1) = 0.$$

Thus, we can view each node $i \in S$ as a source with supply $\|b_{.,i}\|_1 - \|b_{i,.}\|_1$, and each node $i \in T$ as a sink with demand $\|b_{i,.}\|_1 - \|b_{.,i}\|_1$, and the total supply equals the total demand. We now add some weighted arcs connecting the nodes in S to the nodes in T. These arcs carry the supply at the nodes in S to the demand at the nodes in T. Note that we may add arcs that are parallel to some existing arcs in G_B. Such arcs can be replaced by adding flow to the parallel existing arcs of G_B. In more detail, to compute the flows of the added arcs (or the added flow to existing arcs), we add arcs inductively as follows. We start with any pair of nodes $i \in S$ and $j \in T$, and add an arc from i to j carrying flow equal to the minimum between the supply at i and the demand at j. Adding this arc will balance one of its endpoints, but in the new graph the sum of supplies at the nodes of S is still equal to the sum of demands at the nodes of T, so we can repeat the process. (Notice that either S or T or both lose one node.) Each additional arc balances at least one unbalanced node, so G_B gets balanced by adding at most n additional arcs from nodes in S to nodes in T. The total flow on the added arcs is exactly $\sum_{i\in S}(\|b_{\cdot,i}\|_1 - \|b_{i,\cdot}\|_1) = \frac{1}{2}\cdot\|\nabla f(\mathbf{x})\|_1$.

Let E' be the set of newly added arcs, and let $G_{B'}$ be the new graph with arc weights given by $B' = (b'_{ij})$. Since $G_{B'}$ is balanced, the arc flows form a valid circulation. We next decompose the total flow of arcs into cycles. Consider a cycle C in $G_{B'}$ that contains at least one arc from E' (i.e., $C \cap E' \neq \emptyset$). Reduce the flow on all arcs in C by $\alpha = \min_{ij\in C} b'_{ij}$. This can be viewed as peeling off from $G_{B'}$ a circulation carrying flow α. This reduces the flow on at least one arc to zero, and the remaining flow on arcs is still a valid circulation, so we can repeat the process. It can be repeated as long as there is positive flow on some arc in E'. Eliminating the flow on all arcs in E' using cycles reduces the total flow on the arcs by at most n times the total initial flow on the arcs in E' (i.e., $\frac{n}{2}\cdot\|\nabla f(\mathbf{x^{(1)}})\|_1$), because each cycle contains at most n arcs and its flow α that is peeled off reduces the flow on at least one arc in E' by α. After peeling off all the flow on all arcs in E', all the arcs with positive flow are original arcs of G_B. Let $G_{B''}$ be the graph with the remaining arcs and their flows which are given by $B'' = (b''_{ij})$. The total flow on the arcs of $G_{B''}$ is at least $f(\mathbf{x}) + \frac{1}{2}\cdot\|\nabla f(\mathbf{x})\|_1 - \frac{n}{2}\cdot\|\nabla f(\mathbf{x})\|_1 \geq f(\mathbf{x}) - \frac{n}{2}\cdot\|\nabla f(\mathbf{x})\|_1$.

Next we show that the total flow on the arcs of $G_{B''}$ is a lower bound on $f(\mathbf{x}^*)$. Our key tool for this is the fact that balancing operations preserve the product of arc flows on any cycle in the original graph G_B, because balancing a node i multiplies the flow on the arcs into i by some factor r and the flow on the arcs out of i by $\frac{1}{r}$. Thus, the geometric mean of the flows of the arcs on any cycle is not changed by a balancing operation. The arc flows in $G_{B''}$ form a valid circulation, and thus can be decomposed into flow cycles $C_1,\ldots,C_q$ by a similar peeling-off process that was described earlier. Let $n_1,\ldots,n_q$ be the lengths of cycles, and let $\alpha_1,\ldots,\alpha_q$ be their flows. The total flow on arcs in $G_{B''}$ is, therefore, $\sum_{k=1}^{q} n_k\cdot\alpha_k$. Notice that, by construction, $b''_{ij}\leq b_{ij}$, and the decomposition into cycles gives that $b''_{ij} = \sum_{k:ij\in C_k}\alpha_k$. Thus,

$$
\begin{aligned}
f(\mathbf{x}^*) &= \sum_{i,j=1}^{n} b_{ij} e^{x_i^{**} - x_j^{**}} \\
&\geq \sum_{i,j=1}^{n} b''_{ij} e^{x_i^{**} - x_j^{**}} \\
&= \sum_{i,j=1}^{n} \sum_{k:ij \in C_k} \alpha_k e^{x_i^{**} - x_j^{**}} \\
&= \sum_{k=1}^{q} \sum_{ij \in C_k} \alpha_k e^{x_i^{**} - x_j^{**}} \\
&\geq \sum_{k=1}^{q} n_k \left(\prod_{ij \in C_k} \alpha_k e^{x_i^{**} - x_j^{**}} \right)^{1/n_k} \\
&= \sum_{k=1}^{q} n_k \alpha_k = \sum_{i,j=1}^{n} b''_{ij}
\end{aligned}
$$

where the last inequality uses the arithmetic-geometric mean inequality. Notice that the right-hand side is the total flow on the arcs of $G_{B''}$, which is at least $f(\mathbf{x}) - \frac{n}{2} \cdot \|\nabla f(\mathbf{x})\|_1$. Thus, $f(\mathbf{x}^*) \geq f(\mathbf{x}) - \frac{n}{2} \cdot \|\nabla f(\mathbf{x})\|_1$, and this completes the proof of the lemma. □

Corollary 2. *Let* $\mathbf{x} = (x_1, \ldots, x_n)$*, let* $\epsilon > 0$*, and let* $\epsilon' = \frac{\epsilon}{n^{3/2}}$*. If* $D = \mathtt{diag}(e^{x_1}, \ldots, e^{x_n})$ *is a diagonal matrix that* ϵ'*-balances* A*, then* $(1 - \epsilon) f(\mathbf{x}) \leq f(\mathbf{x}^*)$*.*

Proof. By Lemma 2, $f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{n}{2} \cdot \|\nabla f(\mathbf{x})\|_1$. Dividing both sides by $f(\mathbf{x})$, we get

$$
1 - \frac{f(\mathbf{x}^*)}{f(\mathbf{x})} \leq \frac{n}{2} \frac{\|\nabla f(\mathbf{x})\|_1}{f(x)} \leq \frac{n^{3/2}}{2} \frac{\|\nabla f(\mathbf{x})\|_2}{f(x)} \leq \frac{n^{3/2}}{2} \epsilon' \leq \epsilon
$$

where the second inequality uses the Cauchy-Schwarz inequality, and the third inequality follows from the definition of ϵ-balance in 1.3. Thus $(1 - \epsilon) f(\mathbf{x}) \leq f(\mathbf{x}^*)$. □

The above corollary justifies the usefulness of the notion of ϵ-balancing because according to [9] the numerical stability of eigenvalue calculations depends on the magnitude of the Frobenius norm of the balanced matrix. In case of L_1 balancing, the Frobenius norm is determined by $f(\mathbf{x})$. Thus, minimizing $f(\mathbf{x})$ amounts to optimizing the numerical stability of eigenvalue computations on the balanced matrix. According to Corollary 2, ϵ'-balancing matrix A approximately minimizes

$f(\mathbf{x})$ to within multiplicative factor of $1 - \epsilon$, and this is good enough for numerical stability of eigenvalue computations.

CHAPTER 2

Convergence to an ϵ-balanced matrix

In this chapter, we study the convergence rate of Osborne's algorithm to an ϵ-balanced matrix. Specifically, we prove an upper bound on the convergence of Osborne's original algorithm to an ϵ-balanced matrix. We also propose a greedy and a randomized variant of Osborne's iteration and show that they quickly converge to an ϵ-balanced matrix. These variants only differ in the order of balancing operations.

This chapter is organized as follows. In section 2.1, we introduce a greedy variant of Osborne's iteration, and bound its convergence rate. In section 2.2, we bound the convergence of Osborne's original algorithm, and finally in section 2.3 we introduce a randomized variant of the algorithm and show that it converges to an ϵ-balanced matrix in nearly linear time in m (i.e. the number of non-zero entries of A).

2.1 Greedy Balancing

Here we present and analyze a greedy variant of Osborne's iteration. Instead of balancing indices in a fixed round-robin order, the greedy modification chooses at iteration t an index i_t of $A^{(t)}$ such that balancing the chosen index results in the largest decrease in the sum of entries of $A^{(t)}$. In other words, we pick i_t such that:

$$i_t = \arg\max_{i \in [n]} \left(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1} \right)^2. \tag{2.1}$$

We give two analyses of this variant, one that shows that the number of balancing operations is nearly linear in the size of G_A, and another that shows that the number of operations is nearly linear in $1/\epsilon$. More specifically, we prove the following theorem.

Theorem 2. *Given an $n \times n$ matrix A, let $m = |E(G_A)|$, the greedy implementation of the Osborne's iterative algorithm outputs an ϵ-balanced matrix in K iterations which cost a total of $O(m + Kn \log n)$ arithmetic operations over $O(n \log(w/\epsilon))$-bit numbers , where* $K = O\left(\min\left\{\epsilon^{-2}\log w, \epsilon^{-1}n^{3/2}\log(w/\epsilon)\right\}\right)$.

The proof uses the convex optimization framework introduced in Section 1.3. Recall that $A^{(t)} = \bar{D}^{(t)}A(\bar{D}^{(t)})^{-1}$. If we let $\bar{D}^{(t)} = \mathtt{diag}(e^{x_1^{(t)}}, \ldots, e^{x_n^{(t)}})$, the iterative sequence can be viewed as generating a sequence of points $\mathbf{x^{(1)}}, \mathbf{x^{(2)}}, \ldots, \mathbf{x^{(t)}}, \ldots$ in $\mathbb{R}^n$, where $\mathbf{x^{(t)}} = (x_1^{(t)}, \ldots, x_n^{(t)})$ and $A^{(t)} = \bar{D}^{(t)}A(\bar{D}^{(t)})^{-1} = (a_{ij}e^{x_i^{(t)} - x_j^{(t)}})_{n\times n}$. Initially, $\mathbf{x^{(1)}} = (0, \ldots, 0)$, and $\mathbf{x^{(t+1)}} = \mathbf{x^{(t)}} + \alpha_t \mathbf{e}_i$, where $\alpha_t = \ln\sqrt{\|a_{.,i}^{(t)}\|_1/\|a_{i,.}^{(t)}\|_1}$ and $\mathbf{e}_i$ is the ith vector of the standard basis for $\mathbb{R}^n$. By Equation 1.1, the value $f(\mathbf{x^{(t)}})$ is sum of the entries of the matrix $A^{(t)}$. The following key lemma allows us to lower bound the decrease in the value of $f(\mathbf{x^{(t)}})$ in terms of a value that can be later related to the stopping condition.

Lemma 3. *If index i_t defined in Equation 2.1 is picked to balance $A^{(t)}$, then $f(\mathbf{x^{(t)}}) - f(\mathbf{x^{(t+1)}})) \geq \frac{\|\nabla f(\mathbf{x^{(t)}})\|_2^2}{4f(\mathbf{x^{(t)}})}$.*

Proof. The value $f(\mathbf{x^{(t)}})$ is the sum of the entries of $A^{(t)}$. By Lemma 1, balancing the i-th index of $A^{(t)}$ reduces the value of $f(\mathbf{x^{(t)}})$ by $\left(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1}\right)^2$. To simplify notation, we drop the superscript t in the following equations. We have

$$\begin{aligned}\left(\sqrt{\|a_{.,i}\|_1} - \sqrt{\|a_{i,.}\|_1}\right)^2 &= \frac{(\|a_{.,i}\|_1 - \|a_{i,.}\|_1)^2}{\left(\sqrt{\|a_{.,i}\|_1} + \sqrt{\|a_{i,.}\|_1}\right)^2} \\ &\geq \frac{(\|a_{.,i}\|_1 - \|a_{i,.}\|_1)^2}{2\left(\|a_{.,i}\|_1 + \|a_{i,.}\|_1\right)}. \end{aligned} \tag{2.2}$$

It is easy to see that

$$\max_{i\in[n]} \frac{(\|a_{.,i}\|_1 - \|a_{i,.}\|_1)^2}{(\|a_{.,i}\|_1 + \|a_{i,.}\|_1)} \geq \frac{\sum_{i=1}^n (\|a_{.,i}\|_1 - \|a_{i,.}\|_1)^2}{\sum_{i=1}^n (\|a_{.,i}\|_1 + \|a_{i,.}\|_1)}. \tag{2.3}$$

But the right hand side of the above inequality (after resuming the use of the superscript t) equals $\frac{\|\nabla f(\mathbf{x^{(t)}})\|_2^2}{2f(\mathbf{x^{(t)}})}$. This is because for all i, $\left(\|a_{i,.}^{(t)}\|_1 - \|a_{.,i}^{(t)}\|_1\right)$ is by Equation 1.2 the i-th coordinate of $\nabla f(\mathbf{x^{(t)}})$, and in the denominator $\sum_{i=1}^{n}\left(\|a_{i,.}^{(t)}\|_1 + \|a_{.,i}^{(t)}\|_1\right) = 2f(\mathbf{x^{(t)}})$. Together with Equations 2.2 and 2.3, this implies that balancing $i_t = \arg\max_{i\in[n]}\left\{\left(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1}\right)^2\right\}$ decreases $f(\mathbf{x^{(t)}})$ by the claimed value. □

Corollary 3. *If matrix $A^{(t)}$ is not ϵ-balanced, by balancing index i_t at iteration t, we have $f(\mathbf{x^{(t)}}) - f(\mathbf{x^{(t+1)}})) > \frac{\epsilon^2}{4}\cdot f(\mathbf{x^{(t)}})$.*

Proof. From Equation 1.3, if $A^{(t)}$ is not ϵ-balanced, $\dfrac{\|\nabla f(\mathbf{x^{(t)}})\|_2}{f(\mathbf{x^{(t)}})} > \epsilon$. Therefore by Lemma 3, $f(\mathbf{x^{(t)}}) - f(\mathbf{x^{(t+1)}})) \geq \frac{\|\nabla f(\mathbf{x^{(t)}})\|_2^2}{4f(\mathbf{x^{(t)}})} = \frac{1}{4}\cdot\left(\frac{\|\nabla f(\mathbf{x^{(t)}})\|_2}{f(\mathbf{x^{(t)}})}\right)^2 \cdot f(\mathbf{x^{(t)}}) > \frac{\epsilon^2}{4}\cdot f(\mathbf{x^{(t)}})$. □

Before we prove Theorem 2, we give a lower bound on $a_{ij}^{(t)}$ for all i, j and t. This lemma is needed to show that finite precision arithmetic using $O(n\log(w/\epsilon))$ bits of precision is sufficient for ϵ-balancing. Recall that $w = \frac{1}{a_{\min}}\cdot\sum_{i,j} a_{i,j}$, where $a_{\min} = \min\{a_{ij} : a_{ij} \neq 0\}$.

Lemma 4. *For every i, j and for every iteration t, $a_{ij}^{(t)} \geq \frac{1}{w^n}\cdot\sum_{ij} a_{ij}$.*

Proof. By Theorem 1 the graph corresponding to A is strongly connected, so (i, j) is contained in at least one directed cycle. Choose a cycle C containing (i, j). Let k be its length. Since the product of weights of arcs in a cycle is preserved by balancing operations,

$$\prod_{(u,v)\in C} a_{uv}^{(t)} = \prod_{(u,v)\in C} a_{uv}.$$

Therefore,

$$a_{ij}^{(t)} = \frac{\displaystyle\prod_{(u,v)\in C} a_{uv}}{\displaystyle\prod_{(u,v)\in C-\{(i,j)\}} a_{uv}^{(t)}} \geq \frac{(a_{\min})^k}{(\sum_{ij} a_{ij})^{k-1}}$$
$$\geq \left(\frac{a_{\min}}{\sum_{ij} a_{ij}}\right)^n \cdot \sum_{ij} a_{ij} = \frac{1}{w^n}\sum_{ij} a_{ij}.$$

The first inequality holds because for every (u, v), $a_{uv}^{(t)} \leq f(\mathbf{x}^{(t)}) \leq f(\mathbf{x}^{(0)}) = \sum_{ij} a_{ij}$. □

Corollary 4. *At any time t, $\frac{1}{w^n}\sum_{ij} a_{ij} \leq f(\mathbf{x^{(t)}}) \leq \sum_{ij} a_{ij}$.*

Proof of Theorem 2. By Corollary 3, while $A^{(t)}$ is not ϵ-balanced, there exists an index i_t to balance such that $f(\mathbf{x^{(t)}}) - f(\mathbf{x^{(t+1)}})) > \frac{\epsilon^2}{4} \cdot f(\mathbf{x^{(t)}})$. Thus, $f(\mathbf{x^{(t+1)}}) < \left(1 - \frac{\epsilon^2}{4}\right) \cdot f(\mathbf{x^{(t)}})$. Iterating for t steps yields $f(\mathbf{x^{(t+1)}}) < \left(1 - \frac{\epsilon^2}{4}\right)^t \cdot f(\mathbf{x^{(1)}})$. So, on the one hand, $f(\mathbf{x^{(1)}}) = \sum_{i,j=1}^n a_{ij}$ since $f(\mathbf{x^{(1)}})$ is the sum of entries in $A^{(1)}$. On the other hand, we argue that the value of $f(\mathbf{x^{(t+1)}})$ is at least $\min_{(i,j)\in E} a_{ij}$. To see this, consider a directed cycle in the graph G_A. Since balancing operations preserve the product of weights of the arcs on any cycle, the weight of at least one arc in the cycle is at least its weight in the input matrix A. Therefore,

$$a_{\min} \leq f(\mathbf{x^{(t+1)}}) < \left(1 - \frac{\epsilon^2}{4}\right)^t \cdot f(\mathbf{x^{(1)}}) = \left(1 - \frac{\epsilon^2}{4}\right)^t \cdot \sum_{i,j=1}^{n} a_{ij}.$$

Thus, $t \leq \frac{4}{\epsilon^2} \cdot \ln w$ and this is an upper bound on the number of balancing operations before an ϵ-balanced matrix is obtained.

The algorithm initially computes $\|a_{.,i}\|_1$ and $\|a_{i,.}\|_1$ for all $i \in [n]$ in $O(m)$ time. Also the algorithm initially computes the value of $\left(\sqrt{\|a_{i,.}\|_1} - \sqrt{\|a_{.,i}\|_1}\right)^2$ for all i in $O(m)$ time and inserts the values in a priority queue in $O(n \log n)$ time. The values of $\|a^{(t)}_{i,.}\|_1$, $\|a^{(t)}_{.,i}\|_1$ for all i and $\left(\sqrt{\|a^{(t)}_{i,.}\|_1} - \sqrt{\|a^{(t)}_{.,i}\|_1}\right)^2$ are updated after each balancing operation. In each iteration the weights of at most $2n$ arcs change. Updating the values of $\|a^{(t)}_{i,.}\|_1$ and $\|a^{(t)}_{.,i}\|_1$ takes $O(n)$ time and updating the values of $\left(\sqrt{\|a^{(t)}_{i,.}\|_1} - \sqrt{\|a^{(t)}_{.,i}\|_1}\right)^2$ involves at most n updates of values in the priority queue, each taking time $O(\log n)$. Thus, the first iteration takes $O(m)$ operations and each iteration after that takes $O(n \log n)$ operations, so the total number of arithmetic operations performed by the algorithm is $O(m + (n \log n \log w)/\epsilon^2)$.

An alternative analysis completes the proof. Notice that $\|\nabla f(\mathbf{x^{(t)}})\|_2 \leq \|\nabla f(\mathbf{x^{(t)}})\|_1 \leq \sqrt{n} \cdot \|\nabla f(\mathbf{x^{(t)}})\|_2$. Therefore,

$$\begin{aligned} f(\mathbf{x^{(t)}}) - f(\mathbf{x^{(t+1)}})) &\geq \frac{\|\nabla f(\mathbf{x^{(t)}})\|_2^2}{4 f(\mathbf{x^{(t)}})} \geq \frac{\|\nabla f(\mathbf{x^{(t)}})\|_2}{4\sqrt{n} \cdot f(\mathbf{x^{(t)}})} \cdot \|\nabla f(\mathbf{x^{(t)}})\|_1 \\ &\geq \frac{1}{2n^{3/2}} \cdot \frac{\|\nabla f(\mathbf{x^{(t)}})\|_2}{f(\mathbf{x^{(t)}})} \cdot (f(\mathbf{x^{(t)}}) - f(\mathbf{x}^*)) \end{aligned}$$

where the first inequality follows from Lemma 3, and the last inequality follows from Lemma 2.

Therefore, while A^t is not ϵ-balanced (so $\frac{\|\nabla f(\mathbf{x^{(t)}})\|_2}{f(\mathbf{x^{(t)}})} > \epsilon$), we have that

$$f(\mathbf{x^{(t)}}) - f(\mathbf{x^{(t+1)}}) \geq \frac{\epsilon}{2n^{3/2}} \cdot (f(\mathbf{x^{(t)}}) - f(\mathbf{x}^*))$$

Rearranging the terms, we get $f(\mathbf{x^{(t+1)}}) - f(\mathbf{x}^*) \leq \left(1 - \frac{\epsilon}{2n^{3/2}}\right) \cdot (f(\mathbf{x^{(t)}}) - f(\mathbf{x}^*))$. Therefore, $f(\mathbf{x^{(t+1)}}) - f(\mathbf{x}^*) \leq \left(1 - \frac{\epsilon}{2n^{3/2}}\right)^t \cdot (f(\mathbf{x^{(1)}}) - f(\mathbf{x}^*))$. Notice that by Lemma 3,

$$\begin{aligned} f(\mathbf{x^{(t+1)}}) - f(\mathbf{x}^*) \geq f(\mathbf{x^{(t+1)}}) - f(\mathbf{x^{(t+2)}}) &\geq \left(\frac{\|\nabla f(\mathbf{x^{(t+1)}})\|_2}{2f(\mathbf{x^{(t+1)}})}\right)^2 \cdot f(\mathbf{x^{(t+1)}}) \\ &\geq \left(\frac{\|\nabla f(\mathbf{x^{(t+1)}})\|_2}{2f(\mathbf{x^{(t+1)}})}\right)^2 \cdot a_{\min} \end{aligned}$$

On the other hand, $f(\mathbf{x^{(1)}}) - f(\mathbf{x}^*) \leq f(\mathbf{x^{(1)}}) \leq \sum_{i,j=1}^n a_{ij}$. Thus, for $t = 2\epsilon^{-1} \cdot n^{3/2} \ln(4w/\epsilon^2)$, we have that $\frac{\|\nabla f(\mathbf{x^{(t+1)}})\|_2}{f(\mathbf{x^{(t+1)}})} \leq \epsilon$, so the matrix is ϵ-balanced.

Thus far, assuming exact arithmetics with infinite precision we have shown that the greedy implementation of Osborne's algorithm converges in the claimed number of arithmetic operations. It remains to show that the algorithm still works if all numbers are represented with only $O(n \log(w/\epsilon))$-bits. For every i, j and t let $\widehat{a}_{ij}^{(t)}$ represent $a_{ij}^{(t)}$ truncated to $O(n \log(w/\epsilon))$ bits of precision. We set the hidden constant so that the truncation error is $r < (\epsilon/w)^{2n}$. By Lemma 4, this error is negligible compared to $a_{ij}^{(t)}$. Specifically,

$$(1 - (\frac{\epsilon}{w})^n) a_{ij}^{(t)} \leq a_{ij}^{(t)} - r \leq \widehat{a}_{ij}^{(t)} \leq a_{ij}^{(t)},$$

where the first inequality uses Lemma 4 and assumes without loss of generality that A is scaled so that $\sum_{i,j} a_{i,j} \geq 1$. For every i and j, the algorithm maintains the value of $\widehat{a}_{ij}^{(t)}$ instead of $a_{ij}^{(t)}$. To balance the i-th index at time t, the algorithm computes $\sqrt{\|\widehat{a}_{.,i}^{(t)}\|_1 / \|\widehat{a}_{i,.}^{(t)}\|_1}$ truncated to $O(n \log(w/\epsilon))$ bits of precision and saves the value in a variable $\widehat{d}_{ii}^{(t)}$. Then it multiplies row i by $\widehat{d}_{ii}^{(t)}$ and divides column i by $\widehat{d}_{ii}^{(t)}$. Thus, for all j, $a_{ij}^{(t+1)} = a_{ij}^{(t)} \cdot \widehat{d}_{ii}^{(t)}$ and $a_{ji}^{(t+1)} = a_{ji}^{(t)} / \widehat{d}_{ii}^{(t)}$. The algorithm computes $a_{ij}^{(t+1)}$ truncated to $O(n \log(w/\epsilon))$ bits of precision and saves this value in $\widehat{a}_{ij}^{(t+1)}$.

Thus, balancing index i reduces the value of function f by

$$
\begin{aligned}
& f(\mathbf{x}^{(t)}) - f(\mathbf{x}^{(t+1)}) = \\
& \left(\|a_{.,i}^{(t)}\|_1 + \|a_{i,.}^{(t)}\|_1\right) - \left(\|a_{.,i}^{(t+1)}\|_1 + \|a_{i,.}^{(t+1)}\|_1\right) \\
& = \left(\|a_{.,i}^{(t)}\|_1 + \|a_{i,.}^{(t)}\|_1\right) - \left(\frac{\|a_{.,i}^{(t)}\|_1}{\widehat{d}_{ii}^{(t)}} + \|a_{i,.}^{(t)}\|_1 \cdot \widehat{d}_{ii}^{(t)}\right) \\
& \geq \left(\|a_{.,i}^{(t)}\|_1 + \|a_{i,.}^{(t)}\|_1\right) - 2(1 + O((\frac{\epsilon}{w})^n))\sqrt{\|a_{.,i}^{(t)}\|_1 \|a_{i,.}^{(t)}\|_1} \\
& = \left(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1}\right)^2 - O((\frac{\epsilon}{w})^n)\sqrt{\|a_{.,i}^{(t)}\|_1 \|a_{i,.}^{(t)}\|_1} \\
& = (1 - o(1))\left(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1}\right)^2 .
\end{aligned}
$$

In the last equation we used the fact that the index i being balanced at time t is not ϵ-balanced, and thus $O((\frac{\epsilon}{w})^n)\sqrt{\|a_{.,i}^{(t)}\|_1 \|a_{i,.}^{(t)}\|_1} = o(1)\big(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1}\big)^2$.

Hence, using $O(n \log(w/\epsilon))$ bits of precision affects the reduction in the value of function f in each iteration by at most a constant factor, and therefore the proof of Theorem 2 using finite precision is essentially the same as the proof above using infinite precision. □

2.2 Round-Robin Balancing (the original algorithm)

Recall that original Osborne's algorithm balances indices in a fixed round-robin order. Although the greedy variant of the Osborne's iteration is a simple modification of the implementation, the convergence rate of the original algorithm (with no change) is interesting. This is important because the original algorithm has a slightly simpler implementation, and also because this is the implementation used in almost all numerical linear algebra software including MATLAB, LAPACK and EISPACK (refer to [13, 7] for further background). We answer this question in the following theorem.

Theorem 3. *Given an $n \times n$ matrix A, the original implementation of the Osborne's iteration outputs an ϵ-balanced matrix in $O(\epsilon^{-2} n^2 \log w)$ iterations totaling $O(\epsilon^{-2} mn \log w)$ arithmetic operations over $O(n \log(w/\epsilon))$-bit numbers (m is the number of non-zero entries of A).*

Proof. In the original Osborne algorithm, the indices are balanced in a fixed round-robin order. A *round* of balancing is a sequence of n balancing operations where each index is balanced exactly once. Thus, in the OPR algorithm all n indices are balanced in the same order every round. We prove a more general statement that any algorithm that balances indices in rounds (even if the indices are not balanced in the same order every round) obtains an ϵ-balanced matrix in at most $O((n\log w)/\epsilon^2)$ rounds. To this end, we show that applying a round of balancing to a matrix that is not ϵ-balanced reduces the value of function f at least by a factor of $1-\epsilon^2/16n$.

To simplify notation, we consider applying a round of balancing to the initial matrix $A^{(1)} = A$. The argument clearly holds for any time-t matrix $A^{(t)}$. If A is not ϵ-balanced, by Lemma 3 and Corollary 3, there exists an index i such that by balancing i the value of f is reduced by:

$$f(\mathbf{x}^{(\mathbf{1})}) - f(\mathbf{x}^{(\mathbf{2})}) = \left(\sqrt{\|a_{.,i}\|_1} - \sqrt{\|a_{i,.}\|_1}\right)^2 \geq \frac{\epsilon^2}{4} f(\mathbf{x}^{(\mathbf{1})}). \tag{2.4}$$

If i is the first index to balance in the next round of balancing, then in that round the value of f is reduced at least by a factor of $1-\epsilon^2/4 \geq 1-\epsilon^2/16n$, and we are done. Consider the graph G_A corresponding to the matrix A. If node i is not the first node in G_A to be balanced, then some of its neighbors in the graph G_A might be balanced before i. The main problem is that balancing neighbors of i before i may reduce the imbalance of i significantly, so we cannot argue that when we reach i and balance it the value of f reduces significantly. Nevertheless, we show that balancing i and its neighbors in this round will reduce the value of f by at least the desired amount. Let t denote the time that i is balanced in the round. For every arc (j,i) into i, let $\delta_j = |a_{ji} - a_{ji}^{(t)}|$, and for every arc (i,j) out of i let $\sigma_j = |a_{ij} - a_{ij}^{(t)}|$. These values measure the weight change of these arcs due to balancing a neighbor of i at any time since the beginning of the round. Note that in the current round and before balancing i, the weight of each arc incident on i is changed at most once by balancing its other endpoint. Thus, if and arc a_{ij} changes value to $a_{ij}^{(t)}$ at time t, we know that the weight change must have been due to balancing node j. The next lemma shows if the weight of an arc incident on i has changed since the beginning of the round, it must have reduced the value of f.

Claim 1. *If balancing node j changes a_{ji} to $a_{ji}+\Delta$, then the balancing reduces the value of f by* *at least Δ^2/a_{ji}. Similarly if balancing node j changes a_{ij} to $a_{ij}+\Delta$, then the balancing reduces the value of f by at least Δ^2/a_{ij}.*

Proof. To simplify notation we assume without loss of generality that j is balanced in the first iteration of the round. If balancing j changes a_{ji} to $a_{ji}+\Delta$, then by the definition of balancing,

$$\frac{a_{ji}+\Delta}{a_{ji}} = \sqrt{\frac{\|a_{.,j}\|_1}{\|a_{j,.}\|_1}}. \tag{2.5}$$

Thus, by Lemma 1 the value of f reduces by

$$\begin{aligned}\left(\sqrt{\|a_{.,j}\|_1} - \sqrt{\|a_{j,.}\|_1}\right)^2 &= \left(\sqrt{\frac{\|a_{.,j}\|_1}{\|a_{j,.}\|_1}} - 1\right)^2 \|a_{j,.}\|_1 \\ &= \left(\frac{a_{ji}+\Delta}{a_{ji}} - 1\right)^2 \|a_{j,.}\|_1 \\ &= \left(\frac{\Delta}{a_{ji}}\right)^2 \|a_{j,.}\|_1 \geq \frac{\Delta^2}{a_{ji}}\end{aligned}$$

The proof for the second part of the claim is similar. □

Going back to the proof of Theorem 3, let t denote the iteration in the round that i is balanced. By Claim 1, balancing neighbors of i has already reduced the value of f by

$$\sum_{j:(j,i)\in E} \frac{\delta_j^2}{a_{ji}} + \sum_{j:(i,j)\in E} \frac{\sigma_j^2}{a_{ij}}. \tag{2.6}$$

Balancing i reduces value of f by an additional $\left(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1}\right)^2$, so the value of f in the current round is reduced by at least:

$$R = \sum_{j:(j,i)\in E} \frac{\delta_j^2}{a_{ji}} + \sum_{j:(i,j)\in E} \frac{\sigma_j^2}{a_{ij}} + \left(\sqrt{\|a_{.,i}^{(t)}\|_1} - \sqrt{\|a_{i,.}^{(t)}\|_1}\right)^2$$

Assume without loss of generality that $\|a_{i,.}\|_1 > \|a_{.,i}\|_1$. To lower bound R, we consider two cases:

case (i): $\displaystyle\sum_{j:(j,i)\in E} \delta_j + \sum_{j:(i,j)\in E} \sigma_j \geq \frac{1}{2}(\|a_{i,.}\|_1 - \|a_{.,i}\|_1)$.

In this case,

$$
\begin{aligned}
R &\geq \sum_{j:(j,i)\in E} \frac{\delta_j^2}{a_{ji}} + \sum_{j:(i,j)\in E} \frac{\sigma_j^2}{a_{ij}} \\
&\geq \frac{1}{\|a_{.,i}\|_1} \sum_{j:(j,i)\in E} \delta_j^2 + \frac{1}{\|a_{i,.}\|_1} \sum_{j:(i,j)\in E} \sigma_j^2 \\
&\geq \frac{1}{n\|a_{.,i}\|_1} (\sum_{j:(j,i)\in E} \delta_j)^2 + \frac{1}{n\|a_{i,.}\|_1} (\sum_{j:(i,j)\in E} \sigma_j)^2, \qquad (2.7)
\end{aligned}
$$

where the last inequality follows by Cauchy-Schwarz inequality. By assumption of case (i),

$$
\max(\sum_{j:(j,i)\in E} \delta_j, \sum_{j:(i,j)\in E} \sigma_j) \geq \frac{1}{4}(\|a_{i,.}\|_1 - \|a_{.,i}\|_1). \qquad (2.8)
$$

Equations 2.7 and 2.8 together imply that

$$
\begin{aligned}
R &\geq \frac{(\sum_{j:(j,i)\in E} \delta_j)^2 + (\sum_{j:(i,j)\in E} \sigma_j)^2}{n \max(\|a_{.,i}\|_1, \|a_{i,.}\|_1)} \\
&\geq \frac{1}{16n} \frac{(\|a_{i,.}\|_1 - \|a_{.,i}\|_1)^2}{\max(\|a_{.,i}\|_1, \|a_{i,.}\|_1)} \\
&= \frac{(\sqrt{\|a_{.,i}\|_1} - \sqrt{\|a_{i,.}\|_1})^2 (\sqrt{\|a_{.,i}\|_1} + \sqrt{\|a_{i,.}\|_1})^2}{16n \max(\|a_{.,i}\|_1, \|a_{i,.}\|_1)} \\
&\geq \frac{1}{16n} \left(\sqrt{\|a_{.,i}\|_1} - \sqrt{\|a_{i,.}\|_1}\right)^2 .
\end{aligned}
$$

case (ii): $\displaystyle\sum_{j:(j,i)\in E} \delta_j + \sum_{j:(i,j)\in E} \sigma_j < \frac{1}{2}(\|a_{i,.}\|_1 - \|a_{.,i}\|_1).$

By definition of δ_j's and σ_j's:

$$
\|a_{.,i}\|_1 - \sum_{j:(j,i)\in E} \delta_j \leq \|a_{.,i}^{(t)}\|_1 \leq \|a_{.,i}\|_1 + \sum_{j:(j,i)\in E} \delta_j \qquad (2.9)
$$

$$
\|a_{i,.}\|_1 - \sum_{j:(i,j)\in E} \sigma_j \leq \|a_{i,.}^{(t)}\|_1 \leq \|a_{i,.}\|_1 + \sum_{j:(i,j)\in E} \sigma_j . \qquad (2.10)
$$

Combining Equations 2.9 and 2.10, and the assumption of case (ii) gives:

$$
\begin{aligned}
&\|a_{i,.}^{(t)}\|_1 + \|a_{.,i}^{(t)}\|_1 \\
&\qquad \leq \|a_{i,.}\|_1 + \|a_{.,i}\|_1 + \sum_{j:(i,j)\in E} \sigma_j + \sum_{j:(j,i)\in E} \delta_j \\
&\qquad \leq 2\left(\|a_{i,.}\|_1 + \|a_{.,i}\|_1\right) \qquad (2.11)
\end{aligned}
$$

and

$$
\begin{aligned}
&\|a^{(t)}_{i,.}\|_1 - \|a^{(t)}_{.,i}\|_1 \\
&\qquad \geq \|a_{i,.}\|_1 - \|a_{.,i}\|_1 - \sum_{j:(i,j)\in E} \sigma_j - \sum_{j:(j,i)\in E} \delta_j \\
&\qquad \geq \frac{1}{2}\left(\|a_{i,.}\|_1 - \|a_{.,i}\|_1\right). \qquad (2.12)
\end{aligned}
$$

Using Equations 2.11 and 2.12, we can write:

$$
\begin{aligned}
R &\geq \left(\sqrt{\|a^{(t)}_{i,.}\|_1} - \sqrt{\|a^{(t)}_{.,i}\|_1}\right)^2 \\
&= \frac{\left(\|a^{(t)}_{.,i}\|_1 - \|a^{(t)}_{i,.}\|_1\right)^2}{\left(\sqrt{\|a^{(t)}_{.,i}\|_1} + \sqrt{\|a^{(t)}_{i,.}\|_1}\right)^2} \geq \frac{\left(\|a_{i,.}\|_1 - \|a_{.,i}\|_1\right)^2}{4\left(\sqrt{\|a^{(t)}_{.,i}\|_1} + \sqrt{\|a^{(t)}_{i,.}\|_1}\right)^2} \\
&\geq \frac{\left(\|a_{i,.}\|_1 - \|a_{.,i}\|_1\right)^2}{8\left(\|a^{(t)}_{i,.}\|_1 + \|a^{(t)}_{.,i}\|_1\right)} \geq \frac{\left(\|a_{i,.}\|_1 - \|a_{.,i}\|_1\right)^2}{16\left(\|a_{i,.}\|_1 + \|a_{.,i}\|_1\right)} \\
&\geq \frac{1}{16}\left(\sqrt{\|a_{i,.}\|_1} - \sqrt{\|a_{.,i}\|_1}\right)^2.
\end{aligned}
$$

Thus, we have shown in both cases that in one round the balancing operations on node i and its neighbors reduces the value of f by at least

$$
\frac{1}{16n}\left(\sqrt{\|a_{.,i}\|_1} - \sqrt{\|a_{i,.}\|_1}\right)^2, \qquad (2.13)
$$

which in turn is at least $\Omega(\frac{\epsilon^2}{n} f(\mathbf{x^{(1)}}))$ by Equation 2.4. Thus, we have shown that if A is not ϵ-balanced, one round of balancing (where each index is balanced exactly once) reduces the objective function f by a factor of at least $1 - \Omega\left(\frac{\epsilon^2}{n} f(\mathbf{x^{(1)}})\right)$. By an argument similar to the one in the proof of Theorem 2, we get that the algorithm obtains an ϵ-balanced matrix in at most $O(\epsilon^{-2} n \log w)$ rounds. Since each round has n balancing operations, the total number of balancing operations is at most $O(\epsilon^{-2} n^2 \log w)$. The number of arithmetic operations in each round is $O(m)$ because the number of arithmetic operations in balancing each node is proportional to the number of arcs incident on that node. Thus, the original Osborne's algorithm obtains an ϵ-balanced matrix using $O(\epsilon^{-2} mn \log w)$ arithmetic operations.

The proof above assumes exact arithmetics with infinite precision. As with Theorem 2, it is easy to modify the proof to show that the algorithm has similar performance if all numbers are truncated to $O(n\log(w/\epsilon))$ bits of precision. □

2.3 Randomized Balancing

In variants of Osborne's algorithm that we considered so far (see Theorem 2 and Theorem 3), the arithmetic operations were applied to $O(n\log(w/\epsilon))$-bit numbers. This will cause an additional factor of $O(n\log(w/\epsilon))$ in the running time of the algorithm. In this section we fix this issue by presenting a randomized variant of the algorithm that applies arithmetic operations to numbers of $O(\log(nw/\epsilon))$ bits. Thus, we obtain an algorithm for balancing that runs in nearly linear time in m. While the greedy algorithm works by picking the node i that maximizes $(\sqrt{\|a_{i,\cdot}\|} - \sqrt{\|a_{\cdot,i}\|})^2$, the key idea of the randomized algorithm is sampling a node for balancing using sampling probabilities that do not depend on the difference in arc weights (the algorithm uses low-precision rounded weights, so this can affect significantly the difference). Instead, our sampling probabilities depend on the sum of weights of the arcs incident on a node.

We first introduce some notation. For convenience we use in this section $\|\cdot\|$ instead of $\|\cdot\|_1$ to denote the L_1 norm. We use $O(\log(nw/\epsilon))$ bits of precision to approximate x_i-s with $\hat{x}_i$-s. Thus, $x_i - 2^{-O(\log(nw/\epsilon))} \le \hat{x}_i \le x_i$. In addition to maintaining $\widehat{\mathbf{x}}^{(t)} = (\hat{x}_1^{(t)}, \hat{x}_2^{(t)}, \dots, \hat{x}_n^{(t)})$ at every time t, the algorithm also maintains for every i and j the value of $\widehat{a}_{ij}^{(t)}$ which is $a_{ij}^{(t)} = a_{ij}e^{\hat{x}_i^{(t)} - \hat{x}_j^{(t)}}$ truncated to $O(\log(nw/\epsilon))$ bits of precision. We set the hidden constant to give a truncation error of $r = (\epsilon/wn)^{10}a_{\min}$, so $a_{ij}^{(t)} - r \le \widehat{a}_{ij}^{(t)} \le a_{ij}^{(t)}$. To be precise, the hidden constant is 11, and we get a truncation error of $(\epsilon/wn)^{11} \le (\epsilon/wn)^{10}a_{\min} = r$, where the inequality assumes w.l.o.g. that $\sum_{i,j} a_{ij} \ge 1$, which can be ensured by scaling A.

The algorithm also maintains for every i, $\|\widehat{a}_{i,\cdot}^{(t)}\| = \sum_{j=1}^n \widehat{a}_{ij}^{(t)}$ and $\|\widehat{a}_{\cdot,i}^{(t)}\| = \sum_{j=1}^n \widehat{a}_{ji}^{(t)}$. For every i, we use the notation $\|a_{i,\cdot}^{(t)}\| = \sum_{j=1}^n a_{ij}^{(t)}$ and $\|a_{\cdot,i}\| = \sum_{j=1}^n a_{ji}^{(t)}$. Note that the algorithm does not maintain the values $a_{ij}^{(t)}$, $\|a_{\cdot,i}^{(t)}\|$ or $\|a_{i,\cdot}^{(t)}\|$.

Algorithm 2 RandomBalance(A, ϵ)

Input: Matrix $A \in \mathbb{R}^{n\times n}$, ϵ

Output: An ϵ-balanced matrix

$r = a_{\min} \cdot (\epsilon/wn)^{10}$
Let $\widehat{a}_{ij}^{(1)} = a_{ij}^{(1)}$ for all i and j
Let $\|\widehat{a}_{i,\cdot}^{(t)}\| = \|a_{i,\cdot}^{(t)}\|$ and $\|\widehat{a}_{\cdot,i}^{(t)}\| = \|a_{\cdot,i}^{(t)}\|$ for all i
for $t = 1$ to $O(\epsilon^{-2} \ln w)$ **do**
  Pick i randomly with probability $p_i = \frac{\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\|}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}}$
  if $\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\| \geq \epsilon a_{\min}/10wn$ **then**
    $\widehat{M}_i = \max\{\|\widehat{a}_{i,\cdot}^{(t)}\|, \|\widehat{a}_{\cdot,i}^{(t)}\|\}$, $\widehat{m}_i = \min\{\|\widehat{a}_{i,\cdot}^{(t)}\|, \|\widehat{a}_{\cdot,i}^{(t)}\|\}$
    if $\widehat{m}_i = 0$ **or** $\widehat{M}_i/\widehat{m}_i \geq 1 + \epsilon/n$ **then**
      if $\widehat{m}_i \neq 0$ **then** $\alpha = \frac{1}{2}\ln(\|\widehat{a}_{\cdot,i}^{(t)}\|/\|\widehat{a}_{i,\cdot}^{(t)}\|)$
      else if $\widehat{m}_i = \|\widehat{a}_{\cdot,i}^{(t)}\| = 0$ **then** $\alpha = \frac{1}{2}\ln(nr/\|\widehat{a}_{i,\cdot}^{(t)}\|)$
      else if $\widehat{m}_i = \|\widehat{a}_{i,\cdot}^{(t)}\| = 0$ **then** $\alpha = \frac{1}{2}\ln(\|\widehat{a}_{\cdot,i}^{(t)}\|/nr)$
      end if
      Let $\widehat{\mathbf{x}}^{(t+1)} \leftarrow \widehat{\mathbf{x}}^{(t)} + \alpha \mathbf{e}_i$ $\big($truncated to $O(\log(\frac{nw}{\epsilon}))$ bits of precision$\big)$
      for $j = 1$ to n **do**
        if j is a neighbor of i **then**
          $\widehat{a}_{ij}^{(t+1)} \leftarrow a_{ij} e^{\hat{x}_i^{(t+1)} - \hat{x}_j^{(t+1)}}$ and
          $\widehat{a}_{ji}^{(t+1)} \leftarrow a_{ji} e^{\hat{x}_j^{(t+1)} - \hat{x}_i^{(t+1)}}$, $\big($truncated to $O(\log(nw/\epsilon))$ bits$\big)$
          $\|\widehat{a}_{j,\cdot}^{(t+1)}\| = \|\widehat{a}_{j,\cdot}^{(t)}\| - \widehat{a}_{ji}^{(t)} + \widehat{a}_{ji}^{(t+1)}$ and $\|\widehat{a}_{\cdot,j}^{(t+1)}\| = \|\widehat{a}_{\cdot,j}^{(t)}\| - \widehat{a}_{ij}^{(t)} + \widehat{a}_{ij}^{(t+1)}$
        end if
      end for
      $\|\widehat{a}_{i,\cdot}^{(t+1)}\| = \sum_{j=1}^{n} \widehat{a}_{ij}^{(t+1)}$ and
      $\|\widehat{a}_{\cdot,i}^{(t+1)}\| = \sum_{j=1}^{n} \widehat{a}_{ji}^{(t+1)}$
    end if
  end if
end for
return the resulting matrix

The algorithm works as follows (see the pseudo-code of Algorithm 2). In each iteration it samples an index i with probability

$$p_i = \frac{\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\|}{\sum_i \|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\|} = \frac{\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\|}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}}.$$

If i is sampled, a balancing operation is applied to index i only if the arcs incident on i have significant weight (line 6), and i's imbalance is sufficiently large (line 8). Put $\widehat{M_i} = \max\{\|\widehat{a}_{i,\cdot}^{(t)}\|, \|\widehat{a}_{\cdot,i}^{(t)}\|\}$ and $\widehat{m}_i = \min\{\|\widehat{a}_{i,\cdot}^{(t)}\|, \|\widehat{a}_{\cdot,i}^{(t)}\|\}$. The imbalance is considered large if $\widehat{m}_i = 0$ (this can happen because of the low precision), or if $\widehat{m}_i \neq 0$ and $\frac{\widehat{M_i}}{\widehat{m}_i} \geq 1 + \frac{\epsilon}{n}$. A balancing operation is done by adding α to $\widehat{\mathbf{x}}_i^{(t)}$, where $\alpha = \frac{1}{2}\ln(\|\widehat{a}_{\cdot,i}^{(t)}\|/\|\widehat{a}_{i,\cdot}^{(t)}\|)$, unless $\widehat{m}_i = 0$, in which case we replace the 0 value by nr. This updates the weights of the arcs incident on i. Also, the L_1 norms of changed rows and columns are updated.

Note that in the pseudo-code, $\leftarrow$ indicates an assignment where the value on the right-hand side is computed to $O(\log(nw/\epsilon))$ bits of precision. In contrast, $=$ indicates an exact equlaity. Thus, we have

$$\alpha - (\epsilon/wn)^{10} \leq \widehat{x}_i^{(t+1)} - \widehat{x}_i^{(t)} \leq \alpha \tag{2.14}$$

and

$$a_{ij}e^{\hat{x}_i^{(t+1)} - \hat{x}_j^{(t+1)}} - r \leq \widehat{a}_{ij}^{(t+1)} \leq a_{ij}e^{\hat{x}_i^{(t+1)} - \hat{x}_j^{(t+1)}},$$
$$a_{ji}e^{\hat{x}_i^{(t+1)} - \hat{x}_j^{(t+1)}} - r \leq \widehat{a}_{ji}^{(t+1)} \leq a_{ji}e^{\hat{x}_i^{(t+1)} - \hat{x}_j^{(t+1)}}.$$

We are now ready to state and prove the main Theorem of this section.

Theorem 4. *With probability at least $\frac{9}{10}$, Algorithm 2 returns an ϵ-balanced matrix using $O(m + \epsilon^{-2}n\log w)$ arithmetic operations over $O(\log(wn/\epsilon))$-bit numbers.*

The idea of proof is to show that in every iteration of the algorithm we reduce $f(\cdot)$ by at least a factor of $1 - \Omega(\epsilon^2)$. Before we prove the theorem, we state and prove a couple of useful lemmas.

Fix an iteration t, and define three sets of indices as follows: $A = \{i : \|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\| \geq \epsilon a_{\min}/10wn\}$, $B = \{i : \widehat{m}_i \neq 0 \wedge \widehat{M}_i/\widehat{m}_i \geq 1 + \epsilon/n\}$, and $C = \{i : \widehat{m}_i = 0\}$. If the random index

i satisfies $i \notin A$ or $i \in A \setminus (B \cup C)$, the algorithm does not perform any balancing operation on i. The following lemma states that the expected decrease due to balancing such indices is small, and thus skipping them does not affect the speed of convergence substantially.

Lemma 5. *For every iteration t, $\sum_{i \notin A \cap (B \cup C)} p_i \cdot \left(\sqrt{\|a_{i,.}^{(t)}\|} - \sqrt{\|a_{.,i}^{(t)}\|}\right)^2 < \frac{3\epsilon^2}{n} \cdot f(\widehat{\mathbf{x}}^{(t)})$, where p is the probability distribution over indices at time t.*

Proof. Notice that for every i,

$$\begin{aligned}
&(\|\widehat{a}_{i,.}^{(t)}\| + \|\widehat{a}_{.,i}^{(t)}\|) \cdot \left(\sqrt{\|a_{i,.}^{(t)}\|} - \sqrt{\|a_{.,i}^{(t)}\|}\right)^2 \\
&\leq (\|\widehat{a}_{i,.}^{(t)}\| + \|\widehat{a}_{.,i}^{(t)}\|) \cdot \frac{\left(\|a_{i,.}^{(t)}\| - \|a_{.,i}^{(t)}\|\right)^2}{\|a_{i,.}^{(t)}\| + \|a_{.,i}^{(t)}\|} \\
&\leq \left(\|a_{i,.}^{(t)}\| - \|a_{.,i}^{(t)}\|\right)^2, \qquad (2.15)
\end{aligned}$$

because $\|\widehat{a}_{i,.}^{(t)}\| + \|\widehat{a}_{.,i}^{(t)}\| \leq \|a_{i,.}^{(t)}\| + \|a_{.,i}^{(t)}\|$. We first bound the sum over $i \notin A$.

$$\begin{aligned}
&\sum_{i \notin A} p_i \cdot \left(\sqrt{\|a_{i,.}^{(t)}\|} - \sqrt{\|a_{.,i}^{(t)}\|}\right)^2 \\
&= \sum_{i \notin A} \frac{\|\widehat{a}_{i,.}^{(t)}\| + \|\widehat{a}_{.,i}^{(t)}\|}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot \left(\sqrt{\|a_{i,.}^{(t)}\|} - \sqrt{\|a_{.,i}^{(t)}\|}\right)^2 \\
&\leq \frac{1}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot \sum_{i \notin A} \left(\|a_{i,.}^{(t)}\| - \|a_{.,i}^{(t)}\|\right)^2 \\
&\leq \frac{1}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot \sum_{i \notin A} \left(\|\widehat{a}_{i,.}^{(t)}\| + \|\widehat{a}_{.,i}^{(t)}\| + 2nr\right)^2 \\
&\leq \frac{1}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot \sum_{i \notin A} \left(\frac{2\epsilon a_{\min}}{10wn}\right)^2 \\
&\leq \frac{1}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot n \cdot \left(\frac{\epsilon a_{\min}}{5wn}\right)^2 \\
&= \frac{\epsilon^2}{50n} \frac{1}{\sum_{i,j} \widehat{a}_{ij}^{(t)}} \frac{a_{\min}^2}{w^2} \\
&\leq \frac{\epsilon^2}{25n} \cdot a_{\min} \leq \frac{\epsilon^2}{25n} \cdot f(\widehat{\mathbf{x}}^{(t)}) \qquad (2.16)
\end{aligned}$$

where the first inequality holds by (2.15), the second inequality follows because, for every j, $a_{ij}^{(t)} \leq$

$\widehat{a}_{ij}^{(t)} + r$ and $a_{ji}^{(t)} \leq \widehat{a}_{ji}^{(t)} + r$, and the third inequality follows because $\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\| < \epsilon a_{\min}/10wn$ and $2nr < \epsilon a_{\min}/10wn$.

Next, we bound the sum over $i \in A \setminus (B \cup C)$. Recall $\widehat{M_i} = \max\{\|\widehat{a}_{i,\cdot}^{(t)}\|, \|\widehat{a}_{\cdot,i}^{(t)}\|\}$ and $\widehat{m}_i = \min\{\|\widehat{a}_{i,\cdot}^{(t)}\|, \|\widehat{a}_{\cdot,i}^{(t)}\|\}$. Also let M_i and m_i denote the untruncated versions of $\widehat{M_i}$ and $\widehat{m}_i$ respectively. Specifically,

$$M_i = \begin{cases} \|a_{i,\cdot}^{(t)}\| & \text{if } \widehat{M_i} = \|\widehat{a}_{i,\cdot}^{(t)}\| \\ \|a_{\cdot,i}^{(t)}\| & \text{if } \widehat{M_i} = \|\widehat{a}_{\cdot,i}^{(t)}\| \end{cases}$$

and m_i is similarly defined to correspond to $\widehat{m}_i$.

Let $k = \arg\max_{i \in A \setminus (B \cup C)} (M_i - m_i)^2$. We have

$$\begin{aligned}
&\sum_{i \in A \setminus (B \cup C)} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \\
&= \sum_{i \in A \setminus (B \cup C)} \frac{\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\|}{2 \sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \\
&\leq \frac{1}{2 \sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot \sum_{i \in A \setminus (B \cup C)} \left(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\| \right)^2 \qquad \text{By (2.15)} \\
&\leq \frac{1}{2 \sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot \sum_{i \in A \setminus (B \cup C)} (M_i - m_i)^2 \\
&\leq \frac{1}{2 \sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot n \cdot m_k^2 \left(\frac{M_k}{m_k} - 1 \right)^2 .
\end{aligned} \tag{2.17}$$

To bound the last quantity, we prove an upper bound on $\frac{M_k}{m_k}$ using the fact that $\frac{\widehat{M_k}}{\widehat{m}_k} < 1 + \frac{\epsilon}{n}$. As $k \in A$, we have $\widehat{M_k} + \widehat{m}_k = \|\widehat{a}_{k,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,k}^{(t)}\| \geq \frac{\epsilon a_{\min}}{10wn}$. Thus, $\widehat{M_k} \geq \frac{\epsilon a_{\min}}{20wn}$. Combining this with $\frac{\widehat{M_k}}{\widehat{m}_k} < 1 + \frac{\epsilon}{n}$ implies that $\widehat{m}_k > \frac{1}{2}\widehat{M_k} \geq \frac{\epsilon a_{\min}}{40wn}$. Hence,

$$\begin{aligned}
\frac{M_k}{m_k} &\leq \frac{M_k}{\widehat{m}_k} \leq \frac{\widehat{M_k} + nr}{\widehat{m}_k} \leq \frac{\widehat{M_k}}{\widehat{m}_k} + \frac{nr}{\epsilon a_{\min}/40wn} \\
&= \frac{\widehat{M_k}}{\widehat{m}_k} + 40n \cdot \left(\frac{\epsilon}{wn} \right)^9 \leq \frac{\widehat{M_k}}{\widehat{m}_k} + \frac{40\epsilon^9}{n^8} \leq 1 + \frac{2\epsilon}{n}
\end{aligned}$$

where we used $w \geq 1$. Using the upper bound on $\frac{M_k}{m_k}$, we obtain

$$\begin{aligned}
&\sum_{i \in A \setminus (B \cup C)} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \\
&\leq \frac{1}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot n \cdot m_k^2 \left(\frac{M_k}{m_k} - 1 \right)^2 && \text{By (2.17)} \\
&\leq \frac{1}{2\sum_{i,j} \widehat{a}_{ij}^{(t)}} \cdot n \cdot m_k^2 \left(\frac{2\epsilon}{n} \right)^2 \\
&\leq \frac{2\epsilon^2}{n} \cdot m_k \leq \frac{2\epsilon^2}{n} \cdot f(\widehat{\mathbf{x}}^{(t)}), && (2.18)
\end{aligned}$$

where the penultimate inequality uses the fact that $m_k \leq \widehat{m}_k + nr \leq \widehat{m}_k + \frac{\epsilon a_{\min}}{40wn} < 2\widehat{m}_k \leq \widehat{M}_k + \widehat{m}_k \leq \sum_{i,j} \widehat{a}_{ij}^{(t)}$. Together (2.16) and (2.18) complete the proof of Lemma 5. □

Thus far, we have proved that the expected reduction in $f(\cdot)$ by balancing index i such that $i \notin A$ or $i \in A \setminus (B \cup C)$ is negligible. Next, we consider the rest of the indices. The following lemma shows a lower bound on the decrease in $f(\cdot)$, if a node $i \in A \cap (B \cup C)$ is balanced.

Lemma 6. *If $i \in A \cap (B \cup C)$ is balanced in iteration t, then $f(\widehat{\mathbf{x}}^{(t)}) - f(\widehat{\mathbf{x}}^{(t+1)}) \geq \frac{1}{10} \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2$.*

Proof. We will assume that $\epsilon < \frac{1}{10}$. We first consider the case that $i \in A \cap B$ (notice that $B \cap C = \emptyset$). The update using $O(\ln(wn/\epsilon))$ bits of precision gives $\widehat{x}_i^{(t)} + \alpha - (\epsilon/nw)^{10} \leq \widehat{x}_i^{(t+1)} \leq \widehat{x}_i^{(t)} + \alpha$, so

$$\sqrt{\frac{\|\widehat{a}_{\cdot,i}^{(t)}\|}{\|\widehat{a}_{i,\cdot}^{(t)}\|}} \cdot e^{\widehat{x}_i^{(t)} - (\epsilon/wn)^{10}} \leq e^{\widehat{x}_i^{(t+1)}} \leq \sqrt{\frac{\|\widehat{a}_{\cdot,i}^{(t)}\|}{\|\widehat{a}_{i,\cdot}^{(t)}\|}} \cdot e^{\widehat{x}_i^{(t)}}.$$

Therefore,

$$\begin{aligned}
\|a_{i,\cdot}^{(t+1)}\| &= \sum_{j=1}^{n} a_{ij} e^{\widehat{x}_i^{(t+1)} - \widehat{x}_j^{(t+1)}} \\
&\leq \sqrt{\frac{\|\widehat{a}_{\cdot,i}^{(t)}\|}{\|\widehat{a}_{i,\cdot}^{(t)}\|}} \cdot \sum_{j=1}^{n} a_{ij} e^{\widehat{x}_i^{(t)} - \widehat{x}_j^{(t)}} = \sqrt{\frac{\|\widehat{a}_{\cdot,i}^{(t)}\|}{\|\widehat{a}_{i,\cdot}^{(t)}\|}} \cdot \|a_{i,\cdot}^{(t)}\|, && (2.19)
\end{aligned}$$

and

$$\|a_{.,i}^{(t+1)}\| = \sum_{j=1}^{n} a_{ji} e^{\widehat{x}_j^{(t+1)} - \widehat{x}_i^{(t+1)}}$$

$$\leq e^{(\epsilon/wn)^{10}} \cdot \sqrt{\frac{\|\widehat{a}_{i,.}^{(t)}\|}{\|\widehat{a}_{.,i}^{(t)}\|}} \cdot \sum_{j=1}^{n} a_{ji} e^{\widehat{x}_j^{(t)} - \widehat{x}_i^{(t)}}$$

$$\leq (1 + 2(\epsilon/wn)^{10})) \cdot \sqrt{\frac{\|\widehat{a}_{i,.}^{(t)}\|}{\|\widehat{a}_{.,i}^{(t)}\|}} \cdot \|a_{.,i}^{(t)}\|. \tag{2.20}$$

We used the fact that $e^x \leq 1 + 2x$ for $x \leq \frac{1}{2}$. We will now use the notation $\widehat{M_i}$, $\widehat{m}_i$, M_i, and m_i (the reader can recall the definitions from the proof of Lemma 5). We first prove the following useful bounds on M_i and $\frac{m_i}{M_i}$.

Claim 2. *For every $i \in A$, $M_i \geq \dfrac{\epsilon a_{\min}}{20wn}$, and if i is in $A \cap B$, $\dfrac{m_i}{M_i} \leq 1 - \dfrac{\epsilon}{2n}$.*

Proof. For every $i \in A \cap B$, $\widehat{M_i} + \widehat{m_i} > \frac{\epsilon a_{\min}}{10wn}$, so $M_i \geq \widehat{M_i} \geq \frac{\widehat{M_i} + \widehat{m}_i}{2} > \frac{\epsilon a_{\min}}{20wn}$. Also, $m_i \leq \widehat{m}_i + nr$, so $\dfrac{m_i}{M_i} \leq \dfrac{\widehat{m}_i + nr}{\widehat{M_i}} \leq \dfrac{1}{1 + \epsilon/n} + \dfrac{nr}{\widehat{M_i}} \leq 1 - \dfrac{\epsilon}{2n}$. □

We also put $\delta = 2(\epsilon/wn)^{10}$, and $\sigma = \frac{\widehat{M_i}/\widehat{m}_i}{M_i/m_i}$. Thus, by inequality (2.19) and (2.20) decrease of function $f(\cdot)$ due to balancing i is

$$f(\widehat{\mathbf{x}}^{(t)}) - f(\widehat{\mathbf{x}}^{(t+1)}) = M_i + m_i - \|a_{.,i}^{(t+1)}\| - \|a_{i,.}^{(t+1)}\|$$

$$\geq M_i + m_i - (1 + \delta) \left(M_i \sqrt{\widehat{m}_i / \widehat{M_i}} + m_i \sqrt{\widehat{M_i} / \widehat{m}_i} \right)$$

$$= M_i + m_i - (1 + \delta) \left(\sqrt{1/\sigma} + \sqrt{\sigma} \right) \cdot \sqrt{M_i m_i} =$$

$$\left(\sqrt{M_i} - \sqrt{m_i} \right)^2 - \left(\frac{1 + \delta}{\sqrt{\sigma}} + (1 + \delta)\sqrt{\sigma} - 2 \right) \cdot \sqrt{M_i m_i} \tag{2.21}$$

Note that inequality (2.19) and (2.20) are used in the first inequality above.

To prove the lemma, we now consider three cases, and in each case show that

$$\left(\frac{1 + \delta}{\sqrt{\sigma}} + (1 + \delta)\sqrt{\sigma} - 2 \right) \cdot \sqrt{M_i m_i} \leq \frac{9}{10} \cdot \left(\sqrt{M_i} - \sqrt{m_i} \right)^2. \tag{2.22}$$

case (i): $1 \leq \sigma \leq 1 + \dfrac{\epsilon^4}{n^2}$.

Since $\epsilon < \frac{1}{10}$, we have

$$\begin{aligned}
&\left(\frac{1+\delta}{\sqrt{\sigma}} + (1+\delta)\sqrt{\sigma} - 2\right) \cdot \sqrt{\frac{m_i}{M_i}} \\
&\leq \left((1+\delta) + (1+\delta) \cdot \left(1 + \frac{\epsilon^4}{n^2}\right) - 2\right) \\
&\leq \frac{4\epsilon^4}{n^2} \\
&\leq \frac{9}{10}\left(1 - \sqrt{1 - \frac{\epsilon}{2n}}\right)^2 \\
&\leq \frac{9}{10}\left(1 - \sqrt{\frac{m_i}{M_i}}\right)^2
\end{aligned}$$

where the first inequality holds because $m_i/M_i \leq 1$ and $\sigma \in [1, 1 + \epsilon^4/n^2]$, the second inequality holds by definition of δ and the last inequality follows from Claim 2. By multiplying both sides of the inequality by M_i we obtain the desired bound.

case (ii): $\sigma < 1$.

We first prove a lower bound on the value of σ, as follows: $\frac{\widehat{M}_i}{\widehat{m}_i} \geq \frac{\widehat{M}_i}{m_i} \geq \frac{M_i - nr}{m_i} \geq \frac{M_i}{m_i}(1 - \frac{nr}{M_i}) \geq \frac{M_i}{m_i}\left(1 - \frac{20\epsilon^9}{n^8}\right)$, where the last inequality use the lower bound on M_i from Claim 2, and the value of r. Therefore, $\sigma = \frac{\widehat{M}_i/\widehat{m}_i}{M_i/m_i} \geq 1 - \frac{20\epsilon^9}{n^8}$. So we have

$$\begin{aligned}
&\left(\frac{1+\delta}{\sqrt{\sigma}} + (1+\delta)\sqrt{\sigma} - 2\right) \cdot \sqrt{\frac{m_i}{M_i}} \\
&\leq \frac{1+\delta}{\sqrt{1 - \frac{20\epsilon^9}{n^8}}} + (1+\delta) - 2 \\
&\leq (1+\delta)\left(1 + \frac{20\epsilon^9}{n^8}\right) + (1+\delta) - 2 \\
&\leq \frac{24\epsilon^9}{n^8} < \frac{4\epsilon^4}{n^2} \leq \frac{9}{10}\left(1 - \sqrt{1 - \frac{\epsilon}{2n}}\right)^2 \\
&\leq \frac{9}{10} \cdot \left(1 - \sqrt{\frac{m_i}{M_i}}\right)^2,
\end{aligned}$$

proving the desired inequality in this case. The first inequality holds because $\frac{m_i}{M_i} \leq 1$ and $1 - \frac{20\epsilon^9}{n^8} \leq \sigma \leq 1$.

case (iii): $\sigma > 1 + \dfrac{\epsilon^4}{n^2}$.

The idea is to show that M_i/m_i is large so the desired inequality follows. We know that $\frac{\sigma M_i}{m_i} = \frac{\widehat{M}_i}{\widehat{m}_i} \leq \frac{M_i}{\widehat{m}_i}$ and therefore $\widehat{m}_i \leq \frac{m_i}{\sigma}$. On the other hand, $\widehat{m}_i \geq m_i - nr$, so $m_i \leq \frac{nr}{1-1/\sigma}$. Clearly, $1/\sigma < 1 - \frac{\epsilon^4}{2n^2}$, so $m_i < \frac{nr}{\epsilon^4/2n^2}$. Also, by Claim 2, $M_i \geq \frac{\epsilon a_{\min}}{20wn}$. Therefore, $\frac{M_i}{m_i} \geq \frac{\epsilon a_{\min}/20wn}{2n^3 r/\epsilon^4} \geq \frac{n^6}{40\epsilon^5}$. Next, notice that since $\widehat{m}_i > 0$ it must be that $\widehat{m}_i \geq r$. Therefore, $m_i \leq \widehat{m}_i + nr \leq 2n\widehat{m}_i$. This implies that $\frac{\widehat{M}_i}{\widehat{m}_i} \leq \frac{M_i}{\widehat{m}_i} \leq 2n \cdot \frac{M_i}{m_i}$, so $\sigma \leq 2n$. Finally,

$$\left(\frac{1+\delta}{\sqrt{\sigma}} + (1+\delta)\sqrt{\sigma} - 2\right) \leq (1+\delta) \cdot \sqrt{2n} \leq \frac{1}{10} \cdot \sqrt{\frac{M_i}{m_i}},$$

with room to spare (using the lower bound on $\frac{M_i}{m_i}$). Multiplying both sides by $\sqrt{\frac{M_i}{m_i}}$ gives

$$\begin{aligned}\left(\frac{1+\delta}{\sqrt{\sigma}} + (1+\delta)\sqrt{\sigma} - 2\right) \cdot \sqrt{\frac{M_i}{m_i}} &\leq \frac{1}{10}\frac{M_i}{m_i} \\ &\leq \frac{9}{10}\left(\sqrt{\frac{M_i}{m_i}} - 1\right)^2,\end{aligned}$$

with more room to spare. Thus, we have shown that if $i \in A \cap B$, then (2.22) holds. This combined with (2.21) implies that in the case $i \in A \cap B$:

$$\begin{aligned} f(\widehat{\mathbf{x}}^{(t)}) - f(\widehat{\mathbf{x}}^{(t+1)}) &\geq \left(\sqrt{M_i} - \sqrt{m_i}\right)^2 - \left(\frac{1+\delta}{\sqrt{\sigma}} + (1+\delta)\sqrt{\sigma} - 2\right) \cdot \sqrt{M_i m_i} \\ &\geq \frac{1}{10}\left(\sqrt{M_i} - \sqrt{m_i}\right)^2. \end{aligned}$$

We now move on to the case $i \in A \cap C$, so $\widehat{M}_i + \widehat{m}_i \geq \frac{\epsilon a_{\min}}{10wn}$ and $\widehat{m}_i = 0$. In the algorithm, $\alpha = \frac{1}{2}\ln(nr/\|\widehat{a}^{(t)}_{i,\cdot}\|)$ or $\alpha = \frac{1}{2}\ln(\|\widehat{a}^{(t)}_{\cdot,i}\|/nr)$. Therefore, the idea is that we replace $\widehat{m}_i$ which is 0 by nr in some of the equations. In particular, $f(\widehat{\mathbf{x}}^{(t)}) - f(\widehat{\mathbf{x}}^{(t+1)}) = M_i + m_i - \|a^{(t+1)}_{\cdot,i}\| - \|a^{(t+1)}_{i,\cdot}\| \geq M_i + m_i - (1+\delta)\left(M_i\sqrt{\frac{nr}{\widehat{M}_i}} + m_i\sqrt{\frac{\widehat{M}_i}{nr}}\right)$ by replacing $\widehat{m}_i$ instead of nr in inequality (2.19) and (2.20). Note that since $\widehat{m}_i = 0$ then $m_i \leq nr$. Therefore, $\frac{\widehat{M}_i}{nr} \leq \frac{M_i}{nr} \leq \frac{M_i}{m_i}$. On the other hand, since

$i \in A$ by Claim 2 $\widehat{M_i} \geq \frac{\epsilon a_{\min}}{20wn}$, so $\frac{\widehat{M_i}}{nr} \geq \frac{\epsilon a_{\min}/20wn}{n(\epsilon/wn)^{10} a_{\min}} \geq \frac{n^8}{20\epsilon^9}$. Thus we get

$$\begin{aligned}
& f(\widehat{\mathbf{x}}^{(t)}) - f(\widehat{\mathbf{x}}^{(t+1)}) \\
& \geq M_i + m_i - (1+\delta)\left(M_i \sqrt{\frac{nr}{\widehat{M_i}}} + m_i \sqrt{\frac{\widehat{M_i}}{nr}} \right) \\
& \geq M_i + m_i - (1+\delta)\left(M_i \sqrt{\frac{20\epsilon^9}{n^8}} + m_i \sqrt{\frac{M_i}{m_i}} \right) \\
& \geq M_i + m_i - 2(1+\delta) M_i \sqrt{\frac{20\epsilon^9}{n^8}} \\
& \geq M_i \left(1 - \frac{20\epsilon^4}{n^4}\right) \geq \frac{1}{10} M_i \geq \frac{1}{10}(\sqrt{M_i} - \sqrt{m_i})^2,
\end{aligned}$$

where the third inequality holds because $m_i \sqrt{\frac{M_i}{m_i}} = M_i \sqrt{\frac{m_i}{M_i}} \leq M_i \sqrt{\frac{nr}{\widehat{M_i}}}$. This completes the proof of Lemma 6. □

Using Lemma 5 and Lemma 6, we can now prove Theorem 4.

Proof of Theorem 4. By Lemma 6, the expected decrease in $f(\cdot)$ in iteration t is lower bounded as follows.

$$\begin{aligned}
& \mathbb{E}[f(\widehat{\mathbf{x}}^{(t)}) - f(\widehat{\mathbf{x}}^{(t+1)})] \\
& \geq \sum_{i \in A \cap (B \cup C)} p_i \cdot \frac{1}{10} \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \\
& = \frac{1}{10} \cdot \sum_{i \in A} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 - \\
& \qquad \frac{1}{10} \cdot \sum_{i \notin A \cap (B \cup C)} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 .
\end{aligned}$$

The second term can be bounded, using Lemma 5, by $\sum_{i \notin A \cap (B \cup C)} \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \leq$

$\frac{3\epsilon^2}{n} \cdot f(\widehat{\mathbf{x}}^{(t)})$. For the first term, we can write

$$\begin{aligned}
&\sum_{i \in A} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \\
&\geq \sum_{i \in A} p_i \cdot \frac{(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\|)^2}{2(\|a_{i,\cdot}^{(t)}\| + \|a_{\cdot,i}^{(t)}\|)} \\
&= \sum_{i \in A} \frac{\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\|}{2\sum_{ij} \widehat{a}_{ij}^{(t)}} \cdot \frac{(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\|)^2}{2(\|a_{i,\cdot}^{(t)}\| + \|a_{\cdot,i}^{(t)}\|)} \\
&\geq \frac{1}{16} \sum_{i \in A} \frac{(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\|)^2}{\sum_{ij} a_{ij}^{(t)}} \\
&= \frac{1}{16} \Big(\sum_{i=1}^{n} \frac{(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\|)^2}{\sum_{ij} a_{ij}^{(t)}} - \sum_{i \notin A} \frac{(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\|)^2}{\sum_{ij} a_{ij}^{(t)}} \Big).
\end{aligned}$$

The last inequality above holds because $\frac{\widehat{M}_i}{M_i} \geq \frac{M_i - nr}{M_i} \geq \frac{1}{2}$ for all i in A, so $\frac{\|\widehat{a}_{i,\cdot}^{(t)}\| + \|\widehat{a}_{\cdot,i}^{(t)}\|}{\|a_{i,\cdot}^{(t)}\| + \|a_{\cdot,i}^{(t)}\|} \geq \frac{\widehat{M}_i}{2M_i} \geq \frac{1}{4}$. By definition $\sum_{i=1}^{n} \frac{(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\|)^2}{\sum_{ij} a_{ij}^{(t)}} = \frac{\|\nabla f(\widehat{\mathbf{x}}^{(t)})\|_2^2}{f(\widehat{\mathbf{x}}^{(t)})} > \epsilon^2 f(\widehat{\mathbf{x}}^{(t)})$ where the last inequality holds as long as the matrix is not ϵ-balanced, and $\frac{\|\nabla f(\widehat{\mathbf{x}}^{(t)})\|_2}{f(\widehat{\mathbf{x}}^{(t)})} > \epsilon$. Also,

$$\begin{aligned}
\sum_{i \notin A} \frac{(\|a_{i,\cdot}^{(t)}\| - \|a_{\cdot,i}^{(t)}\|)^2}{\sum_{ij} a_{ij}^{(t)}} &\leq \sum_{i \notin A} \frac{M_i^2}{\sum_{ij} a_{ij}^{(t)}} \leq \frac{(\sum_{i \notin A} M_i)^2}{n \sum_{ij} a_{ij}^{(t)}} \\
&\leq \frac{(2\sum_{i \notin A} \widehat{M}_i)^2}{n \sum_{ij} a_{ij}^{(t)}} \leq \frac{(2n\widehat{M}_i)^2}{n \sum_{ij} a_{ij}^{(t)}} \\
&\leq \frac{(2n\frac{\epsilon a_{\min}}{10nw})^2}{n \sum_{ij} a_{ij}^{(t)}} \leq \frac{\epsilon^2}{25n} f(\widehat{\mathbf{x}}^{(t)})
\end{aligned}$$

Thus, $\sum_{i \in A} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \geq \frac{1}{16}[\epsilon^2 f(\widehat{\mathbf{x}}^{(t)}) - \frac{\epsilon^2}{25n} f(\widehat{\mathbf{x}}^{(t)})] \geq \frac{\epsilon^2}{20} f(\widehat{\mathbf{x}}^{(t)})$. Combining everything together, we get

$$\begin{aligned}
&\mathbb{E}[f(\widehat{\mathbf{x}}^{(t)}) - f(\widehat{\mathbf{x}}^{(t+1)})] \geq \\
&\frac{1}{10} \cdot \sum_{i \in A} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 - \\
&\qquad \frac{1}{10} \cdot \sum_{i \notin A \cap (B \cup C)} p_i \cdot \left(\sqrt{\|a_{i,\cdot}^{(t)}\|} - \sqrt{\|a_{\cdot,i}^{(t)}\|} \right)^2 \\
&\geq \frac{1}{10} \cdot \left(\frac{\epsilon^2}{20} \cdot f(\widehat{\mathbf{x}}^{(t)}) - \frac{3\epsilon^2}{n} \cdot f(\widehat{\mathbf{x}}^{((t))}) \right) \geq \frac{\epsilon^2}{500} \cdot f(\widehat{\mathbf{x}}^{(t)}),
\end{aligned}$$

where the last inequality assumes $n \geq 100$. This implies that the expected number of iterations to obtain an ϵ-balanced matrix is $O(\epsilon^{-2}\log w)$ using arguments similar to the proof of Theorem 2. Markov's inequality implies that with probability $\frac{9}{10}$ an ϵ-balanced matrix is obtained in $O(\epsilon^{-2}\log w)$ iterations. It is easy to see that each iteration of the algorithm takes $O(n)$ arithmetic operations, and initializations take $O(m)$ arithmetic operations. So the total number of arithmetic operations of the algorithm is $O(m + \epsilon^{-2} n \log w)$ over $O(\log(wn/\epsilon))$-bit numbers. $\square$

CHAPTER 3

Convergence to a strictly ϵ-balanced matrix

In the previous chapter, we considered three variant of Osborne's iteration and proved upper bounds on their convergence rate to an ϵ-balanced matrix. In this chapter, we consider convergence to a *strictly* ϵ-balanced matrix. Recall from Definition 1, that a matrix A is ϵ-balanced if

$$\frac{\sqrt{\sum_{i=1}^{n}(\|a_{.,i}\|_1 - \|a_{i,.}\|_1)^2}}{\sum_{i,j} a_{i,j}} \leq \epsilon.$$

and it is strictly ϵ-balanced if for every index i of A (where $i \in [n]$)

$$\frac{\max\{\|a_{.,i}\|_1, \|a_{i,.}\|_1\}}{\min\{\|a_{.,i}\|_1, \|a_{i,.}\|_1\}} \leq 1 + \epsilon. \tag{3.1}$$

Being strictly ϵ-balanced requires that condition (3.1) holds for every index of the matrix, while being ϵ-balanced is a much weaker condition. The following lemma proves the relationship between the two definitions.

Lemma 7. *If matrix A is strictly ϵ-balanced, then it is also 2ϵ-balanced.*

Proof. For every $i \in [n]$ let $M_i = \max \|a_{.,i}\|_1, \|a_{i,.}\|_1$ and $m_i = \min \|a_{.,i}\|_1, \|a_{i,.}\|_1$. Being strictly ϵ-balanced implies that for all i, $\frac{M_i}{m_i} \leq 1 + \epsilon$. Thus, for every i, $\frac{M_i - m_i}{M_i + m_i} \leq \frac{M_i - \frac{M_i}{1+\epsilon}}{M_i + \frac{M_i}{1+\epsilon}} \leq \epsilon$. This implies that,

$$\frac{\sqrt{\sum_{i=1}^{n}(\|a_{.,i}\|_1 - \|a_{i,.}\|_1)^2}}{\sum_{i,j} a_{i,j}} \leq \frac{\sqrt{\sum_{i=1}^{n}(M_i - m_i)^2}}{\frac{1}{2}\sum_{i=1}^{n}(M_i + m_i)} \leq \frac{2\sum_{i=1}^{n}(M_i - m_i)}{\sum_{i=1}^{n}(M_i + m_i)} \leq 2\epsilon.$$

□

As the definition and the above lemma suggest bounding the rate of convergence to a strictly ϵ-balanced matrix is much harder than bounding the convergence to an ϵ-balanced matrix. Although, we showed that Osborne's algorithm and its variants converge in at most quadratic time to an ϵ-balanced matrix, these proofs don't provide any evidence that Osborne's algorithm converges to a strictly ϵ-balanced matrix even in a polynomial number of iterations.

The main difficulty with respect to previous work is the following. In the past chapter, to bound convergence rate to an ϵ-balanced matrix, we interpreted Osborne's balancing step to coordinate descent in optimizing the following convex function:

$$f(\mathbf{x}) = \sum_{i,j=1}^{n} a_{ij} e^{x_i - x_j}.$$

The convergence rate of coordinate descent can be bounded effectively as long as there is a choice of coordinate (i.e., index) for which the drop in the objective function in a single step is non-negligible compared with the current objective value. But if this is not the case, then one can argue only about the balance of each index relative to the sum of norms of all rows and columns. Indices that have relatively heavy weight (row norm + column norm) will indeed be balanced at this point. However, light-weight indices may be highly unbalanced. The naive remedy to this problem is to work down by scales. After balancing the matrix globally, heavy-weight indices are balanced, approximately, so they can be left alone, deactivated. Now there are light-weight indices that have become heavy-weight with respect to the remaining active nodes, so we can continue balancing the active indices until the relatively heavy-weight among them become approximately balanced, and so forth. The problem with the naive solution is that balancing the active indices shifts the weights of both active and inactive indices, and they move out of their initial scale. If the scale sets of indices keep changing, it is hard to argue that the process converges. Shifting between scales is precisely what our algorithm in this chapter and proof deal with. Light-weight indices that have become heavy-weight are easy to handle. They can keep being active. Heavy-weight indices that have become light-weight cannot continue to be inactive, because they are no longer guaranteed to be approximately balanced. Thus, in order to analyze convergence effectively, we need to bound the number (and global effect on weight) of these reactivation events.

3.1 Preliminaries

Recall that any implementation of Osborne's iteration can be thought of as computing vectors $\mathbf{x}^{(\mathbf{t})} \in \mathbb{R}^n$ for $t = 1, 2, \ldots$, where iteration t is applied to the matrix $(a_{ij}^{(t)}) = DAD^{-1}$ for $D = \mathtt{diag}(e^{x_1^{(t)}}, e^{x_2^{(t)}}, \ldots, e^{x_n^{(t)}})$. Thus, for all i, j, $a_{ij}^{(t)} = a_{ij} \cdot e^{x_i^{(t)} - x_j^{(t)}}$. Initially, $\mathbf{x}^{(1)} = (0, 0, \ldots, 0)$. A balancing step of the iteration chooses an index i, then sets $x_i^{(t+1)} = x_i^{(t)} + \frac{1}{2} \cdot \left(\ln \|a_{.,i}^{(t)}\|_1 - \ln \|a_{i,.}^{(t)}\|_1\right)$, and for all $j \neq i$, keeps $x_j^{(t+1)} = x_j^{(t)}$. For $\mathbf{x} \in \mathbb{R}^n$, we denote the sum of entries of the matrix DAD^{-1} for $D = \mathtt{diag}(e^{x_1}, e^{x_2}, \ldots, e^{x_n})$ by $f(\mathbf{x}) = f_A(\mathbf{x}) = \sum_{ij} a_{ij} \cdot e^{x_i - x_j}$. For any $n \times n$ non-negative matrix $B = (b_{ij})$, we denote by G_B the weighted directed graph with node set $\{1, 2, \ldots, n\}$, arc set $\{(i, j) : b_{ij} > 0\}$, where an arc (i, j) has weight b_{ij}.

Before describing our variant of Osborne's algorithm, we prove the following useful lemma that states a global condition on indices being ϵ-balanced.

Lemma 8. *Consider a matrix $B = DAD^{-1} = (b_{ij})_{n \times n}$, where $D = \mathtt{diag}(e^{x_1}, e^{x_2}, \ldots, e^{x_n})$, that was derived from A by a sequence of zero or more balancing operations. Let $\epsilon \in (0, 1/2]$, and put $\epsilon' = \frac{\epsilon^2}{64n^4}$. Suppose that $\|\nabla f_A(\vec{0})\|_1 \leq \epsilon' \cdot f_A(\vec{0})$. Then, for every $i \in [n]$ we have the following implication. If $\|b_{.,i}\|_1 + \|b_{i,.}\|_1 \geq \frac{1}{8n^3} \cdot f_A(\mathbf{x})$, then index i is ϵ-balanced in B.*

Proof. We will show the contrapositive claim that if a node is not ϵ-balanced then it must have low weight (both with respect to B). Let i be an index that is not ϵ-balanced in B. Without loss of generality we may assume that the in-weight is larger than the out-weight, so $\|b_{.,i}\|_1 / \|b_{i,.}\|_1 > 1 + \epsilon$. Consider what would happen if we balance index i in B, yielding a vector $\mathbf{x}'$ that differs from $\mathbf{x}$ only in the i-th coordinate.

$$\begin{aligned} f_A(\mathbf{x}) - f_A(\mathbf{x}') &= \left(\sqrt{\|b_{.,i}\|_1} - \sqrt{\|b_{i,.}\|_1}\right)^2 \\ &> \|b_{.,i}\|_1 \cdot \left(1 - \sqrt{\frac{1}{1+\epsilon}}\right)^2 \\ &> \frac{\epsilon^2}{16} \cdot \left(\|b_{.,i}\|_1 + \|b_{i,.}\|_1\right), \end{aligned} \tag{3.2}$$

where the equation follows from Lemma 1 and the last inequality uses the fact that $\epsilon \leq \frac{1}{2}$. On the other hand, we have

$$\begin{aligned} f_A(\mathbf{x}) - f_A(\mathbf{x}') &\le f_A(\vec{0}) - f_A(\mathbf{x}^*) \\ &\le \frac{n}{2} \cdot \|\nabla f_A(\vec{0})\|_1 \\ &\le \frac{n}{2} \cdot \epsilon' \cdot f_A(\vec{0}) \\ &= \frac{\epsilon^2}{128n^3} \cdot f_A(\vec{0}). \end{aligned} \tag{3.3}$$

where the first inequality follows from the the fact that every balancing step decreases f_A, the second inequality follows from Lemma 2, the third inequality follows from the assumption on $f_A(\vec{0})$, and the last equation follows from the choice of ϵ'. Combining the bounds on $f_A(\mathbf{x})-f_A(\mathbf{x}')$ in Equations (3.2) and (3.3) gives

$$\|b_{\cdot,i}\|_1 + \|b_{i,\cdot}\|_1 < \frac{1}{8n^3} \cdot f_A(\vec{0}),$$

and this completes the proof. □

3.2 Strict Balancing

We are now ready to present a variant of Osborne's iteration and prove that it converges in polynomial time to a strictly ϵ-balanced matrix. The algorithm, a procedure named StrictBalance, is defined in pseudocode labeled Algorithm 3 on page 41. Lemma 8 above motivates the main idea of contracting heavy nodes in step 14 of StrictBalance.

Our main theorem is the following.

Theorem 5. *StrictBalance(A, ϵ) returns a strictly ϵ-balanced matrix $B = DAD^{-1}$ after at most*

$$O\left(\epsilon^{-2} n^9 \log(wn/\epsilon) \log w / \log n\right)$$

balancing steps, using $O\left(\epsilon^{-2} n^{10} \log(wn/\epsilon) \log w / \log n\right)$ arithmetic operations over $O(n \log(w/\epsilon))$-bit numbers.

Algorithm 3 StrictBalance(A, ϵ)

Input: Matrix $A \in \mathbb{R}^{n \times n}$, ϵ

Output: A strictly ϵ-balanced matrix

$\mathcal{B}_1 = \varnothing$, $\tau_1 = 0$, $s = 1$, $\epsilon' = \epsilon^2/64n^4$, $\mathbf{x^{(1)}} = (0, \ldots, 0)$, $t = 1$
while $\mathcal{B}_s \neq [n]$ and there is $i \in [n]$ that is not ϵ-balanced **do**
  Define $f^{(\mathcal{B}_s)} : \mathbb{R}^n \to \mathbb{R}$, $f^{(\mathcal{B}_s)}(\mathbf{x}) = \sum_{i,j: i \notin \mathcal{B}_s \text{ or } j \notin \mathcal{B}_s} a_{ij} e^{x_i - x_j}$
  while $\dfrac{\|\nabla f^{(\mathcal{B}_s)}(\mathbf{x^{(t)}})\|_1}{f^{(\mathcal{B}_s)}(\mathbf{x^{(t)}})} > \epsilon'$ **do**
    Pick $i = \arg\max_{i \notin \mathcal{B}_s} \left\{ \left(\sqrt{\|a^{(t)}_{.,i}\|_1} - \sqrt{\|a^{(t)}_{i,.}\|_1} \right)^2 \right\}$
    Balance ith node: $\mathbf{x^{(t+1)}} = \mathbf{x^{(t)}} + \alpha_t \mathbf{e}_i$, where $\alpha_t = \ln \sqrt{\|a^{(t)}_{.,i}\|_1 / \|a^{(t)}_{i,.}\|_1}$
    $t \leftarrow t + 1$
    if $s > 1$ and $\|a^{(t)}_{.,i}\|_1 + \|a^{(t)}_{i,.}\|_1 < \tau_s$ for some $i \in \mathcal{B}_s \setminus \mathcal{B}_{s-1}$ **then**
      $\mathcal{B}_s = \mathcal{B}_s \setminus \{i \notin \mathcal{B}_{s-1} : \|a^{(t)}_{.,i}\|_1 + \|a^{(t)}_{i,.}\|_1 < \tau_s\}$
      Redefine $f^{(\mathcal{B}_s)} : \mathbb{R}^n \to \mathbb{R}$, $f^{(\mathcal{B}_s)}(\mathbf{x}) = \sum_{i,j: i \notin \mathcal{B}_s \text{ or } j \notin \mathcal{B}_s} a_{ij} e^{x_i - x_j}$
    end if
  end while
  $\tau_{s+1} = \dfrac{1}{4n^3} f^{(\mathcal{B}_s)}(\mathbf{x^{(t)}})$
  $\mathcal{B}_{s+1} = \mathcal{B}_s \cup \left\{ i : \|a^{(t)}_{.,i}\|_1 + \|a^{(t)}_{i,.}\|_1 \geq \tau_{s+1} \right\}$
  $s \leftarrow s + 1$
end while
return the resulting matrix

This polynomial but very large upper bound of $\tilde{O}(n^9)$ on the number of iterations should be viewed in the light of a $O(n^3)$ lower bound. Chen [2] shows that Osborne's algorithm on L_∞ needs at least $O(n^3)$ iterations to converge to a strictly ϵ-balanced matrix. She proves this lower bound for cycles of length n, so this lower bound naturally extends to other L_p norms. However, our work still leaves a large gap between lower and upper bound of the convergence rate of Osborne's iteration to a strictly ϵ-balanced matrix.

The proof of Theorem 5 uses a few arguments, given in the following lemmas. A *phase* of StrictBalance is one iteration of the outer while loop. Notice that in the beginning of this loop the variable s indexes the phase number (i.e., $s-1$ phases were completed thus far). Also in the beginning of the inner while loop the variable t indexes the total iteration number from all phases (i.e., $t-1$ balancing operations from all phases were completed thus far).

We identify outer loop iteration s with an interval $[t_s, t_{s+1}) = \{t_s, t_s+1, \dots, t_{s+1}-1\}$ of the inner loop iterations executed during phase s. At any time during the algorithm, variable $\mathcal{B}_s$ denotes the set of nodes of graph G_A that are contracted. We denote by $\mathcal{B}_{s,t}$ the value of $\mathcal{B}_s$ in the beginning of the inner while loop iteration number t (dubbed time t). If $t \in [t_j, t_{j+1})$, then $\mathcal{B}_{s,t}$ is defined only for $s \le j$. We also use $G^{(\mathcal{B}_{s,t})}$ to denote the graph that is obtained by contracting the nodes of set $\mathcal{B}_{s,t}$ in graph G_A. Also $f^{(\mathcal{B}_{s,t})}$ is the function corresponding to graph $G^{(\mathcal{B}_{s,t})}$ and $f^{(\mathcal{B}_{s,t})}(\mathbf{x}^{(\mathbf{t})})$ denotes the sum of weights of arcs of graph $G^{(\mathcal{B}_{s,t})}$ at time t. If set $\mathcal{B}_s$ is unchanged during an interval and there is no confusion, we may use $G^{(\mathcal{B}_s)}$ instead of $G^{(\mathcal{B}_{s,t})}$. Particularly we use $f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t})})$ instead of $f^{(\mathcal{B}_{s,t})}(\mathbf{x}^{(\mathbf{t})})$. We refer to the quantity $\|a^{(t)}_{.,i}\|_1 + \|a^{(t)}_{i,.}\|_1$ as the *weight* of node i at time t.

Lemma 9. *For every phase $s \ge 1$, for every $t \ge t_{s+1}$, $\mathcal{B}_{s,t} = \mathcal{B}_{s,t_{s+1}}$.*

Proof. The claim follows easily from the fact that any iteration $t \ge t_{s+1}$ belongs to a phase $s' > s$, so $\mathcal{B}_{s,t_{s+1}} \cap (\mathcal{B}_{s',t} \setminus \mathcal{B}_{s'-1,t}) = \emptyset$, and by line 8 and 9 of StrictBalance none of the nodes in $\mathcal{B}_{s,t_{s+1}}$ will be removed. □

Lemma 10. *For all $s > 1$, for all $t \in [t_s, t_{s+1})$, $f^{(\mathcal{B}_{s,t})}(\mathbf{x}^{(\mathbf{t})}) \le (n - |\mathcal{B}_{s,t}|) \cdot \tau_s$.*

Proof. Let $t_s = t_{s,1} < t_{s,2} < t_{s,3} < \cdots < t_{s,\ell_s}$ denote the time steps before which $\mathcal{B}_s$ changes during phase s. For simplicity, we abuse notation and use $\mathcal{B}_{s,j}$ instead of $\mathcal{B}_{s,t_{s,j}}$. Clearly $\mathcal{B}_{s,1} \supseteq$

$\mathcal{B}_{s,2} \ldots \supseteq \mathcal{B}_{s,\ell_s}$, because we only remove nodes from $\mathcal{B}_s$ once it is set. Fix $s > 1$. We prove this lemma by induction on $r \in \{1, 2, \ldots, \ell_s\}$. For the basis, let $r = 1$. Clearly, by the way the algorithm sets $\mathcal{B}_s$ before time $t_{s,1}$, all nodes with weight $\geq \tau_s$ are in $\mathcal{B}_s$, and therefore every node $i \notin \mathcal{B}_s$ has weight at most τ_s, so the lemma follows. Now, assume that the lemma is true for every $t \leq t_{s,r}$, we show that the lemma holds for every $t \leq t_{s,r+1}$. If $t \in [t_{s,r}, t_{s,r+1})$, then $\mathcal{B}_{s,t} = \mathcal{B}_{s,t_{s,r}}$, and we have:

$$f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t})}) \leq f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t_{s,r}})}) \leq \left(n - \left|\mathcal{B}_{s,t_{s,r}}\right|\right) \cdot \tau_s = (n - |\mathcal{B}_{s,t}|) \cdot \tau_s.$$

The first inequality holds because balancing operations from time $t_{s,r}$ to time t only reduce the value of $f^{(\mathcal{B}_s)}$, and the second inequality holds by the induction hypothesis.

Just before iteration $t = t_{s,r+1}$, the set $\mathcal{B}_s$ changes, and one or more nodes are removed from it. However, every removed node has weight at most τ_s, and its removal does not change the weights of the other nodes in $[n] \setminus \mathcal{B}_s$. Therefore, if k nodes are removed from $\mathcal{B}_s$,

$$f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t_{s,r+1}})}) \leq \left(n - \left|\mathcal{B}_{s,t_{s,r}}\right|\right) \cdot \tau_s + k \cdot \tau_s = \left(n - \left|\mathcal{B}_{s,t_{s,r+1}}\right|\right) \cdot \tau_s.$$

This completes the proof. □

Corollary 5. *For all $s > 1$, $f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t_{s+1}})}) \leq \frac{1}{4n^2} \cdot f^{(\mathcal{B}_{s-1})}(\mathbf{x}^{(\mathbf{t_s})})$. If $s > 2$, then $\tau_s \leq \frac{\tau_{s-1}}{4n^2}$.*

Proof. Notice that

$$f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t_{s+1}})}) \leq n \cdot \tau_s = \frac{1}{4n^2} \cdot f^{(\mathcal{B}_{s-1})}(\mathbf{x}^{(\mathbf{t_s})}),$$

where the inequality follows from Lemma 10, and the equation follows from line 13 of StrictBalance. This proves the first assertion. As for the second assertion, notice that if $s > 2$ then $s-1 > 1$, so using line 13 of StrictBalance and Lemma 10 again,

$$\tau_s = \frac{1}{4n^3} \cdot f^{(\mathcal{B}_{s-1})}(\mathbf{x}^{(\mathbf{t_s})}) \leq \frac{1}{4n^3} \cdot n\tau_{s-1} = \frac{1}{4n^2} \cdot \tau_{s-1},$$

as stipulated. □

Lemma 11. *For every phase $s > 1$, for every $t \geq t_s$, all the nodes in $\mathcal{B}_{s,t}$ have weight $\geq \tau_s/2$ and are ϵ-balanced at time t.*

Proof. Fix $s > 1$ and let $i \in \mathcal{B}_{s,t}$. Without loss of generality $i \notin \mathcal{B}_{s-1,t}$, otherwise we can replace s with $s-1$. (Recall that $\mathcal{B}_1 = \emptyset$ at all times.) Also note that it must be the case that $i \in \mathcal{B}_{s,t_s}$, because $\mathcal{B}_s$ does not accumulate additional nodes after being created. If $t \in [t_s, t_{s+1}]$, then lines 13-14 and 8-9 of StrictBalance guarantee that if $i \in \mathcal{B}_{s,t} \setminus \mathcal{B}_{s-1,t}$, then its weight at time t is at least τ_s.

Otherwise, consider $t > t_{s+1}$ and let $s' > s$ be the phase containing t. Consider a phase $j > s$. By Lemma 10 the total weight of $f^{(\mathcal{B}_j)}$ during phase j is at most $n\tau_j$, and $f^{(\mathcal{B}_j)}$ never drops below 0. So, the total weight that a node $i \in \mathcal{B}_j$ can lose (which is at most the total weight that $f^{(\mathcal{B}_j)}$ can lose) is at most $n\tau_j$. By Corollary 5, for every $j > s$, $\tau_{j+1} \leq \frac{\tau_j}{4n^2}$. Now, suppose that t is an iteration in phase $s' > s$. Then, the weight of i at time t is at least

$$\tau_s - \sum_{j=s+1}^{s'} n\tau_j \geq \tau_s \cdot \left(1 - n \cdot \sum_{k=1}^{s'-s} (2n)^{-2k}\right) \geq \frac{\tau_s}{2}.$$

Thus we have established that at any time $t \geq t_s$, if $i \in \mathcal{B}_{s,t}$ then its weight is at least $\frac{\tau_s}{2} = \frac{1}{8n^3} f^{(\mathcal{B}_{s-1})}(\mathbf{x}^{(\mathbf{t_s})})$. By line 4 of StrictBalance, $\|\nabla f^{(\mathcal{B}_{s-1,t_s})}(\mathbf{x}^{(\mathbf{t_s})})\|_1 \leq \epsilon' \cdot f^{(\mathcal{B}_{s-1,t_s})}(\mathbf{x}^{(\mathbf{t_s})})$. By Lemma 9, $\mathcal{B}_{s-1}$ does not change in the interval $[t_s, t]$. Therefore, we conclude from Lemma 8 that i is ϵ-balanced at time t. □

Lemma 12. *Suppose that $t < t'$ satisfies $[t, t') \subseteq [t_s, t_{s+1})$, and furthermore, during the iterations in the interval $[t, t')$ the set $\mathcal{B}_s$ does not change (it could change after balancing step $t' - 1$). Then, the length of the interval*

$$t' - t = O\left(\epsilon^{-2} n^7 \log(wn/\epsilon)\right).$$

Proof. Rename the nodes so that $\mathcal{B}_{s,t} = \mathcal{B}_{s,t'-1} = \{p, p+1, \ldots, n\}$. The assumption that $\mathcal{B}_s$ does not change during the interval $[t, t')$ means that the weights of all the nodes $p, p+1, \ldots, n$ remain at least τ_s for the duration of this interval. During the interval $[t, t')$, the graph $G^{(\mathcal{B}_s)}$ (which remains fixed) is obtained by contracting the nodes $p, p+1, \ldots, n$ in G_A. So $G^{(\mathcal{B}_s)}$ has p nodes $1, 2, \ldots, p-1, p$, where the last node p is the contracted node. In each iteration in the interval $[t, t')$, one of the nodes $1, 2, \ldots, p-1$ is balanced. Consider some time step $t'' \in [t, t')$, and let I_i and O_i, respectively, denote the current sums of weights of the arcs of $G^{(\mathcal{B}_s)}$ into and out of node

i, respectively. Let $j \in [p-1]$ be the node that maximizes $\frac{(I_j-O_j)^2}{I_j+O_j}$. We have

$$\begin{aligned}
f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')}) - f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}''+\mathbf{1})}) &= \max_{i \in [p-1]} \left(\sqrt{I_i} - \sqrt{O_i}\right)^2 \geq \left(\sqrt{I_j} - \sqrt{O_j}\right)^2 \geq \frac{(I_j - O_j)^2}{2(I_j + O_j)} \\
&\geq \frac{\sum_{i=1}^{p-1}(I_i - O_i)^2}{2\sum_{i=1}^{p-1}(I_i + O_i)} \geq \frac{\left(\sum_{i=1}^{p-1}|I_i - O_i|\right)^2}{2n\sum_{i=1}^{p}(I_i + O_i)} \geq \frac{\left(\sum_{i=1}^{p}|I_i - O_i|\right)^2}{8n\sum_{i=1}^{p}(I_i + O_i)} \\
&= \frac{1}{16n} \cdot \frac{\|\nabla f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')})\|_1^2}{f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')})}. \qquad (3.4)
\end{aligned}$$

The first equation follows from the choice of i in line 5 StrictBalance, and Lemma 2. The third inequality follows from an averaging argument and the choice of j. The fourth inequality uses Cauchy-Schwarz. The last inequality holds because $\sum_{i=1}^{p}(I_i - O_i) = 0$, so $|I_p - O_p| = \left|\sum_{i=1}^{p-1}(I_i - O_i)\right| \leq \sum_{i=1}^{p-1}|I_i - O_i|$, and therefore $\sum_{i=1}^{p}|I_i - O_i| \leq 2\sum_{i=1}^{p-1}|I_i - O_i|$.

Since the interval $[t, t')$ is contained in phase s, the stopping condition for the phase does not hold, so

$$\frac{\|\nabla f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')})\|_1}{f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')})} > \epsilon' = \frac{\epsilon^2}{64n^4}.$$

Therefore,

$$\begin{aligned}
f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')}) - f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}''+\mathbf{1})}) &\geq \frac{1}{16n} \cdot \frac{\|\nabla f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')}\|_1^2}{f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')})} \\
&> \frac{\epsilon'}{16n} \cdot \|\nabla f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')})\|_1 \\
&\geq \frac{\epsilon'}{8n^2} \cdot (f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*)),
\end{aligned}$$

where the last inequality follows from Lemma 2. Rearranging the terms gives

$$f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}''+\mathbf{1})}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*) \leq \left(1 - \frac{\epsilon'}{8n^2}\right) \cdot (f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*)).$$

Iterating for T step yields

$$f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}+\mathbf{T})}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*) \leq \left(1 - \frac{\epsilon'}{8n^2}\right)^T \cdot (f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t})}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*)).$$

Now by Corollary 4, we have that $f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t})}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*) \leq f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t})}) \leq f(\mathbf{x}^{(\mathbf{t})}) \leq \sum_{i,j=1}^{n} a_{ij}$, and for all t'', $f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'')}) \geq \frac{1}{w^n}\sum_{i,j=1}^{n} a_{ij}$. Therefore, if $t' - t \geq \frac{8n^2}{\epsilon'} \cdot \ln(16nw^n/(\epsilon')^2) + 1$, then

$$f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'-\mathbf{1})}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*) \leq \left(\frac{\epsilon'}{4\sqrt{n}}\right)^2 \cdot \frac{1}{w^n} \cdot \sum_{i,j=1}^{n} a_{ij} \leq \left(\frac{\epsilon'}{4\sqrt{n}}\right)^2 \cdot f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t}'-\mathbf{1})}).$$

Therefore,

$$\frac{1}{16n}\cdot\frac{\|\nabla f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t'-1})})\|_1^2}{(f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t'-1})}))^2} \le \frac{f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t'-1})}) - f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t'})})}{f^{(\mathcal{B})}(\mathbf{x}^{(\mathbf{t'-1})})} \le \frac{f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t'-1})}) - f^{(\mathcal{B}_s)}(\mathbf{x}^*)}{f^{(\mathcal{B})}(\mathbf{x}^{(\mathbf{t'-1})})} \le \left(\frac{\epsilon'}{4\sqrt{n}}\right)^2,$$

where the first inequality follows from (3.4), and the second inequality holds because $f^{(\mathcal{B}_s)}(\mathbf{x}^*) \le f^{(\mathcal{B}_s)}(\mathbf{x}^{(\mathbf{t'-1})})$. We get that $\frac{\|\nabla f^{(\mathcal{B})}(\mathbf{x}^{(\mathbf{t'-1})})\|_1}{f^{(\mathcal{B})}(\mathbf{x}^{(\mathbf{t'-1})})} \le \epsilon'$, in contradiction to our assumption that the phase does not end before the start of iteration t'. □

Corollary 6. *In any phase, the number of balancing steps is at most $O\left(\epsilon^{-2}n^8\log(wn/\epsilon)\right)$.*

Proof. In the beginning of phase s the set $\mathcal{B}_s$ contains at most $n-1$ nodes. Partition the phase into intervals $[t, t')$ where $\mathcal{B}_s$ does not change during an interval, but does change between intervals. By Lemma 12, each interval consists of at most $O\left(\epsilon^{-2}n^7\log(wn/\epsilon)\right)$ balancing steps. Since nodes that are removed from $\mathcal{B}_s$ between intervals are never returned to $\mathcal{B}_s$, the number of such intervals is at most $n-1$. Hence, the total number of balancing steps in the phase is at most $O\left(\epsilon^{-2}n^8\log(wn/\epsilon)\right)$. □

Lemma 13. *The total number of phases of the algorithm is $O(n\log w/\log n)$.*

Proof. Let $s > 2$ be a phase of the algorithm and $t \in [t_s, t_{s+1})$. By Corollary 4, $f^{(\mathcal{B}_{s,t})}(\mathbf{x}^{(\mathbf{t})}) \ge \frac{1}{w^n}\cdot\sum_{ij} a_{ij}$. On the other hand, by Lemma 10 and Corollary 5, $\tau_s \le \frac{1}{(4n^2)^{s-2}}\cdot\tau_2 \le \frac{1}{(4n^2)^{s-2}}\cdot\sum_{ij} a_{ij}$, and $f^{(\mathcal{B}_{s,t})}(\mathbf{x}^{(\mathbf{t})}) \le n\tau_s$. Combining these gives $\frac{1}{w^n}\cdot\sum_{ij} a_{ij} \le n\tau_s \le \frac{n}{(4n^2)^{s-2}}\cdot\sum_{ij} a_{ij}$ which implies that $s \le \frac{\log(nw^n)}{\log(4n^2)} + 2$. □

We are now ready to prove Theorem 5.

Proof of Theorem 5. By Lemma 13, for some $s = O(n\log w/\log n)$, StrictBalance terminates, so $\mathcal{B}_{s,t_s} = [n]$. By Corollary 6, the number of balancing steps in a phase is at most $O\left(\epsilon^{-2}n^8\log(wn/\epsilon)\right)$. Therefore, the total number of balancing steps is at most $O\left(\epsilon^{-2}n^9\log(wn/\epsilon)\log w/\log n\right)$. These balancing steps require at most $O\left(\epsilon^{-2}n^{10}\log(wn/\epsilon)\log w/\log n\right)$ arithmetic operations, and as in the proof of Theorem 2 we can show that working with $O(n\log(w/\epsilon))$ bits of precision is enough. When the algorithm terminates at time t_s, all the nodes are in $\mathcal{B}_{s,t_s}$, and by Lemma 11 they are all ϵ-balanced, so the matrix is strictly ϵ-balanced. □

CHAPTER 4

A Lower Bound on the Rate of Convergence

In the previous chapters we have shown various upper bounds on the rate of convergence of Osborne's iteration for L_p for finite p. In all these upper bounds the dependency on $1/\epsilon$ is polynomial while in a similar upper bound obtained by Schulman and Sinclair [12] for L_∞ the dependency is $\log(1/\epsilon)$. This raises the question whether our analysis could be improved to show a better dependency on $1/\epsilon$. We answer this question by proving a lower bound of $1/\sqrt{\epsilon}$ on the rate of convergence. Therefore, proving that $\log(1/\epsilon)$ dependency is impossible and the dependency of our upper bounds on the $1/\epsilon$ is in the right ballpark.

Theorem 6. *There are matrices for which all variants of the Osborne's iterative algorithm (i.e., regardless of the order of indices chosen to balance) require $\Omega(1/\sqrt{\epsilon})$ iterations to ϵ-balance the matrix.*

Before proving this theorem, we present the claimed construction. Let A be the following 4×4 matrix, and let A^* denote the corresponding fully balanced matrix.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & \beta+\epsilon & 0 \\ 0 & \epsilon & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad A^* = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & \sqrt{\epsilon(\beta+\epsilon)} & 0 \\ 0 & \sqrt{\epsilon(\beta+\epsilon)} & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Here $\epsilon > 0$ is arbitrarily small, and $\beta = 100\epsilon$. It's easy to see that $A^* = D^* A D^{*-1}$ where $D^* = \texttt{diag}\left(1, 1, \sqrt{\frac{\beta+\epsilon}{\epsilon}}, \sqrt{\frac{\beta+\epsilon}{\epsilon}}\right)$. To prove Theorem 6, we show that balancing A to the relative error of ϵ requires $\Omega(1/\sqrt{\epsilon})$ iterations, regardless of the order of balancing operations. Notice that in order to fully balance A, we simply need to replace a_{23} and a_{32} by their geometric mean. We measure the rate of convergence using the ratio a_{32}/a_{23}. This ratio is initially $\frac{\epsilon}{\beta+\epsilon} = \frac{1}{101}$. When the matrix is fully balanced, the ratio becomes 1. We show that this ratio increases by a small factor in each iteration, and that it has to increase sufficiently for the matrix to be ϵ-balanced. This is summarized in the following two lemmas.

Lemma 14 (change in ratio)**.** *For every iteration t,* $\dfrac{a_{32}^{(t+1)}}{a_{23}^{(t+1)}} \leq \left(\dfrac{1+7\sqrt{\beta}}{1+\epsilon}\right) \cdot \dfrac{a_{32}^{(t)}}{a_{23}^{(t)}}$.

Lemma 15 (stopping condition)**.** *If matrix $A^{(t)}$ is ϵ-balanced, then* $\dfrac{a_{32}^{(t)}}{a_{23}^{(t)}} > \dfrac{1}{100}$.

Before proving the two lemmas we show how they lead to the proof of Theorem 6.

Proof of Theorem 6. By Lemma 14, $\frac{a_{32}^{(t+1)}}{a_{23}^{(t+1)}} \leq \left(\frac{1+7\sqrt{\beta}}{1+\epsilon}\right)^t \cdot \frac{a_{32}}{a_{23}} = \left(\frac{1+7\sqrt{\beta}}{1+\epsilon}\right)^t \cdot \frac{\epsilon}{\beta+\epsilon}$. By Lemma 15, if $A^{(t+1)}$ is ϵ-balanced, then $\frac{1}{100} < \frac{a_{32}^{(t+1)}}{a_{23}^{(t+1)}} \leq \left(\frac{1+7\sqrt{\beta}}{1+\epsilon}\right)^t \cdot \frac{\epsilon}{\beta+\epsilon} \leq \left(1+7\sqrt{\beta}\right)^t \cdot \frac{\epsilon}{\beta+\epsilon}$. Using $\beta = 100\epsilon$, we get the condition that $(1+7\sqrt{\beta})^t > \frac{101}{100}$, which implies that $t = \Omega(1/\sqrt{\epsilon})$. □

Proof of Lemma 14. Using the notation we defined earlier, we have that $f(\mathbf{x}^{(\mathbf{1})}) = \sum_{i,j=1}^4 a_{ij} = 4 + 2\epsilon + \beta$ and $f(\mathbf{x}^*) = \sum_{i,j=1}^4 a_{ij}^* = 4 + 2\sqrt{\epsilon(\beta+\epsilon)}$, so $f(\mathbf{x}^{(\mathbf{1})}) - f(\mathbf{x}^*) < \beta$. We observe that at each iteration t, $a_{12}^{(t)}a_{21}^{(t)} = a_{34}^{(t)}a_{43}^{(t)} = 1$ and $a_{23}^{(t)}a_{32}^{(t)} = \epsilon(\beta+\epsilon)$ because the product of weights of arcs on any cycle in G_A is preserved (for instance, arcs $(1,2)$ and $(2,1)$ form a cycle and initially $a_{12}a_{21} = 1$).

The ratio $a_{32}^{(t)}/a_{23}^{(t)}$ is only affected in iterations that balance index 2 or 3. Let's assume a balancing operation at index 2, a similar analysis applies to balancing at index 3. By balancing at index 2 at time t we have

$$\frac{a_{32}^{(t+1)}}{a_{32}^{(t)}} = \frac{a_{23}^{(t)}}{a_{23}^{(t+1)}} = \sqrt{\frac{a_{21}^{(t)} + a_{23}^{(t)}}{a_{12}^{(t)} + a_{32}^{(t)}}}. \tag{4.1}$$

Thus, to prove Lemma 14, it suffices to show that

$$\frac{a_{32}^{(t+1)}}{a_{32}^{(t)}} \cdot \frac{a_{23}^{(t)}}{a_{23}^{(t+1)}} = \frac{a_{21}^{(t)} + a_{23}^{(t)}}{a_{12}^{(t)} + a_{32}^{(t)}} \leq \frac{1+7\sqrt{\beta}}{1+\epsilon}. \tag{4.2}$$

By our previous observation, $a_{12}^{(t)}a_{21}^{(t)} = 1$, so if $a_{21}^{(t)} = y$, then $a_{12}^{(t)} = 1/y$. Similarly $a_{23}^{(t)}a_{32}^{(t)} = \epsilon(\beta+\epsilon)$ implies that there exists z such that $a_{23}^{(t)} = (\beta+\epsilon)z$ and $a_{32}^{(t)} = \epsilon/z$. Therefore:

$$\frac{a_{21}^{(t)} + a_{23}^{(t)}}{a_{12}^{(t)} + a_{32}^{(t)}} = \frac{y + (\beta+\epsilon)z}{(1/y) + (\epsilon/z)} \tag{4.3}$$

We bound the right hand side of Equation 4.3 by proving upper bounds on y and z. We first show that $y < 1 + 2\sqrt{\beta}$. To see this notice that on the one hand, $f(\mathbf{x^{(t)}}) = \sum_{i,j=1}^{4} a_{ij}^{(t)} = a_{12}^{(t)} + a_{21}^{(t)} + a_{23}^{(t)} + a_{32}^{(t)} + a_{34}^{(t)} + a_{43}^{(t)} \geq y + \frac{1}{y} + 2\sqrt{\epsilon(\beta+\epsilon)} + 2$, where we used $a_{34}^{(t)} + a_{43}^{(t)} \geq 2$ and $a_{23}^{(t)} + a_{32}^{(t)} \geq 2\sqrt{\epsilon(\beta+\epsilon)}$, both implied by the arithmetic-geometric mean inequality. On the other hand, $f(\mathbf{x^{(t)}}) \leq f(\mathbf{x^{(1)}}) \leq f(\mathbf{x}^*) + \beta = 4 + 2\sqrt{\epsilon(\beta+\epsilon)} + \beta$. Combining this upper bound on $f(\mathbf{x^{(t)}})$ with the latter lower bound on $f(\mathbf{x^{(t)}})$, we have $y + (1/y) - 2 \leq \beta$. For sufficiently small ϵ, the last inequality implies, in particular, that $y < 2$. Thus, we have $(y-1)^2 \leq y\beta < 2\beta$, and this implies that $y < 1 + 2\sqrt{\beta}$.

Next we show that $z \leq 1$. Assume for contradiction that $z > 1$. By the arithmetic-geometric mean inequality $a_{12}^{(t)} + a_{21}^{(t)} \geq 2$ and $a_{34}^{(t)} + a_{43}^{(t)} \geq 2$. Thus,

$$f(\mathbf{x^{(t)}}) = \sum_{i,j=1}^{4} a_{ij}^{(t)} \geq 2 + (\beta+\epsilon)z + \frac{\epsilon}{z} + 2 = 4 + \beta z + \epsilon\left(z + \frac{1}{z}\right) > 4 + \beta + 2\epsilon = f(\mathbf{x^{(1)}}),$$

where the last inequality follows because $z > 1$, and $z + 1/z > 2$. But this is a contradiction, because each balancing iteration reduces the value of F, so $f(\mathbf{x^{(t)}}) \leq f(\mathbf{x^{(1)}})$.

We can now bound $(a_{21}^{(t)} + a_{23}^{(t)})/(a_{12}^{(t)} + a_{32}^{(t)})$. By Equation 4.3, and using our bounds for y and z, we have,

$$\begin{aligned}
\frac{a_{21}^{(t)} + a_{23}^{(t)}}{a_{12}^{(t)} + a_{32}^{(t)}} &= \frac{y + (\beta+\epsilon)z}{(1/y) + (\epsilon/z)} \\
&\leq \frac{(1+2\sqrt{\beta}) + (\beta+\epsilon)}{\dfrac{1}{1+2\sqrt{\beta}} + \epsilon} \\
&\leq \frac{1+4\sqrt{\beta}}{\dfrac{1}{1+2\sqrt{\beta}} + \dfrac{\epsilon}{1+2\sqrt{\beta}}} \\
&\leq \frac{1+7\sqrt{\beta}}{1+\epsilon}.
\end{aligned}$$

The last line uses the fact that $\sqrt{\beta} \gg \beta = 100\epsilon \geq \epsilon$, which holds if ϵ is sufficiently small. □

Proof of Lemma 15. Let $t-1$ be the last iteration before an ϵ-balanced matrix is obtained. We argued that there is $z \leq 1$ such that $a_{23}^{(t)} = (\beta+\epsilon)z$ and $a_{32}^{(t)} = \epsilon/z$. Assume for the sake of contradiction that $a_{32}^{(t)}/a_{23}^{(t)} < 1/100$. This implies that $(\epsilon/z)/((\beta+\epsilon)z) < 1/100$, and thus $z^2 > 100/101$. So, we get

$$\begin{aligned}
f(\mathbf{x^{(t)}}) - f(\mathbf{x}^*) &\geq a_{23}^{(t)} + a_{32}^{(t)} - 2\sqrt{a_{23}^{(t)} a_{32}^{(t)}} \\
&= \left(\sqrt{a_{23}^{(t)}} - \sqrt{a_{32}^{(t)}}\right)^2 && (4.4) \\
&\geq a_{23}^{(t)} \left(1 - \sqrt{\frac{1}{100}}\right)^2 && (4.5) \\
&= 0.81 \cdot (\beta + \epsilon) z && (4.6) \\
&\geq 0.81 \cdot (\beta + \epsilon) \cdot \sqrt{\frac{100}{101}} \geq 81 \cdot \epsilon. && (4.7)
\end{aligned}$$

By Lemma 2, the left hand side the of above can be bounded as follows.

$$f(\mathbf{x^{(t)}}) - f(\mathbf{x}^*) \leq n \|\nabla f(\mathbf{x^{(t)}})\|_1 \leq n^2 \|\nabla f(\mathbf{x^{(t)}})\|_2 \tag{4.8}$$

Note that for sufficiently small ϵ, $f(\mathbf{x^{(t)}}) \leq f(\mathbf{x^{(1)}}) \leq 5$. Combining Equations 4.6 and 4.8, and using $n = 4$ and $f(\mathbf{x^{(t)}}) \leq 5$, we get that

$$\frac{\|\nabla f(\mathbf{x^{(t)}})\|_2}{f(\mathbf{x^{(t)}})} > \frac{81}{80} \cdot \epsilon > \epsilon. \tag{4.9}$$

By Equation 1.3, this contradicts our assumption that $t - 1$ is the last iteration. □

REFERENCES

[1] *EISPACK implementation*, *http://www.netlib.org/eispack/balanc.f.*

[2] T.-Y. Chen, *Balancing sparse matrices for computing eigenvalues*, Master's thesis, UC Berkeley, May 1998.

[3] B. C. Eaves, A. J. Hoffman, U. G. Rothblum, and H. Schneider, *Line-sum-symmetric scalings of square nonnegative matrices*, Mathematical Programming Essays in Honor of George B. Dantzig Part II, Springer, 1985, pp. 124–141.

[4] J. Grad, *Matrix balancing*, The Computer Journal **14** (1971), no. 3, 280–284.

[5] D. J. Hartfiel, *Concerning diagonal similarity of irreducible matrices*, Proceedings of the American Mathematical Society, 1971, pp. 419–425.

[6] B. Kalantari, L. Khachiyan, and A. Shokoufandeh, *On the complexity of matrix balancing*, SIAM Journal on Matrix Analysis and Applications **118** (1997), no. 2, 450–463.

[7] D. Kressner, *Numerical methods for general and structured eigenvalue problems*, Princeton University Press, 2005.

[8] E. E. Osborne, *On pre-conditioning of matrices*, Journal of the ACM (JACM) **7** (1960), no. 4, 338–345.

[9] B. N. Parlett and C. Reinsch, *Balancing a matrix for calculation of eigenvalues and eigenvectors*, Numerische Mathematik **13** (1969), no. 4, 293–304.

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes: The art of scientific computing, 3rd edition*, Cambridge University Press, 2007.

[11] H. Schneider and M. H. Schneider, *Max-balancing weighted directed graphs and matrix scaling*, Mathematics of Operations Research **16** (1991), no. 1, 208–222.

[12] L. J. Schulman and A. Sinclair, *Analysis of a classical matrix preconditioning algorithm*, Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, 2015, pp. 831–840.

[13] L. N. Trefethen and M. Embree, *Spectra and pseudospectra: The behavior of nonnormal matrices and operators*, Springer, 2005.

[14] N. E. Young, R. E. Tarjan, and J. B. Orlin, *Faster parametric shortest path and minimum-balance algorithms*, Networks **21** (1991), no. 2, 205–221.