Quantifying and Reducing Imbalance in Networks

Yoosof Mashayekhi
Bo Kang
yoosof.mashayekhi@ugent.be
bo.kang@ugent.be
Ghent University
Ghent, Belgium

Jefrey Lijffijt
Tijl De Bie
jefrey.lijffijt@ugent.be
tijl.debie@ugent.be
Ghent University
Ghent, Belgium

ABSTRACT

Real-world data can often be represented as a heterogeneous network relating nodes of different types. For example, envision a network of the job market, where nodes may be job seekers, skills, and jobs, and where links to skill nodes could indicate having that skill (if linked to a job seeker) or having the skill as a requirement (for jobs). It can be relevant to consider the *imbalance* in such a network between the nodes of different types. In the example, this imbalance could correspond to the mismatch between supply and demand of jobs due to a mismatch in skills, an effect known as 'friction'. Identifying and reducing such friction is a problem of great economic and societal significance.

We introduce a quantification of the imbalance in a network between two sets of nodes (nodes of different types, attributes, etc.) based on the embedding of a network, i.e., a real-valued vector space representation of the network nodes. Moreover, we introduce an algorithm named `GraB` (Graph Balancing) which ranks unconnected node pairs according to how well they would reduce the imbalance in a network if an edge were added between them. E.g., in the job network, `GraB` could be used to rank skills that job seekers do not yet have but could strive to acquire, to move them closer in the embedding towards an area where there is an abundance of jobs, and hence to reduce job market imbalance. We evaluated `GraB` on several datasets, including a job market network, and find that `GraB` outperforms baselines in reducing network imbalance.

CCS CONCEPTS

• **Computing methodologies** → *Learning latent representations.*

KEYWORDS

Network Imbalance, Network Embedding, Representation learning

1 INTRODUCTION

Graphs (or networks) are natural models for a wide range of real-world structures [4], arising from e.g., social networks [8], biology (e.g., Protein-Protein interaction networks) [26], and linguistics (e.g., word co-occurrence networks) [5]. Network embedding provides an efficient way to solve graph analytics problems by mapping nodes into a real-valued space, which can later be used as an input feature vector to a machine learning model [11]. Using these vector representations, machine learning methods can be applied on graph datasets to perform graph analysis tasks such as link prediction [12], information diffusion [9], and node classification [25].

An imbalance between two sets of nodes is an undesirable phenomenon in some networks. This paper studies how to quantify network imbalance using the embedding of a network and proposes a method to reduce network imbalance by adding a limited number of links to the network.

Motivation: There are many networks for which it is be desirable to minimize the imbalance between specific sets of nodes. Let us consider an example of a job market network with three sets of nodes job vacancies, job seekers, and skills, and where job vacancies are connected to the skills they require, and job seekers are connected to the skills they have and possibly to the job vacancies they have shown an interest in.

Imagine there are many Python developers seeking a job, and few vacancies requiring Python programming, while there are many vacancies requiring Java programming but few Java developers. As a result, Java jobs would remain unfilled and many Python programmers would remain unemployed. With an ever faster evolving job market, such imbalances are increasingly common and serious, harming job market efficiency and ultimately the economy. Thus, quantifying such imbalances would provide policy makers with an objective picture of the current state of affairs.

Moreover, the ability to quantify imbalance also opens up the possibility of trying to reduce it through targeted interventions and incentives. While policy makers may not be able to influence employers to shift their requirements, they can provide courses and training material for specific worker profiles lacking sought-after skills, to shift their area of expertise and better meet the demand of the job market. In network terms, it is equivalent to adding a certain number of links connecting job seekers (let us call this the set of *source nodes*) to skills (*auxiliary nodes*), to reduce the imbalance between job seekers (*source nodes*) and job vacancies (*target nodes*).

Our approach: In a job market network, a matching with lowest cost—where cost could be defined as the required training time of employees in the company, or the effort a job seeker has to make to be suited for a job—between job seekers and job vacancies appears to be the ideal situation. However, matching job seekers to connected jobs only is not always desirable, because a link may only indicate a job seeker's expressed interest in a job. Yet, we could let job seekers be matched with jobs even if they are not connected.

Formally. we denote an undirected network by $G = (V, E)$, where V and E are the sets of nodes and links respectively. Moreover, we define three sets of nodes, namely *source nodes* S (e.g. job seekers in the job market network), *target nodes* T (e.g. job vacancies), and *auxiliary nodes* U (e.g. skills). Sets S, T, and U are three disjoint subsets of V. Given such a network with sets S and T, we let every node in S be matched with every node in T.

RecSys in HR 2021, October 1, 2021, Amsterdam, the Netherlands

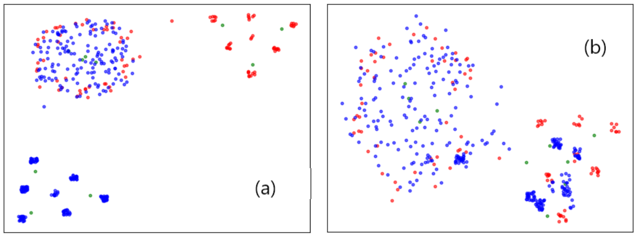

Figure 1: A sample network. Blue nodes denote source nodes, red nodes denote target nodes and green nodes denote auxiliary nodes. (a) A 2D embedding of the original network. (b) A 2D embedding of the network after adding 200 links between source and auxiliary nodes using `GraB`, to reduce the imbalance between source and target nodes.

Hence, in effect we model a complete bipartite network G' from node sets S and T. We define the *imbalance* between S and T in G as the cost of a so-called optimal fractional perfect b-matching (see, e.g., [1]) in G'. Specifying the fractional perfect b-matching requires the definition of the *cost* of matching any pair of nodes from S and T in G'. As the (inverse of the) distance between a pair of nodes' embeddings in a network embedding usually represents some type of affinity between these nodes, we propose to let this cost be based on the embedding distance of nodes in G. More details are presented in Section 2.2.

Next, we define the problem of adding a limited number of links between sets S and U—*in the original graph* G—, to *reduce the imbalance* between sets S and T. Adding links will change the embedding of the network G and thus modify the cost of matching node pairs in G'. We propose a method called Graph Balancing (`GraB`) to tackle this problem, which is based on a local approximation of the imbalance that we may compute analytically, thus providing the necessary scalability.

Example: To understand the relevance of the embedding, consider the sample network shown in Figure 1a. The figure shows three clusters of nodes. In the top-left cluster, source and target nodes are mixed, while the bottom-left cluster only contains nodes in S while the top-right cluster only contains nodes in T. Our goal is to quantify which links between S and U (the green nodes here) would reduce the imbalance between S and T. Figure 1b shows the network embedding after adding the top 200 links from `GraB` to the network. Now, S and T are well-mixed in the network.

Related work: Previous studies on imbalance in supply and demand on the job market mostly focus on the factors influencing the imbalance (such as retirement, salary, etc.) and do not consider the imbalance between nodes individually (different jobs require people with specific skills) [27, 29]. The literature on graph matching is related to our work, as we also define the imbalance in a network between two sets of nodes using the cost of a matching between them. However, the focus in this paper is not on the computational problem of identifying this matching, we simply use the cost of the optimal fractional perfect b-matching as the imbalance measure. These two relations are further discussed in Section 5.

The **main contributions** of this paper are:

- We *define the imbalance* between two sets of nodes, S and T, in a network and propose the *measure* $\boldsymbol{\psi}(\boldsymbol{D}, S, T)$ *for quantifying it*, where we compute the cost of matching node pairs using the embedding distances $\boldsymbol{D}$.
- We introduce the novel problem of *reducing the imbalance* in a network by adding a limited number of links.
- We also propose a novel generic method, *Graph Balancing (*`GraB`*)*, to optimally select those links. Because this is computationally challenging, we introduce a link utility that uses a local imbalance measure as a proxy and employ a greedy algorithm to select the links.
- To better understand the merits of `GraB`, we propose two *baselines* (a naive and a more intelligent baseline) for the novel problem of reducing the imbalance in a network.
- `GraB` is proposed as a generic method, applicable to a wide range of network embedding methods. We also develop a concrete instantiation using *conditional network embedding (CNE)* [16], a state-of-the-art network embedding method.
- We perform several experiments to compare the performance of `GraB` and baselines in reducing the imbalance in a network embedding. The experiments show that `GraB` outperforms the baselines in this task.

Outline. In Section 2, we define and quantify imbalance and formalize the problem. In Section 3, we introduce our method `GraB` to reduce the imbalance in a network using its embedding. In Section 4, we provide an experimental evaluation of `GraB`. In Section 5, we discuss the related work. In Section 6, we conclude and outline avenues for future work.

2 PROBLEM DEFINITION

In this section, we first provide the relevant background and notation. Next, we provide a definition and quantification of imbalance in networks. Finally, we formulate the problem of reducing imbalance by adding links to a network.

2.1 Background

An undirected network is denoted by $G = (V, E)$, where V is the set of $|V| = n$ nodes and $E \subseteq \binom{V}{2}$ is the set of links between nodes. For convenience, we will index the set of nodes by natural numbers, i.e. $V = \{1, \ldots, n\}$. Let $\boldsymbol{A}$ denote the adjacency matrix, and a_{ij} is the element of the adjacency matrix corresponding to the link between node i and node j, i.e. $a_{ij} = 1$ iff $\{i, j\} \in E$. Network embedding methods find a mapping each node $i \in V$ to a d-dimensional real vector $\boldsymbol{x}_i \in \mathbb{R}^d$. For convenience, these may be aggregated in a matrix $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)' \in \mathbb{R}^{n \times d}$. In this paper, we assume there is a given network embedding method to find $\boldsymbol{X}$.

2.2 Network Imbalance

To generically define our proposed notion of imbalance, we develop it first for the concrete example of the job market. There, we define the imbalance as the cost of matching all job seekers to job vacancies. Here, we allow a job seeker and a job to be matched even if they are not connected by a link. Indeed, a link between a job seeker and a job vacancy might mean that the job seeker has applied for the job or has otherwise expressed an interest, not necessarily that they were employed for that job. Moreover, the absence of a link does not imply that the job seeker would not be a good candidate for the job. *It is this property that distinguishes our work from the literature on combinatorial matching problems in graphs.*

However, the skills to which the job seeker and job vacancy are linked, and jobs vacancies they *are* linked to, provide information on whether the job seeker is suited for the job; and the more suited, the smaller the cost of a match between them should be. Hence, the cost of matching a job seeker and a job vacancy should be a function of the network structure, and adding or removing skills to that job seeker or job vacancy should influence the cost of matching them. Hence, the cost could be defined using any model that provides link costs or link probabilities (so the cost of node pairs could be computed based on them) based on the network structure. In this paper, we investigate using the embedding of the network for this, as it is a state-of-the-art approach to summarize the network structure, where proximity between node embeddings reflects the probability that both nodes *should be* linked.

2.2.1 Formal definition of network imbalance. The imbalance can be formalized as follows. We create a complete bipartite network (in which all node pairs between the two sets are linked), and assign equal weights w_i to each node i in a set, such that the sum of weights in two sets are equal. Given a cost defined for each link $\boldsymbol{D} = [d_{ij}]$ (e.g., based on the link probability or distance in the embedding space), we define the imbalance as the cost of the *fractional perfect b-matching* with minimum cost in the bipartite network.

A fractional perfect b-matching $\boldsymbol{F} = [f_{ij}]$ is defined as an assignment of nonnegative real numbers f_{ij} to the links of a network such that the sum of those numbers over links incident to any node i is equal to a specified weight w_i of that node [1]. The cost of F in a undirected network with n nodes is then defined as $\sum_{i=1}^{n}\sum_{j=i+1}^{n} f_{ij}d_{ij}$. We thus define the imbalance in a network as follows:

Definition 1 (Imbalance). *Consider a network $G = (V, E)$, two disjoint non-trivial subsets of its nodes referred to as the source nodes $\emptyset \subset S \subset V$ and the target nodes $\emptyset \subset T \subset V$ with $S \cap T = \emptyset$, and the costs d_{ij} of matching the pairs of nodes i and j for all $i \in S$ and $j \in T$, arranged in a matrix $\boldsymbol{D} = [d_{ij}]$. Moreover, consider the complete edge-weighted and node-weighted bipartite network $G' = (V' = S \cup T, E' = S \times T)$, with weight of edge $\{i, j\}$ with $i \in S$ and $j \in T$ equal to d_{ij}, and weight of node $i \in S$ equal to $w_i = \frac{1}{|S|}$ and weight of node $j \in T$ equal to $w_j = \frac{1}{|T|}$. Then the imbalance between nodes in sets S and T, denoted as $\boldsymbol{\psi}(\boldsymbol{D}, S, T)$, is defined as the cost of the minimum cost fractional perfect b-matching in G'.*

Computing the imbalance defined in this way can be done by solving a linear programming problem for finding the matching $\boldsymbol{F} = [f_{ij}]$ in G' that minimizes the overall cost:

$$
\begin{aligned}
C = &\sum_{i\in S}\sum_{j\in T} f_{ij}d_{ij},\\
\text{s.t.}\quad & f_{ij} \geq 0 \quad \forall (i,j) \in S \times T,\\
&\sum_{j\in T} f_{ij} = w_i \quad \forall i \in S, \quad \sum_{i\in S} f_{ij} = w_j \quad \forall j \in T.
\end{aligned}
$$

The imbalance measure ψ is defined as the minimum cost of the optimal matching. (The matching itself is not of interest to us for the purposes of this paper.)

2.2.2 The matching cost, and a relation to the earth mover's distance. While the cost for each edge could be defined in several ways, network embeddings arguably offer a natural way to define them: our approach is to use the distance of nodes in the network embedding of G as the matching cost d_{ij} of each node pair in G'. For embedding methods modeling first-order similarities in networks, these distances are directly related to the link probability between nodes. Moreover, the embedding of a node aggregates all relevant information about the network structure in relation to the node.

Interestingly, with this matching cost, the proposed definition of the imbalance is equivalent with the Earth Mover's Distance (EMD) [23] between the source and target sets in the embedding space. The EMD computes the minimum cost to transform one distribution into another.

Proposition 1 (Network Imbalance based on fractional b–matching and EMD are equivalent). *Consider the two empirical distributions of the node embeddings for S and T in the embedding space. The EMD between these two distributions is equal to the network imbalance measure ψ.*

We refer to Appendix A[1] for a more precise formulation of this proposition and a proof.

2.3 Formulating the problem of reducing network imbalance

As the cost of matching node pairs depends on the structure of the network, it will change by modifying the network. Motivated by the job market example, we propose the operation of adding links as the kind of modification that can be made. We further propose to restrict the problem by allowing links to be added only between nodes from the source set and nodes from an auxiliary set of nodes. This is again motivated by the job market, where we can realistically add new links between skills and job seekers (by training job seekers), but not between job vacancies and skills. More formally, we introduce the following problem:

Problem 1 (Imbalance Reduction). *Given a network $G = (V, E)$ and three mutually disjoint sets of nodes source nodes $\emptyset \subset S \subset V$, target nodes $\emptyset \subset T \subset V$ and auxiliary nodes $\emptyset \subset U \subset V$, the cost of matching each pair of nodes $\boldsymbol{D} = [d_{ij}]$ (that depends on the network structure G), and imbalance measure $\psi(\boldsymbol{D}, S, T)$, find the optimal k links $\mathcal{E}$ connecting nodes from set S with nodes from set U, that reduce the imbalance between the nodes in sets S and T. Formally,*

$$
\underset{\substack{\mathcal{E}\\ \mathcal{E}\subseteq S\times U, \mathcal{E}\cap E=\emptyset}}{argmin} \ \psi(\boldsymbol{D}_{\mathcal{E}}, S, T),
$$

where $D_{\mathcal{E}}$ is computed based on $G_{\mathcal{E}} = (V, E \cup \mathcal{E})$.

In the next section, we introduce GraB to solve this problem.

[1] https://github.com/aida-ugent/GraB/blob/master/GraB_appendix.pdf

3 REDUCING NETWORK IMBALANCE: GRAB

In this section, we introduce GraB, a generic method to solve the imbalance reduction problem, i.e., how to add k links to a network in order to maximally reduce the imbalance in the network, as defined in Definition 1. We first provide a sketch of the solution and then provide more details in the respective sections below.

The exact minimization problem amounts to finding a set of links that jointly minimize the imbalance. This exact approach may be computationally intractable when the number of candidate links is large, since it may require computing the imbalance reduction for every possible set of k links. Besides the vast number of possible sets, to compute the reduction in imbalance we need to re-embed the network and compute the imbalance again. Recomputing the embedding is computationally demanding, and is practically impossible to do even for a moderate number of candidate link sets. Motivated by these two problems, our approach is as follows.

Firstly, rather than re-embedding the network and observing the change in imbalance, *we introduce a proxy measure for the change*, based on an infinitesimal change to any link $\{i, j\}$. We refer to this proxy measure as the *link utility*. The link utility is based on three elements. (1) Since the formulation of ψ is a linear programming problem, the derivative of ψ w.r.t. links cannot be directly computed. Hence, we introduce a local imbalance measure to be used as a proxy for ψ, and quantify the infinitesimal changes in links for that measure. (2) This local imbalance measure is then used to derive a measure of the utility of links. (3) The local imbalance measure relies on the estimation of the density of nodes at any point in the embedding. For this, we employ multivariate Gaussian kernel density estimation. These elements are presented in Section 3.1.

Secondly, the problem that we cannot test the imbalance reduction for all possible sets of links remains. Link utility does not account for any interactions between links on the amount of imbalance in the network. To find a good balance between accuracy and computational tractability, GraB employs a greedy selection strategy using link utility in combination with re-embedding the network every b steps (the batch size). The GraB algorithm implementing this strategy is introduced in Section 3.2.

Finally, we need to select a suitable embedding method. We briefly discuss suitable methods in Section 3.3.

3.1 The Link Utility Measure

In the embedding of a network, there are areas where the set S (or T) is denser than the other one. Given a network with n nodes and their corresponding embeddings $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, the idea behind the proxy measure is that to reduce the imbalance, we should add links connecting source nodes with auxiliary nodes, such that the source nodes are moved to areas with a higher density of target nodes and fewer source nodes. We thus introduce a local imbalance measure to quantify the density imbalance between source and target node sets at any point in the embedding. Moving source nodes to areas with higher local imbalance (more target nodes than source nodes) would reduce the local imbalance in those areas. We define the utility of adding a link using the first order of approximation of this local imbalance measure.

3.1.1 Local Imbalance Quantification. Skipping for a moment how to quantify the density of a set of nodes at a specific point in the embedding, we define the local imbalance measure which evaluates the imbalance between the embedding of two sets of nodes S and T locally at any point $\boldsymbol{x}$ in the embedding space as follows:

Definition 2 (Local Imbalance Measure $\delta_{S,T}$). *Given a network $G = (V, E)$ with the embedding $\boldsymbol{X}$ and two disjoint sets of nodes, source nodes S with $\emptyset \subset S \subset V$ and target nodes T with $\emptyset \subset T \subset V$, denote the density function of the target nodes as estimated based on the embedding $\boldsymbol{X}$ and evaluated at point $\boldsymbol{x}$ by $p_T(\boldsymbol{x}; \boldsymbol{X})$, and let $p_S(\boldsymbol{x}; \boldsymbol{X})$ denote the estimated density function for the source nodes evaluated at $\boldsymbol{x}$. We use the log ratio of the density of the two sets of nodes as local imbalance measure $\delta_{S,T}$ evaluated at point $\boldsymbol{x}$:*

$$\delta_{S,T}(\boldsymbol{x}; \boldsymbol{X}) = \ln\left(\frac{p_T(\boldsymbol{x}; \boldsymbol{X})}{p_S(\boldsymbol{x}; \boldsymbol{X})}\right). \tag{1}$$

If the densities are differentiable and non-zero everywhere, also $\delta_{S,T}$ becomes differentiable and suitable for optimization.

Example: Let us illustrate the idea behind GraB and the local imbalance as a proxy to optimize the imbalance ψ. Figure 2a shows a 2D embedding of a toy network with equal number of source and target nodes. Hence, each source node should be matched with exactly one target node to compute the imbalance ψ. Visually, t_1 should be matched with s_3, since they are close to each other and the cost of matching them is low. Thus, s_1 and s_2 should be matched with t_2 and t_3, although the matching cost (their distance) would be high. GraB can then be used to move source nodes s_1 and s_2 to areas with higher local imbalance δ, which is the area where t_2 and t_3 reside. If we move s_1 and s_2 closer to t_2 and t_3, the matching costs between them would be reduced and the imbalance ψ would be reduced as well. GraB's goal is to find links between source and auxiliary nodes such that adding them would move the source nodes to areas with higher local imbalance δ. GraB uses a link utility measure for this purpose, which will be discussed in the next section. Figure 2b shows the 2D embedding of the network after adding two links $\{s_1, a_3\}$ and $\{s_2, a_3\}$ suggested by GraB, demonstrating that adding the links indeed reduces ψ, from 3.44 to 2.71.

3.1.2 Link Utility. We can now define the utility of a link:

Definition 3 (Link Utility). *The utility of adding a link $\{i, j\}$ for reducing the local imbalance at the embedding $\boldsymbol{x_i}$ of source node $i \in V$ is defined as the rate at which the local imbalance evaluated at $\boldsymbol{x_i}$ changes when increasing a_{ij}, or mathematically: $\frac{\partial \delta_{S,T}(\boldsymbol{x_i};\boldsymbol{X})}{\partial a_{ij}}$.*

Note that two effects of adding the link $\{i, j\}$ are accounted for in this definition: the fact that the embedding $\boldsymbol{x_i}$ of node i will move, possibly to a denser or sparser region, and the fact that the density functions themselves may change. Both effects can be separated by computing the total derivative:

$$\frac{\partial \delta_{S,T}(\boldsymbol{x}; \boldsymbol{X})}{\partial a_{ij}} = \nabla_{\boldsymbol{x}} \delta_{S,T}(\boldsymbol{x}; \boldsymbol{X})^T \frac{\partial \boldsymbol{x}(\boldsymbol{A})}{\partial a_{ij}} \tag{2}$$
$$+ \sum_{r \in S} \nabla_{\boldsymbol{x}_r} \delta_{S,T}(\boldsymbol{x}; \boldsymbol{X})^T \frac{\partial \boldsymbol{x}_r(\boldsymbol{A})}{\partial a_{ij}} + \sum_{r \in T} \nabla_{\boldsymbol{x}_r} \delta_{S,T}(\boldsymbol{x}; \boldsymbol{X})^T \frac{\partial \boldsymbol{x}_r(\boldsymbol{A})}{\partial a_{ij}},$$

where the first term accounts for the change in position $\boldsymbol{x}$ where the densities are measured, and the second and third terms account fo the changes in both estimated densities.

Evaluating all these terms is costly. However, since changing a_{ij} has a direct effect only on the embeddings of nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ [17],

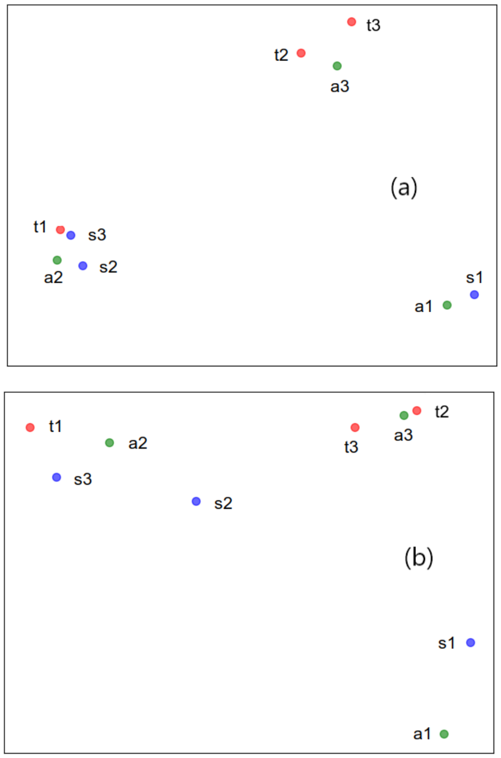

Figure 2: A sample network. Blue nodes denote source nodes, red nodes denote target nodes and green nodes denote auxiliary nodes. (a) A 2D embedding of the original network. Since the number of source and target nodes are the same, s_1 and s_2 should be moved closer to t_2 and t_3 (because no source nodes are close to them. The area near t_2 and t_3 has the highest local imbalance in the embedding space) to reduce the matching cost (imbalance). (b) A 2D embedding of the network after adding 2 links $\{s_1, a_3\}$ and $\{s_2, a_3\}$ using `GraB`, in which s_1 and s_2 have moved closer to t_2 and t_3.

we argue that the summation terms over target nodes and over source nodes where $r \neq i$ can be neglected. With $\boldsymbol{x}$ set equal to $\boldsymbol{x_i}$, we can thus approximate the utility as follows:

$$\frac{\partial \delta_{S,T}(\boldsymbol{x}_i;\boldsymbol{X})}{\partial a_{ij}} \approx \left(\nabla_{\boldsymbol{x}}\delta_{S,T}(\boldsymbol{x};\boldsymbol{X}) + \nabla_{\boldsymbol{x}_i}\delta_{S,T}(\boldsymbol{x};\boldsymbol{X})\right)\Big|^T_{\boldsymbol{x}=\boldsymbol{x_i}} \cdot \frac{\partial \boldsymbol{x}_i(\boldsymbol{A})}{\partial a_{ij}}. \tag{3}$$

Computing this approximation is far more efficient than a brute-force computation of the utility. It is scalable especially if the derivatives $\frac{\partial \boldsymbol{x}_i(\boldsymbol{A})}{\partial a_{ij}}$ of the embeddings can be computed analytically. An embedding method for which that is the case is presented in Sec. 3.3.

3.1.3 KDE as Density Estimation Method. Kernel density estimation (KDE), also known as Parzen window estimation [21], is a non-parametric density estimator. The flexibility arising from KDE's non-parametric nature makes it a very popular approach for data drawn from a complicated distribution [6]. We use KDE as the density estimator in the local imbalance measure.

Definition 4 (Kernel Density Estimation [21]). *Given a set of d-dimensional points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ forming the rows of data matrix $\boldsymbol{X}$, and a kernel function K, the KDE for an arbitrary point $\boldsymbol{x}$ is defined as:*

$$p_{KDE}(\boldsymbol{x};\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{n} K(\boldsymbol{x} - \boldsymbol{x}_i).$$

For the kernel, we use multivariate Gaussian KDE:

$$K(\boldsymbol{x}) = (2\pi)^{-d/2}\,|H|^{-1/2}\,e^{-\frac{1}{2}\boldsymbol{x}^T H^{-1}\boldsymbol{x}},$$

where the so-called bandwidth matrix H is computed using Scott's rule [24]. Thus, we can quantify the utility (Definition 3) of adding link $\{i, j\}$ at node i by rewriting Eq. (3) with p_{KDE} as the density estimator. Doing this, observe that $\nabla_{\boldsymbol{x_i}}K(\boldsymbol{x} - \boldsymbol{x_i})|_{\boldsymbol{x}=\boldsymbol{x_i}} = 0$, such that also $\nabla_{\boldsymbol{x}_i}\delta_{S,T}(\boldsymbol{x};\boldsymbol{X})|_{\boldsymbol{x}=\boldsymbol{x_i}} = 0$. Hence, with this KDE as density estimator, Eq. (3) simplifies to:

$$\frac{\partial \delta_{S,T}(\boldsymbol{x}_i;\boldsymbol{X})}{\partial a_{ij}} \approx \nabla_{\boldsymbol{x}}\delta_{S,T}(\boldsymbol{x};\boldsymbol{X})|^T_{\boldsymbol{x}=\boldsymbol{x_i}} \cdot \frac{\partial \boldsymbol{x_i}}{\partial a_{ij}}(\boldsymbol{A}). \tag{4}$$

3.2 The `GraB` Algorithm

Having defined the network imbalance measure and the link utility measure to be used as a proxy, we can now craft a scalable algorithm to optimize the imbalance. We designed `GraB` using three concepts.

1. Greedy link selection: As discussed, solving problem 1 exactly requires computing the imbalance reduction for every possible set of k links between set S and U, which is computationally infeasible. Instead, `GraB` picks links greedily based on the link utility measure. Although the link selection step aims to provide the most beneficial links to add to the network, the network embedding after adding the links might be different than expected, due to the following two issues.

2. Include links in batches: The first issue is that we are using the utility of adding a link, assuming that the rest of the network embedding remains the same. However, the effect of adding a link may change after another link is included (especially if they are close to each other in the network). A solution would be to re-embed the network after the inclusion of each link, but this is computationally costly. Thus, as a trade-off between cost and accuracy, we add links to the network in several batches: in each iteration, we select a batch of b links from the top candidates ranked by link utility, add them to the network, and re-embed the network. Additionally, since the embedding of a node is affected more by its direct links, we add at most one link per source node in a batch.

3. Post-hoc filtering: The second issue is that adding a link may actually not reduce the imbalance, for example because the source node moves a lot in the embedding and 'jumps over' the area of higher local imbalance $\delta_{S,T}$. This is the case when the derivative from Definition 3 is not a good approximation to the finite difference of the local imbalance measure. An example is shown in Figure 3, which shows the heatmap of $\delta_{S,T}$ on a 2-dimensional embedding of the Weibo network [28] using conditional network embedding. In this example, node s jumps over the area with a high $\delta_{S,T}$ and ends up in a worse position in terms of $\delta_{S,T}$.

Hence, to find the most beneficial b links to add to the network in each batch, we instead select the $l \cdot b$ top candidate links using the link utility measure ($l \geq 1$ controls how many more links than b

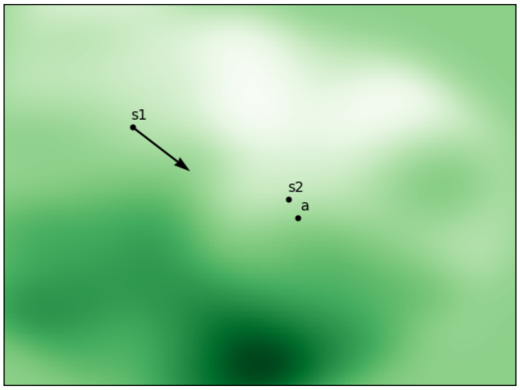

Figure 3: An illustration of link utility error in heatmap of $\delta_{S,T}$ on the 2D embedding of Weibo network using CNE, with females as source nodes and males as target nodes. Point s_1 shows the position of node s before adding the link between nodes s (source node) and a (auxiliary node) that has a positive utility, and s_2 shows the new position of node s after adding the link, where the arrow shows the direction that s would move. However, node s jumps over the area with higher $\delta_{S,T}$ and moves to a position with lower $\delta_{S,T}$.

has to be added to the network for re-embedding). After adding $l \cdot b$ candidate links, we re-embed the network and select the first b links that caused their connected source nodes be moved to positions with higher $\delta_{S,T}$. The parameter l has to be large enough so that at least b links would end up in a better position after re-embedding.

GraB: In summary, we select k links to add to the network to reduce the imbalance in several batches, each of size b. For each batch, we select the $l \cdot b$ top candidate links (at most one link per source node) using the link utility measure, add them to the network, re-embed the network, and select the top b links that caused their connected source nodes to move to positions with higher $\delta_{S,T}$.

We call this method ***Graph Balancing (*GraB*)***. Full pseudocode for the algorithm is given in Appendix B[2].

3.3 Choice of Embedding Method

The generic method GraB applies to network embedding methods where the optimal embeddings are differentiable w.r.t. changes in link strength, i.e. where $\frac{\partial \boldsymbol{x}_i}{\partial a_{ij}}(\boldsymbol{A})$ for $i \in S$ and $j \in T$ and $\boldsymbol{x_i}$ the embedding of node i, can be evaluated. To the best of our knowledge, two methods satisfy this requirement: LINE [25] and Conditional network embedding (CNE) [16]. We chose to use CNE for three reasons. First, LINE uses the inner product as the similarity measure between node embeddings, whereas KDE (and CNE) are based on the Euclidean distance. This mismatch would make the local imbalance proxy less effective. Second, re-embedding the network starting from an initial embedding is easily done with CNE, greatly speeding up GraB. And third, CNE was shown to outperform LINE.

Let $\boldsymbol{H}_i$ denote the (analytically computable) Hessian of the objective function of CNE w.r.t. $\boldsymbol{x}_i$. Then, with γ a hyper-parameter of CNE, Kang et al. [17] showed that $\frac{\partial \boldsymbol{x}_i(\boldsymbol{A})}{\partial a_{ij}}$ is given by:

$$\frac{\partial \boldsymbol{x}_i}{\partial a_{ij}}(\boldsymbol{A}) = -\gamma \boldsymbol{H}_i^{-1}(\boldsymbol{x}_j - \boldsymbol{x}_i).$$

4 EXPERIMENTAL EVALUATION

In this section, we describe the experimental evaluation of GraB. In a qualitative experiment, we investigate question **Q1**: Does GraB work as expected in moving source nodes towards areas with more target nodes compared to source nodes? In the quantitative experiments, we investigate two questions **Q2**: How does GraB perform in reducing the imbalance ψ compared to the baselines? **Q3**: Does GraB constantly reduce imbalance by increasing the number of links added to the networks? In a hyper-parameter sensitivity experiment, we investigate question **Q4**: Is GraB sensitive to the batch size b? Finally, we investigate question **Q5**: How does GraB perform in terms of execution time compared to the baselines?

We first discuss the datasets, baselines, and settings. Next, we present the result of each experiment. The source code for repeating the experiments are available here [3].

4.1 Datasets

We evaluated the methods using three datasets described below:

VDAB [4]: VDAB is the employment service of Flanders in Belgium. It provides a platform for job seekers to find jobs. The dataset contains a sample of the applications made by job seekers to available job vacancies from January 2018 until October 2020. We construct the job market network with three sets of nodes: job seekers, job vacancies, and skills. Job vacancies are connected to job seekers that have applied for them and to the skills they require. Our goal is to reduce the imbalance between job seekers (source nodes) and job vacancies (target nodes) by adding links connecting job seekers with skills. This could be seen as teaching a group of job seekers some skills, in a way that balances the job market network.

Weibo [28]: Weibo is the most popular Chinese microblogging service. The dataset contains tweets of the users and the topic distribution of each tweet. We construct the network with three sets of nodes: male users, female users, and topics. Users are connected to their friends (reciprocal follow relationships) and the top topics of their tweets. To find the top topics for each tweet, we first sort the topic probabilities (relevance of topic for the tweet) in descending order. Next, we select the top topics until the difference of the probabilities of two consecutive topics is greater than a very small threshold (1e-6). We select a sample from the dataset by only considering tweets for the first year of the data. Our goal is to reduce the imbalance between males (source nodes or target nodes) and females (target nodes or source nodes) by adding new links connecting males/females with topics (auxiliary nodes). It is more like recommending tweets with specific topics to the users to increase their interest in that topic. In the experiments, we call this dataset **Weibo-mf** if males are the source nodes and females are the target nodes. Otherwise, we call the dataset **Weibo-fm**.

Movielens [14]: MovieLens is a web-based movie recommender system. The dataset contains 100000 user ratings on movies. We construct the network with three sets of nodes: users, movies, and

[2] https://github.com/aida-ugent/GraB/blob/master/GraB_appendix.pdf

[3] https://github.com/aida-ugent/GraB

[4] https://www.vdab.be/

Table 1: Main statistics of the networks used for evaluation.

Dataset	VDAB	Weibo-mf	Weibo-fm	Movielens
Nodes	5016	1364	1364	2644
Source Nodes	2358	822	442	1682
Target Nodes	2463	442	822	943
Auxiliary Nodes	195	100	100	19
Links	48451	14308	14308	102893
Average Degree	19	20	20	77

movie genres. There is a link between each user and movie for each rating. We also connect each movie to its genres. Our goal is to reduce the imbalance between movies (source nodes or target nodes) and users (target nodes or source nodes) by adding new links connecting movies with genres (auxiliary nodes).

We only used the largest connected component in each network. Table 1 shows the main statistics of each of the networks.

4.2 `GraB`-variants and baselines evaluated

As mentioned earlier, we are the first to introduce the concept of imbalance in a network in the way described in Section 2.2 and to reduce it by adding links to the network. However, there exist other methods that try to add links to the networks to make them more cohesive. We consider two of those methods [10, 20] for comparison.

Parotsidis et al. [20] minimize the average shortest path in a network by adding links. Garimella et al. [10] compute controversy between two sets of nodes using the random walks starting from one set, and ending in the same or the other set. The main difference between the imbalance and controversy is that the amount of links between nodes in the same set has a major effect on the controversy, which is not necessarily the case in computing the imbalance. Moreover, we use the distance in the embedding space as the cost of matching node pairs, while they consider the actual links in the network to compute the controversy.

We also designed a random method combined with our proposed greedy algorithm, and a pure random method for comparison.

In summary, the following methods will be evaluated:

GraB: The main method proposed in this paper.

S-GraB: 'Simple Graph Balancing' is the same as GraB without comparing the value of $\delta_{S,T}$ of the previous and the new embedding of the source node (i.e. without post-hoc filtering). I.e., S-GraB selects b links connecting source nodes with auxiliary nodes for each batch from the link selection step. To select k links to add to the network, S-GraB runs $\frac{k}{b}$ iterations. S-GraB is evaluated to assess the importance of the post-hoc filtering.

Since the problem of graph balancing has not been studied before, we also propose two simple baselines for comparison:

ROV [10]: 'Recommend opposing view' adds links to the network to reduce controversy. In this work, k links between high degree nodes in sets S and T that reduce controversy the most, are added using a greedy algorithm. We adopt this method for our problem setting by adding links between sets S and U using the same method.

SSW [20]: 'Shortcuts for a smaller world' adds links to the network to reduce the average shortest path length. In this work, k links are added to the network using different strategies. We employ the greedy strategy, since it has the best performance [20].

S-Random: The 'Simple random' baseline selects k random links connecting source nodes with auxiliary nodes.

I-Random: The 'Intelligent random' baseline is the same as GraB, except that we select random links connecting source nodes with auxiliary nodes in the link selection step. Since the link selection step is a random algorithm, adding links in several batches probably does not help the performance. Hence, I-Random adds links in one batch and re-embeds the network to select links that moved their source nodes to positions with higher $\delta_{S,T}$ (post-hoc filtering). We expect that adding random links results in making the graph denser and nodes tend to be closer to each other.

In summary, I-Random selects $l \cdot k$ random links connecting source nodes with auxiliary nodes, re-embeds the network, and the same as GraB, it only selects links that helped their source nodes end up in a position with a higher $\delta_{S,T}$ (post-hoc filtering).

4.3 Experimental Settings

In the quantitative experiments, we compare methods in terms of ψ (Definition 1). We conduct the experiments on CNE with dimensions 2 and 4 with a combination of block and degree prior (see [16]). We average the results for I-Random and S-Random over 3 repetitions to smoothen out random fluctuations.

In the qualitative experiment of GraB, we set $k = 10$ and $l = 5$. We run the experiment for 2-dimensional embedding.

In the experiment in Sec. 4.5.1, the methods are evaluated on several datasets. We set $k = 100$ for this experiment. We tune b from values {25, 100} for GraB and S-GraB. We also tune l from values {3, 5} for I-Random and GraB (S-GraB does not have hyper-parameter l, since it does not need to compare the value of $\delta_{S,T}$ of the previous and the new embedding of the source node). We use the author's implementation of computing the controversy in a network between two sets and used their default hyper-parameters, which is used in ROV. We used 10% of the high degree source nodes and 20 high degree auxiliary nodes for the candidate selection in ROV. We also use the author's implementation of SSW. SSW does not require to set any particular hyper-parameter.

In the experiment in Sec. 4.5.2, we analyze the behavior of GraB with different values of k. We increase k from 25 to 1000 by 25 and report ψ values. We set $b = 25$ and $l = 5$.

In the experiment in Sec. 4.6, we analyze the behavior of GraB with different values of batch size b. We set $k = 100$ and $l = 5$ for this experiment and vary b from 1 to 100.

4.4 Qualitative Evaluation

In this section, we show that GraB moves source nodes to the areas in the embedding space with having more target node percentage in their neighborhood than before (**Q1**). We first add 10 links to each dataset. Next, we analyze two sample source nodes that we added links to and compare the percentage of target nodes in their neighborhood. The percentage of target nodes in 25 nearest neighbors (only neighbors from source nodes and target nodes) of two sample source nodes for each network is presented in Table 2.

The number of target nodes in 25 nearest neighbors of the two source nodes is increased for each network. It means that GraB

Table 2: Percentage of target nodes in 25 nearest neighbors of source nodes.

Dataset	Node	Main Graph	GraB
VDAB-CNE2	v_1	28%	**56%**
	v_2	40%	**72%**
Weibo-mf-CNE2	v_1	48%	**64%**
	v_2	20%	**28%**
Weibo-fm-CNE2	v_1	44%	**64%**
	v_2	28%	**48%**
Movielens-CNE2	v_1	32%	**68%**
	v_2	32%	**56%**

moved source nodes to areas where they have more access to target nodes, and also fewer source nodes are competing with them to access the same target nodes.

4.5 Quantitative Evaluation

4.5.1 Baseline Comparison . Here we compare the methods in terms of ψ on all datasets. We report ψ on the main graph as well (**Q2**). Table 3 shows the result of this experiment.

GraB reduces ψ over the main graph and outperforms the other methods in all datasets. S-GraB also outperforms other baselines in most datasets. This shows that the link selection based on the link utility is choosing proper links to add to the network, and also post-hoc filtering of links in GraB improves the results.

For the baselines, I-Random improves ψ over the main graph and S-Random. S-Random however, is not effective and gains the same results as the main graph. Hence, the greedy algorithm with post-hoc filtering proposed in this paper (applied in I-Random) is helpful even with a random link selection.

The other methods SSW and ROV do not perform particularly well (in some datasets even increases the imbalance, and the amount of imbalance reduction in other datasets is less than GraB), since they are designed for a different purpose and objective function.

As a result, we can conclude that both the link selection step based on link utility and the greedy algorithm of GraB with post-hoc filtering are crucial to solve the problem and they are both playing an important role in reducing ψ of the main graph.

4.5.2 Effect of adding a different number of links on GraB. In this experiment, we evaluate the performance of GraB for different values of k, i.e., number of links added to the network (**Q3**). Figure 4 shows the result of this experiment for all datasets.

In most datasets, ψ decreases by adding more links to the network. The amount of reduction in ψ mostly decreases, or even in some datasets ψ increases as we add more links to the network. The reason is that in the beginning, there are more links available to add to the network. As we gradually add the most beneficial links to the network, the remaining candidate links might have smaller utilities and less certainty. Hence, GraB selects links from those candidates which results in worsening the performance globally. So GraB does not constantly reduce imbalance by increasing the number of links added to the networks.

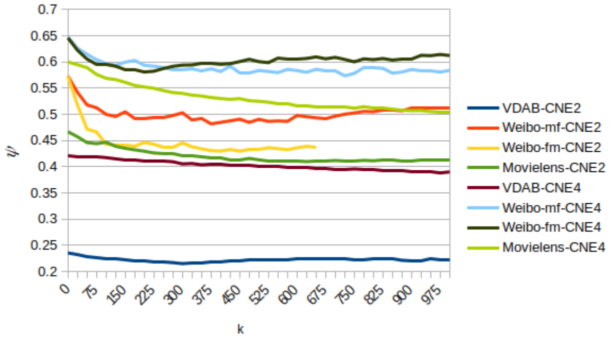

Figure 4: ψ after adding different number of links to each network by GraB.

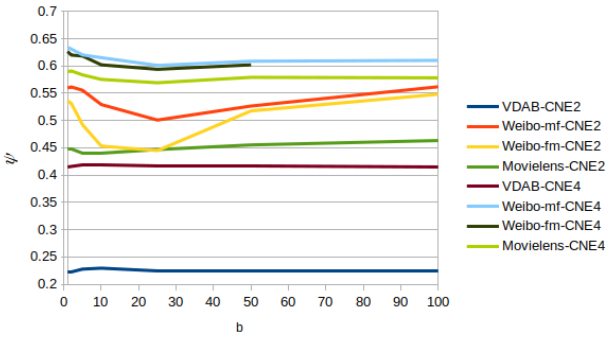

Figure 5: ψ after adding 100 links to each network by GraB for different values of b.

The reason that there is a blank spot in Weibo-fm-CNE2 dataset for $k > 675$ is that the method could not find 25 links to add to the network in the next batch (because after re-embedding, not 25 links moved their source nodes to a position with a higher value of $\delta_{S,T}$) and we stopped running the method after adding 675 links.

4.6 Hyper-Parameter Sensitivity

In this section, we evaluate the sensitivity of GraB w.r.t. the batch size b (**Q4**). Figure 5 shows the result of this experiment.

Smaller b means fewer links are added to the network and hence, the network embedding is more stable. As a result, the algorithm finds links with the highest utilities more accurately. The issue of small values of b is that the effect of adding links on each other is neglected. For example, consider adding link $\{i, j\}$ and $\{p, q\}$ to a network one by one ($b = 1$). Since GraB adds $\{i, j\}$ at the first step, it means that node i has moved to an area with higher local imbalance $\delta_{S,T}$. Yet, it is possible that after adding $\{p, q\}$ the local imbalance $\delta_{S,T}$ at the position of node i becomes less than its original position (since moving node p changes the density of the source nodes in the embedding space). On the other hand, by adding the two links at the same time ($b = 2$), $\{i, j\}$ will be filtered in post-hoc filtering step and will not be selected and there will be the opportunity to select other links to reduce the imbalance.

Table 3: ψ after adding 100 links to each network. Best performance per dataset is highlighted in bold.

Dataset	Main Graph	S-Random	I-Random	SSW	ROV	S-GraB	GraB
VDAB-CNE2	0.2357	0.2335	0.2286	0.2351	0.2362	0.2254	**0.2245**
Weibo-mf-CNE2	0.5729	0.5650	0.5479	0.5361	0.56891	0.5191	**0.4955**
Weibo-fm-CNE2	0.5724	0.5750	0.5419	0.6763	0.5739	0.4917	**0.4425**
Movielens-CNE2	0.4667	0.4653	0.4654	0.4501	0.4652	0.4458	**0.4400**
VDAB-CNE4	0.4211	0.4206	0.4164	0.4195	0.4213	0.4170	**0.4137**
Weibo-mf-CNE4	0.6432	0.6423	0.6245	0.6252	0.6506	0.6269	**0.6049**
Weibo-fm-CNE4	0.6536	0.6486	0.6290	0.7175	0.6446	0.6235	**0.5895**
Movielens-CNE4	0.6031	0.5927	0.5990	0.5856	0.6002	0.5748	**0.5686**

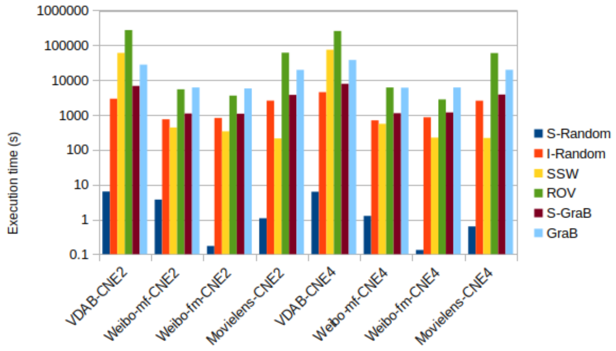

Figure 6: Execution time (log-scale) of all methods for experiment 4.5.1 on each evaluated network.

On the other hand, having a large b also is not always the best option because by adding $l \cdot b$ links at the same time, the density of source nodes and target nodes are not stable and varies a lot (since by filtering some links and selecting only b links, the actual densities after re-embedding will be different). Moreover, by adding so many links at the same time, the behavior of the embeddings is less predictable due to the impact of links on each other. Hence, the method will not be accurate in selecting and adding links to the network, to reduce the imbalance.

Middle values of b show more stable results in most datasets.

There is a blank spot in Weibo-fm-CNE4 dataset for $b = 100$ because GraB could not find 100 links to add to the network (because after re-embedding, not 100 links moved their source nodes to a position with a higher $\delta_{S,T}$). Hence, we did not report ψ.

4.7 Execution Time

In this experiment, we compare the methods in terms of the execution time (**Q5**) with the same settings as experiment 4.5.1. Figure 6 shows the execution time in seconds (log-scale) for all methods, including the time for hyper-parameter tuning.

GraB and ROV have the highest execution time among all methods. GraB has a high execution time due to the number of hyper-parameters to be tuned, the link selection step, and also re-embedding after adding each batch. ROV has a high execution time due to the large number of candidates and the time needed for computing the controversy after adding each candidate link to the graph.

S-GraB has a greater execution time than I-Random. For more analysis, we first count the number of times they need to re-embed the networks. I-Random re-embeds the networks once for each hyper-parameter selection because it does not add links in batches. Since we tune l from 2 values, I-Random selects links and re-embeds the networks 2 times in total. S-GraB re-embeds the networks $\frac{k}{b} - 1$ times for each value of l. Since $k = 100$ and we tune b from values $\{25, 100\}$, S-GraB re-embeds the networks 3 times in total. Besides, S-GraB performs the link selection step $\frac{k}{b}$ times. Thus, it performs the link selection step 5 times in total (once when $b = 100$ and 4 times when $b = 25$). Moreover, the link selection step is more time consuming in S-GraB than I-Random which selects links randomly. Hence, S-GraB is slower than I-Random.

Moreover, S-Random is a simple random method and the execution time is almost zero for all datasets.

5 RELATED WORK

Imbalance in the workforce has been studied in various systems [27, 29]. However, our work differs from these studies. Previous studies are mostly domain-specific and they analyze the supply and demand based on domain-specific features such as educational training program length, retirement, and salary. Previous studies in this area also lack a global measure to quantify the imbalance between two different entities. In contrast to this, we tackle the problem from a graph analysis approach. We propose a quantification of the imbalance in the network using its embedding and a method to reduce the imbalance by adding links to the network.

Another line of research focuses on matching problems [18, 22]. Finding the cost of the fractional perfect b-matching [1] with minimum cost in bipartite networks is related to our work. In this problem, the goal is to find a matching between two sets of nodes in a network with minimum total cost. We define the imbalance as the cost of the minimum cost fractional perfect b-matching on a new bipartite network created from the original network. Our work differs from the studies focusing on matching, since we do not address the computational problem of how to find the matching, we only use the cost of the matching to quantify the imbalance. Moreover, we add links to the network (not directly between the two sets of nodes of interest), to change the cost of links between the two sets of nodes, and hence, reduce the imbalance in the network.

Fairness in node embeddings is studied in various research papers [2, 3]. These studies try to learn an unbiased network embedding. The similarity between learning an unbiased embedding and reducing the imbalance in a network embedding is that they both try to have a mix between nodes with different attributes or different types (job seeker and job vacancy, female and male, etc.). The main difference is that debiasing methods learn an unbiased embedding based on the original network, while we modify the network in order to make it more balanced. A secondary difference is in the quantification of the imbalance or unfairness. In our setting, we intend to bring two sets of nodes closer, to reduce the imbalance, while the goal to reduce unfairness is that the two sets of nodes cannot be separated, which is not important in our case.

There are also several papers aiming to add links to a network to modify the network structure. Some of the researches focus on adding links to a network to make it more cohesive, where cohesiveness is quantified using network properties such as shortest paths [19, 20], diameter [7], information unfairness [15], controversy [10] and structural bias [13]. The paper by Garimella et al. [10] is most related to our work, since they add links to the network to reduce the controversy between two sets of nodes. However, the difference between our works is that we consider a different measure to compute the imbalance and use a different approach to optimize it.

6 CONCLUSIONS AND FUTURE WORK

We defined and quantified the concept of imbalance between two sets of nodes in a network, and introduced the novel problem of reducing that imbalance by introducing new links. We proposed GraB, a scalable algorithm to tackle this problem, leveraging a number of well-motivated heuristics to trade-off speed with accuracy. We presented experiments applying GraB together with CNE as the network embedding method to various networks. The experimental results indicate that GraB outperforms (also newly proposed) baselines for reducing imbalance in a network embedding.

In future work, we plan to investigate the benefits of a new link for individual nodes (e.g., improving the access to a target set of nodes) instead of just for the global balance of the network, as well as other problem settings such as reducing the imbalance in a network by *removing* a specific number of links from the network (e.g., changing job contents and required skills, making jobs more accessible), or both adding and removing links at the same time. Moreover, a more detailed investigation of the impact of hyperparameters such as the KDE bandwidth would be useful.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, and from the FWO (project no. G091017N, G0F9816N, 3G042220). Part of the experiments were conducted on pseudonimized HR data generously provided by VDAB.

REFERENCES

[1] Roger E Behrend. 2013. Fractional perfect b-matching polytopes I: General theory. *Linear Algebra Appl.* 439, 12 (2013), 3822–3858.

[2] Avishek Joey Bose and William L Hamilton. 2019. Compositional fairness constraints for graph embeddings. *arXiv preprint arXiv:1905.10674* (2019).

[3] Maarten Buyl and Tijl De Bie. 2020. DeBayes: a Bayesian method for debiasing network embeddings. *arXiv preprint arXiv:2002.11442* (2020).

[4] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1616–1637.

[5] Ramon Ferrer I Cancho and Richard V Solé. 2001. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 268, 1482 (2001), 2261–2265.

[6] Yen-Chi Chen. 2017. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology* 1, 1 (2017), 161–187.

[7] Erik D Demaine and Morteza Zadimoghaddam. 2010. Minimizing the diameter of a network using shortcut edges. In *Scandinavian Workshop on Algorithm Theory*. Springer, 420–431.

[8] Linton C Freeman. 2000. Visualizing social networks. *Journal of social structure* 1, 1 (2000), 4.

[9] Sheng Gao, Huacan Pang, Patrick Gallinari, Jun Guo, and Nei Kato. 2017. A novel embedding method for information diffusion prediction in social network big data. *IEEE Transactions on Industrial Informatics* 13, 4 (2017), 2097–2105.

[10] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2017. Reducing controversy by connecting opposing views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 81–90.

[11] Palash Goyal and Emilio Ferrara. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151 (2018), 78–94.

[12] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[13] Shahrzad Haddadan, Cristina Menghini, Matteo Riondato, and Eli Upfal. 2021. RePBubLik: Reducing the Polarized Bubble Radius with Link Insertions. *arXiv preprint arXiv:2101.04751* (2021).

[14] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[15] Zeinab S Jalali, Weixiang Wang, Myunghwan Kim, Hema Raghavan, and Sucheta Soundarajan. 2020. On the Information Unfairness of Social Networks. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 613–521.

[16] Bo Kang, Jefrey Lijffijt, and Tijl De Bie. 2018. Conditional network embeddings. *arXiv preprint arXiv:1805.07544* (2018).

[17] Bo Kang, Jefrey Lijffijt, and Tijl De Bie. 2019. Explaine: An approach for explaining network embedding-based link predictions. *preprint arXiv:1904.12694* (2019).

[18] Vladimir Kolmogorov. 2009. Blossom V: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation* 1, 1 (2009), 43–67.

[19] Manos Papagelis, Francesco Bonchi, and Aristides Gionis. 2011. Suggesting ghost edges for a smaller world. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. 2305–2308.

[20] Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2015. Selecting shortcuts for a smaller world. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 28–36.

[21] Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics* 33, 3 (1962), 1065–1076.

[22] Lyle Ramshaw and Robert E Tarjan. 2012. On minimum-cost assignments in unbalanced bipartite graphs. *HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1* (2012).

[23] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 59–66.

[24] David W Scott. 2015. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.

[25] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[26] Athanasios Theocharidis, Stjin Van Dongen, Anton J Enright, and Tom C Freeman. 2009. Network visualization and analysis of gene expression data using BioLayout Express 3D. *Nature protocols* 4, 10 (2009), 1535.

[27] Graham Willis, Andrew Woodward, and Siôn Cave. 2013. Robust workforce planning for the English medical workforce. In *Conference Proceedings, The 31st International Conference of the System Dynamics Society*.

[28] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. 2013. Social influence locality for modeling retweeting behaviors. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

[29] Pascal Zurn, Mario R Dal Poz, Barbara Stilwell, and Orvill Adams. 2004. Imbalance in the health workforce. *Human resources for health* 2, 1 (2004), 1–12.