Finding Hierarchy in Directed Online Social Networks

Mangesh Gupte
Dept of Computer Science
Rutgers University
Piscataway, NJ
mangesh@cs.rutgers.edu

Pravin Shankar
Dept of Computer Science
Rutgers University
Piscataway, NJ
spravin@cs.rutgers.edu

Jing Li
MIT
Cambridge, MA
lijing@mit.edu

S. Muthukrishnan
Dept of Computer Science
Rutgers University
Piscataway, NJ
muthu@cs.rutgers.edu

Liviu Iftode
Dept of Computer Science
Rutgers University
Piscataway, NJ
iftode@cs.rutgers.edu

ABSTRACT

Social hierarchy and stratification among humans is a well studied concept in sociology. The popularity of online social networks presents an opportunity to study social hierarchy for different types of networks and at different scales. We adopt the premise that people form connections in a social network based on their perceived social hierarchy; as a result, the edge directions in directed social networks can be leveraged to infer hierarchy. In this paper, we define a measure of hierarchy in a directed online social network, and present an efficient algorithm to compute this measure. We validate our measure using ground truth including Wikipedia notability score. We use this measure to study hierarchy in several directed online social networks including Twitter, Delicious, YouTube, Flickr, LiveJournal, and curated lists of several categories of people based on different occupations, and different organizations. Our experiments on different online social networks show how hierarchy emerges as we increase the size of the network. This is in contrast to random graphs, where the hierarchy decreases as the network size increases. Further, we show that the degree of stratification in a network increases very slowly as we increase the size of the graph.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Sociology*; E.0 [**Data**]: General

General Terms

Algorithms, Experimentation, Measurement

Keywords

Social Networks, Hierarchy, Measure

1. INTRODUCTION

Social stratification refers to the hierarchical arrangement of individuals in a society into divisions based on various factors such as power, wealth, knowledge and importance [13]. Stratification existed among humans since the very beginning of human society and continues to exist in modern society. In some settings, such as within an organization, the hierarchy is well known, whereas in other settings, such as conferences and meetings between a group of people, the hierarchy is implicit but discernible.

The popularity of online social networks has created an opportunity to study sociological phenomenon at a scale that were earlier unfathomable. Phenomenon such as small diameter in social networks [24] and strength of weak ties [12] have been revisited in light of the large data now available about people and their connections [1, 25, 3]. Online social networks present an opportunity to study how social hierarchy emerges.

Scientists have observed dominance hierarchies within primates. Thorleif Schjelderup-Ebbe showed a *pecking order* among hens [23] where each hen is aware of its place among the hierarchy and there have been various papers that investigate the importance of such a hierarchy [10, 9]. However, data from experimental studies indicates that the dominance graph contains cycles and hence, does not represent true "hierarchy". There has been a lot of work on extracting a chain given this dominance graph [6, 5, 2].

Stratification manifests among humans in the form of a social hierarchy, where people higher up in the hierarchy have higher social status than people lower in the hierarchy. With the wide adoption of online social networks, we can observe the network and can leverage the links between nodes to infer social hierarchy. Most of the popular online social networks today, such as Twitter, Flickr, YouTube, Delicious and LiveJournal contain directed edges.[1] Our central premise is that there is a global "social rank" that every person enjoys, and that individuals are aware of their rank as well as the rank of people they connect to.

Given a social graph, we cannot directly observe the ranks of people in the network, we can only observe the links. We premise that the existence of a link indicates a social rank recommendation; a link $u \to v$ (u is a follower of v) indicates a social recommendation of v from u. If there is no reverse link from v to u, it might indicate that v is higher up in the hierarchy than u. We assume that in social networks, when people connect to other people who are lower in the hierarchy, this causes them *social agony.* To infer the

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

[1] Facebook is an exception with undirected edges.

ranks of the nodes in the network, we find the best possible ranking, i.e. the ranking that gives the least *social agony*.

In this paper, we define a measure that indicates how close the given graph is to a true hierarchy. We also give a polynomial time algorithm to evaluate this measure on general directed graphs and to find ranks of nodes in the network that achieve this measure.

We use our algorithm to measure hierarchy in different online social networks, including Twitter, Delicious, Flickr, YouTube, LiveJournal, and curated lists of several categories of people based on different occupations, and different organizations.

We experimentally find, using a college football dataset, that the edge direction encodes hierarchy information. The social strata of people in a online social networks, measured using our metric, shows strong correlation with human-observed ground truth such as Wikipedia notability, as well as other well-known metrics such as page rank and friend-follower ratio. Our experiments show that hierarchy emerges as the size of an online social network grows. This is in contrast to random graphs, where the hierarchy decreases as the network size increases. Finally, we show that hierarchy in online social networks does not grow indefinitely; instead, there are a small number of levels (strata) that users are assigned to and this number does not grow significantly as the size of the network increases.

The key contributions of this paper are:

1. We define a measure of hierarchy for general directed networks.
2. We give a polynomial time algorithm to find the largest hierarchy in a directed network.
3. We show how hierarchy emerges as the size of the networks increases for different online social networks.
4. We show that, as we increase the size of the graph in our experiments, the degree of stratification in a network does not increase significantly.

2. HIERARCHY IN DIRECTED SOCIAL NETWORKS

One of the most popular ways to organize various positions within an organization is as a tree. A general definition of hierarchy is a (strict) partially ordered set. This definition includes chains (Figure 1a) and trees (Figure 1b) as special cases. We can view a partially ordered set as a graph, where each element of the set is a node and the partial ordering ($u > v$) gives an edge from u to v. The fact that the graph represents a partial order implies that the graph is a Directed Acyclic Graph (DAG). From now on, we use DAGs as examples of perfect hierarchy. Figure 1c shows an example of a DAG.

Let us define a measure of hierarchy for directed graphs that might contain cycles. Consider a network $G = (V, E)$ where each node v has a rank $r(v)$. Formally, the rank is a function $r : V \to \mathbb{N}$ that gives an integer score to each vertex of the graph. Different vertices can have the same score.

In social networks, where nodes are aware of their ranks, we expect that higher rank nodes are less likely to connect to lower rank nodes. Hence, directed edges that go from lower rank nodes to higher rank nodes are more prevalent than edges that go in the other direction. In particular, if $r(u) < r(v)$ then, edge $u \to v$ is expected and does not cause any "agony" to u. However, if $r(u) \geq r(v)$, then edge $u \to v$ causes agony to the user u and the amount of agony depends on the difference between their ranks. We shall assume that the agony caused to u by each such reverse edge is equal to $r(u) - r(v) + 1$.[2] [3]. Hence, the agony to u caused by edge (u, v) relative to a ranking r is $\max(r(u) - r(v) + 1, 0)$.

We define the agony in the network relative to the ranking r as the sum of the agony on each edge:

$$A(G, r) = \sum_{(u,v)\in E} \max(r(u) - r(v) + 1, 0)$$

We defined agony in terms of a ranking, but in online social networks, we can only observe the graph G and cannot observe the rankings. Hence, we need to infer the rankings from the graph itself. Since nodes typically minimize their agony, we shall find a ranking r that minimizes the total agony in the graph. We define the agony of G as the minimum agony over all possible rankings r:

$$A(G) = \min_{r\in\text{Rankings}} \left(\sum_{(u,v)\in E} \max(r(u) - r(v) + 1, 0) \right)$$

For any graph G, $A(G)$ is upper bounded by m, the number of edges in G (we prove this in Section 3.1 Equation 1). This motivates our definition of hierarchy in a graph:

Definition 1 (Hierarchy)**.** *The hierarchy $h(G)$ in a directed graph G is defined as*

$$\begin{aligned} h(G) &= 1 - \frac{1}{m} A(G) \\ &= \max_{r\in Rankings} \left(1 - \frac{1}{m} \sum_{(u,v)\in edges} \max(r(u) - r(v) + 1, 0) \right) \end{aligned}$$

For any graph G, the hierarchy $h(G)$ lies in $[0, 1]$. This follows from the fact that $A(G)$ lies in $[0, m]$. (Equation 1). To gain some intuition into this definition of hierarchy, we shall look at some example graphs and their hierarchy.

2.1 Examples

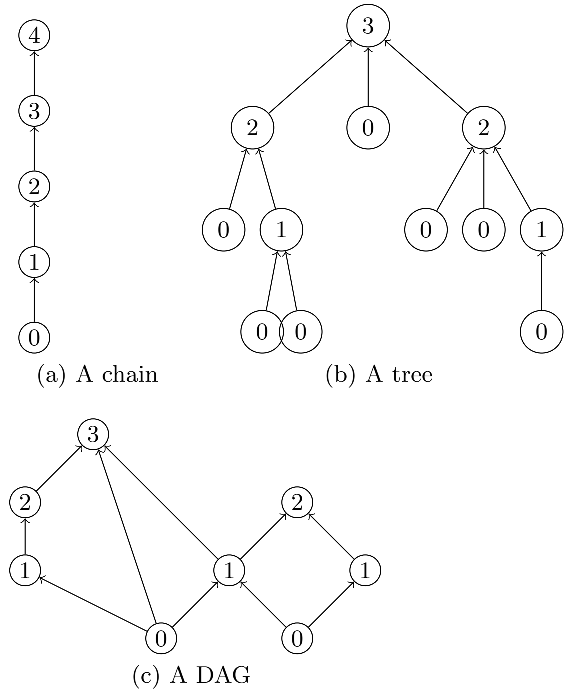
DAGs have perfect hierarchy. $h(G) = 1$ when G is a DAG. This is achieved by setting $r(v) > r(u) + 1$ for each edge (u, v) in the DAG. Figure 1 shows examples of graphs with perfect hierarchy. Nodes are labeled with levels. For this assignment, note that the agony on each edge is 0.

Consider the graph in Figure 2a. The hierarchy of this graph is $1 - \frac{1}{6} \times 2 = \frac{2}{3}$. If instead of the edge (r, l_1), the "deeper" edge (r, l_2) is present, as shown in Figure 2b, then the hierarchy of the new graph becomes $1 - \frac{1}{6} \times 4 = \frac{1}{3}$. This illustrates how hierarchy changes in a very simple setting. We shall explore this more in Section 4.

Directed cycles have no hierarchy. $h(G) = 0$ when G is a collection of edge disjoint directed cycles. We prove in Section 3.1 that for any assignment of labels to nodes, the

[2] Note that $r(u) - r(v)$ does not work, since it gives rise to trivial solutions like $r = 1$ for all nodes. The $+1$ effectively penalizes such degenerate solutions. Using any positive constant threshold c other than 1 does not change the analysis in any way.

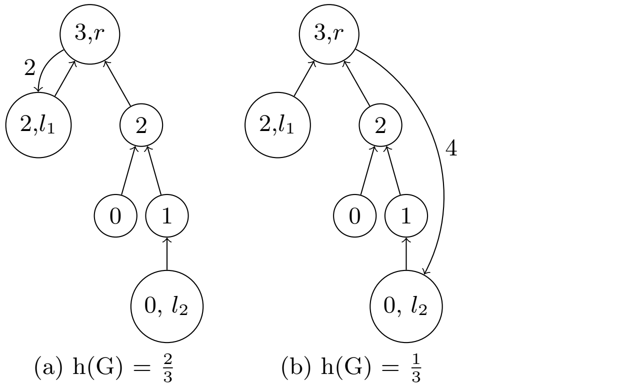
[3] An interesting direction for future work is to investigate a different measure of agony, in particular, a non-linear function like $\log(r(u) - r(v) + 1)$.



(a) A chain (b) A tree

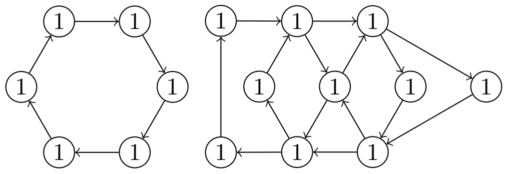
(c) A DAG

Figure 1: **Graphs with perfect hierarchy. h(G) = 1 for each of these graphs. Nodes labels indicate levels. All edges have agony 0.**



(a) h(G) = $\frac{2}{3}$ (b) h(G) = $\frac{1}{3}$

Figure 2: **Graphs with some hierarchy. All unlabeled edges have agony 0.**



(a) A Simple Cycle (b) A collection of cycles

Figure 3: **Graphs with no hierarchy. h(G) = 0 for each of these graphs. All edges have agony 1.**

agony is at least m. Figure 3 shows examples of graphs with 0 hierarchy. If each node is labeled the same, say 1, this is achieved.

3. EFFICIENTLY MEASURING HIERARCHY

To find the hierarchy $h(G)$ for a given graph G, we need to search over all rankings and find the best one. Since the number of rankings r is exponentially large, we need an efficient way to search among them. In Section 3.1, we present an efficient algorithm to find a ranking that gives the highest hierarchy for any directed graph G.[4]

3.1 Algorithm

In this section, we describe an algorithm that finds the optimal hierarchy for a given directed graph $G = (V, E)$. For notational convenience, we shall denote $n = |V|$ and $m = |E|$. For a scoring function $r : V \to \mathbb{N}$, the hierarchy relative to r is:

$$h(G, r) = 1 - \frac{1}{m} \sum_{(i,j)\in\text{edges}} \max(r(i) - r(j) + 1, 0)$$

The task is to find an r such that $h(G, r)$ is maximized over all scoring functions. But maximizing h is the same as minimizing the total agony $A(G, r)$. We formulate minimizing agony as the following integer program:

$$\begin{array}{rll}
\min & \sum_{(i,j)\in E} x(i,j) & \\
& x(i,j) \geq r(i) - r(j) + 1 & \forall (i,j) \in E \\
& x(i,j) \geq 0 & \forall (i,j) \in E \\
& r(i) \geq 0 & \forall i \in V \\
& x(i,j), r(i) \in \mathbb{Z} &
\end{array}$$

We now see a simple upper bound on the minimum value of the integer program. Consider the solution:

$$\begin{aligned}
r(i) &= 0 : \forall i \in V \\
x(i,j) &= 1 : \forall (i,j) \in E \qquad (1)
\end{aligned}$$

This is clearly feasible and the objective value for this is m. This gives a simple upper bound of m on the objective value of the above integer program.

To get insight into this problem, we look at the linear relaxation of this integer program and then form the dual linear program. The dual is:

$$\begin{array}{rll}
\max & \sum_{(i,j)\in E} z(i,j) & \\
& z(i,j) \leq 1 & \forall (i,j) \in E \\
& \sum_{j\in V} z(k,j) \leq \sum_{i\in V} z(i,k) & \forall k \in V \quad \text{(node-degree)} \\
& z(i,j) \geq 0 & \forall (i,j) \in E
\end{array}$$

We can strengthen the node-degree constraints without affecting the solution of the linear program by requiring strict

[4] This ranking may not be unique. In fact, if G is a DAG, then any ordering that gives a topological sort of G gives an optimal ranking.

equality, since if we sum over all k, we get:

$$\sum_{k\in V}\sum_{j\in V} z(k,j) \leq \sum_{k\in V}\sum_{i\in V} z(i,k)$$

Since both sides count the total number of edges in the graph, they are equal. Hence, equality must hold for each individual constraint as well. So, we can rewrite the node-degree condition as:

$$\sum_{j\in V} z(k,j) = \sum_{i\in V} z(i,k) \qquad \forall k \in V \qquad \text{(node-degree)}$$

When we restrict the dual variables to be 0 or 1 instead of in the range $[0,1]$, we can reinterpret the dual program as finding an Eulerian subgraph of the original graph.[5]

The reinterpretation gives us insight into the primal solution. By weak duality, the value of the primal is lower bounded by the value of any feasible dual solution. Hence, the primal value cannot become smaller than the size of the maximum(in terms of number of edges) Eulerian subgraph.

If the original graph G is Eulerian, this gives a lower bound of m. Equation 1 demonstrates a way to get m as the primal solution. Hence, the optimal primal value for Eulerian graphs is in fact m. This proves the observation that, for graphs that are a collection of directed cycles, the agony is m and hence, the hierarchy is 0.

We can directly solve the LP to get the best ranking when we do not restrict the rank of the node to be an integer. We shall prove that the linear program has an integral optimal solution. In fact, we give a combinatorial algorithm that finds the best ranking. We first use Algorithm 1 to construct an integral solution to the dual. Algorithm 2 uses the dual solution to come up with a integral primal solution. We show that the primal and dual solutions have the same objective value which, by LP duality, proves that both are optimal.

Algorithm 1 constructs a maximum Eulerian subgraph of G. Theorem 1 proves the correctness of Algorithm 1; that the subgraph is Eulerian and also that it has the maximum number of edges among such subgraphs. We leave the proofs of Theorems 1, 2 to Section A.

```
Algorithm 1: Finding a Maximum Eulerian Subgraph
  Input: Graph $G = (V, E)$
  Output:
    1. A subgraph $H$ of $G$ such that $H$ is Eulerian and has
       the maximum number of edges.
    2. A DAG such that $H \cup DAG = G$
  Set the weight of each edge $w(u,v) \leftarrow -1$
  while $\exists$ a negative cycle $C$ do
      for edge $(u,v) \in C$ do
          $w(u,v) \leftarrow -w(u,v)$
          Reverse the direction of the edge
      end
  end
  DAG $\leftarrow$ All edges labeled -1
  $H \leftarrow$ Reverse of all edges labeled +1 ($H$ is Eulerian)
```

Theorem 1. *Let H be the subgraph of G that contains the reverse of all (and only those) edges labeled +1 by Algorithm 1. Then, for each vertex v : $indeg_H(v) = outdeg_H(v)$. Also, for every subgraph T of G such that $indeg_T(v) = outdeg_T(v) : \forall v \in T$, number of edges in H is greater than the number of edges in T.*

To find the optimal value of hierarchy in the graph G, we need to assign a score r to the nodes and calculate the agony $x(i,j)$ value on each edge (i,j). Algorithm 2 gives a labeling for each node, from the ±1 edge labels given by Algorithm 1. The input graph to Algorithm 2 is the one output by Algorithm 1.

```
Algorithm 2: Label the graph given as a decomposition
of the Eulerian graph and a DAG
  Input: A Graph $G = (V, H \cup DAG)$ output by
         Algorithm 1. Edges in the Eulerian graph $H$ are
         labeled +1 and edges in $DAG$ are labeled -1.
  Output: A labeling $l$ of all vertices of $G$, such that the
          agony measure on $G$ with the given labels:
          $A(G,l)$, is equal to the size of the Eulerian
          graph $H$.
  Set label $l(v) \leftarrow 0$, for each vertex $v \in V$
  while $\exists$ edge $(u,v)$ such that $l(v) < l(u) - w(u,v)$ do
      $l(v) \leftarrow l(u) - w(u,v)$
  end
  $x(u,v) \leftarrow 0$, for edge $(u,v) \in DAG$
  $x(u,v) \leftarrow l(u) - l(v) + 1$, for edge $(u,v) \in H$
```

Even though the graph output by Algorithm 1 has negative edges, it does not have any negative cycles and Lemma 4 proves that the algorithm terminates. Theorem 2 proves that the labels produced by this algorithm are optimal labels for the primal, and hence, produce the optimal hierarchy.

Theorem 2. *x, l is a feasible solution to the primal. z is a feasible solution to the dual problem. Further,*

$$\sum_{(u.v)\in E} x(u,v) = \sum_{(u,v)\in E} z(u,v)$$

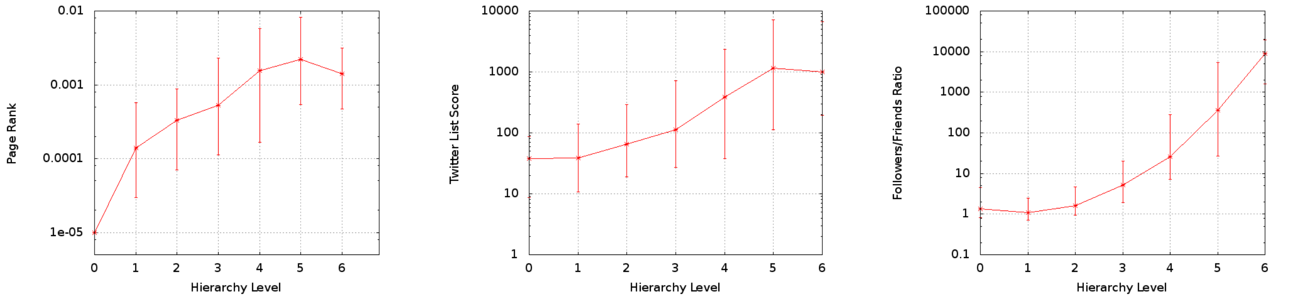
This shows that the value of the primal solution is equal to the value of a dual solution, which shows that both are optimal. We present the proof in Section A. We use this algorithm to find the hierarchy in various social networks.

4. EXPERIMENTS

In this section, we present the results of our experiments, which have the following goals:

- Validate that the notion of hierarchy we propose does correspond to real hierarchy based on ground truth.
- Validate that direction of edges does encode information about hierarchy.
- Compare how hierarchy emerges in online social graphs of different types of people, by using random graphs as baseline.
- Show how hierarchy emerges as the size of the social graph grows, for different online social networks.

[5] We say that a subgraph is Eulerian if the indegree of each vertex is equal to its outdegree. We *do not* impose the requirement that the subgraph be connected.



(a) **Page Rank** (b) **Twitter List Score** (c) **Twitter Follower/Friend Ratio**

Figure 4: **Correlation of hierarchy with popular metrics.**

4.1 Validation of the Hierarchy Measure

We want to validate that our measure of hierarchy corresponds with real hierarchy observed by humans. For this experiment, we collected a curated list of journalists in Twitter, which consists of 961 users. We compute the hierarchy using our measure; the computed hierarchy measure is 0.38. This indicates that there is a medium hierarchy in this graph. There are seven levels (strata) that users are assigned to in the optimal hierarchy. A higher level indicates people who enjoy higher social status.

Wikipedia notability To confirm that our computed hierarchy corresponds to the real hierarchy, we make use of Wikipedia to derive ground truth. Each node(journalist) is assigned a Wikipedia notability score, which is either No Entry (the person does not have an entry in Wikipedia), Small, Medium or Large (depending on the size of the Wikipedia entry). Figure 5 shows how our hierarchy measure compares with the ground truth obtained from Wikipedia. The figure shows that nodes with a low hierarchy level do not have a Wikipedia entry, and nodes higher up in the computed hierarchy are more likely to be noteworthy according to Wikipedia. This result lends credence to our measure of hierarchy.

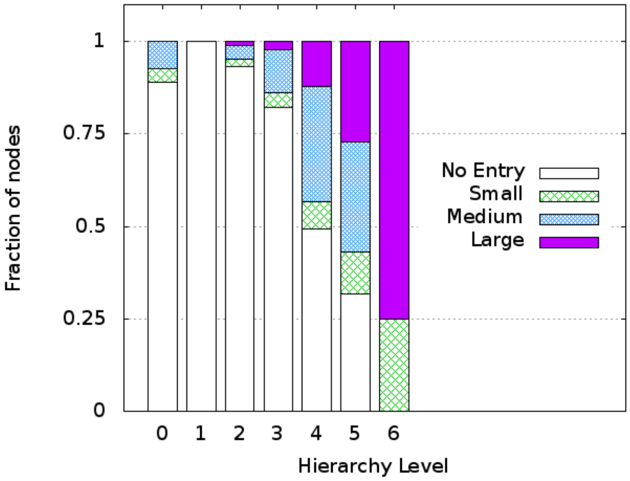


Figure 5: **Correlation with Wikipedia notability score.**

Correlation with well known measures: To get more insight into the factors that contribute to a node's hierarchy level, we measure the correlation of our computed hierarchy level for the journalists graph with the well known measures of social networks: pagerank, friend-follower ratio, and Twitter list score.

Figure 4a plots the median page rank (along with the 10th and the 90th percentile value) for each hierarchy level. The figure shows that people with a high page rank tend to be higher up in the social hierarchy level computed by our measure.

Figure 4b plots the correlation of hierarchy level with the Twitter list score, which corresponds to the number of user-generated Twitter lists that the node is a member of. Presence in a large number of user-generated Twitter lists indicates the user's popularity among Twitter users. The figure shows a high correlation of our computed node hierarchy with this measure of Twitter user popularity.

Finally, we measure the correlation with a popular twitter measure, Follower/Friend ratio, in Figure 4c. Popular users in Twitter tend to have an order of magnitude more followers than friends. We once again see a strong correlation between this measure and our computed hierarchy level.

4.2 Importance of Edge Direction

We now perform an experiment to validate that edge directions encode hierarchy information. For this, we use the college football dataset.

College Football Dataset: This dataset consists of all (American) Football games played by College teams in Division 1 FBS (the highest division, formerly called 1-A) during the last five years. The number of teams varies each year, but is between 150 and 200 for all five years. For each year, we consider the win-loss record of these teams. In the graph, each team is a node, and we place an edge from $u \rightarrow v$ if v played and defeated u during the season. We only consider the win-loss records and do not consider the margin of victory.[6] We also do not consider other factors like home advantage, though these would lead to better predictions. We end up with a directed unweighted graph representing win-loss record for a full season.

For each season, we find the optimal hierarchy. There is a

[6] The margin of victory is not considered even in the official BCS computer rankings, since "running up the scoreboard" is considered bad form and is discouraged.

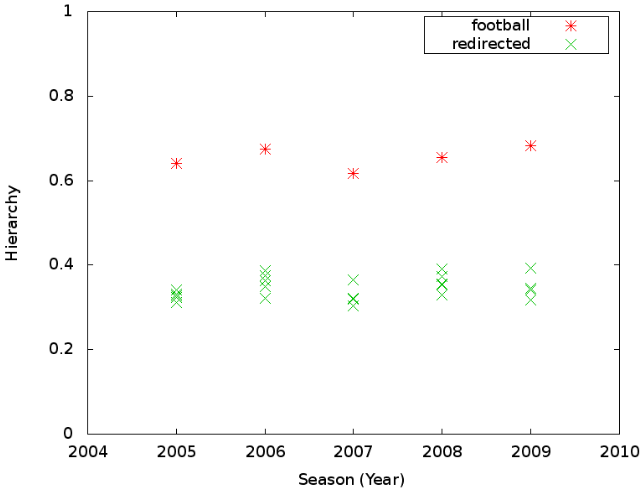


Figure 6: **Hierarchy in the College Football network.**

lot of variation between the quality of college football teams and we expect to see high hierarchy as observed in Figure 6.

Random redirection: Since the complete schedule is fixed before any games are played, we can compare the hierarchy we observe in the directed graph to the hierarchy if all games were decided by a random coin toss. In terms of the graph, this amounts to redirecting each edge in the network randomly. This technique allows us to observe the effect of the directions on hierarchy once the undirected graph is fixed. This random redirection would eliminate any quality difference between the nodes, and we now expect to see a much smaller hierarchy in the redirected graph. To observe the variance of the random redirection, we repeat this experiment five times. The hierarchy for these randomly redirected graphs is also shown in Figure 6. We see that the five randomly redirected graphs have very similar hierarchy, which is significantly lower than the real graph, showing that directions encode important information about hierarchy.

4.3 Hierarchy in Online Social Networks vs Random Graphs

To better understand how hierarchy emerges in a directed graph, we look at the behavior of hierarchy in random graphs to establish a baseline. We generate a random directed graph using the standard Erdös - Rényi random graph model as follows [7] . We fix a probability p that will decide the density of the graph. For each ordered pair of vertices (u, v), we put an edges from u to v with probability p. The outdegree distribution of nodes in this graph is a binomial distribution where each node has expected degree np.

Figure 7a shows that, for random graphs, the hierarchy starts out being large, and monotonically decreases as the size of the graph increases. We can also see that for small graph sizes, the variance is high, but as the graph size increases, the variance become very small.

We also conduct this experiment for different values of density, p. Figure 7b shows the outcome of the experiment with three different values of p. We see that for the same graph size n, hierarchy decreases with density. Hence, for random graphs, sparse graphs have higher hierarchy.

CURATED LISTS ON TWITTER: We now measure hierarchy for different online social networks. For this experiment, we collect different curated lists on Twitter that correspond to different types of users.

Famous people by field: Similar to the journalists dataset described earlier, we collect curated lists of people in the fields of Technology, Journalism, Politics, Anthropology, Finance and Sports. The smallest collection is Anthropology with fifty nine people and the largest is Technology with almost three thousand people.

Organizations: We also look at lists of employees of different organizations that have a team presence on Twitter. These include forrst, tweetdeck, ReadWriteWeb, wikia, techcrunch, Mashable, nytimes and Twitter. The smallest graph, forrst, has just seven employees. The largest is Twitter with two hundred and eighty two employees.

For each of these lists, we reconstruct the Twitter graph restricted to just these nodes, i.e. the nodes in the restricted graph are all the people on a particular list and there is an edge between two nodes if there is an edge between them on Twitter. For all these graphs, we calculate the hierarchy. Figure 8 shows a plot of hierarchy with respect to network size. We see that, among the fields, Sports has the highest hierarchy while Finance has the lowest one, and among organizations, the TODAYshow has the highest hierarchy while TweetDeck and ReadWriteWeb have the lowest one. Another trend that is observed is that, as the network size becomes larger, the hierarchy also increases. This is in contrast to random graphs, where the hierarchy decreases as the network size increases.

Wikipedia administrator voting dataset: Leskovec, Huttenlocher and Kleinberg [18, 19] collected and analyzed votes for electing administrators in Wikipedia. We use the wikivote dataset they collected and observe a very strong hierarchy in this dataset. This is consistent with the finding in [19] that *status* governs these votes more than *balance*.

4.4 Effect of Scaling on Social Hierarchy

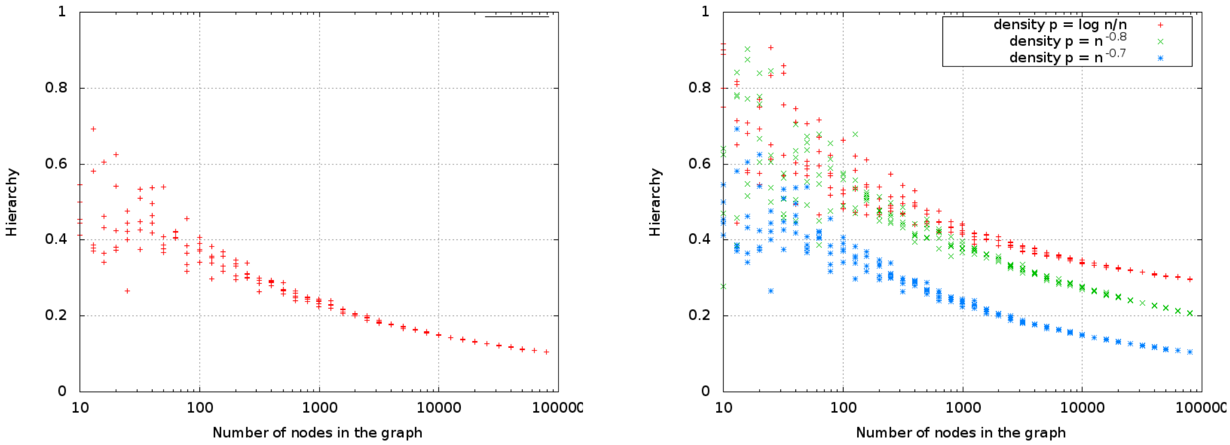
So far, we looked at small and medium sized graphs to get insight on how the measure of hierarchy works. We noticed that the hierarchy increases as the network size increases. Now, we shall consider large graphs to see the effect of scale on hierarchy in social networks.

For this experiment, we sample four popular directed social networks: Delicious, YouTube, LiveJournal and Flickr. The nodes are users and the edges indicate a follower relationship. We start from a single node and crawl nodes in the graph in a breadth first traversal. We plot hierarchy for different sizes of the graph. This is shown in Figure 9a.

We observe that, as a online social network grows in size, the hierarchy either stays the same or increases. This is in contrast with random graphs, where the hierarchy decreases as the graph grows in size. This suggests that, within small groups, social rank does not play an important role while forming connections but, as the group size increases, social rank becomes important to people while forming links.

This result corresponds with the intuition that, in social networks, people form connections with others based on their perceived level in the social hierarchy.

Further, we see that different social networks have different amount of hierarchy: YouTube has the lowest hierarchy, Flickr and LiveJournal have medium hierarchy, and Delicious has the highest hierarchy.



(a) **Effect of network size on hierarchy**

(b) **Effect of network density on hierarchy**

Figure 7: **Hierarchy in random graphs.**

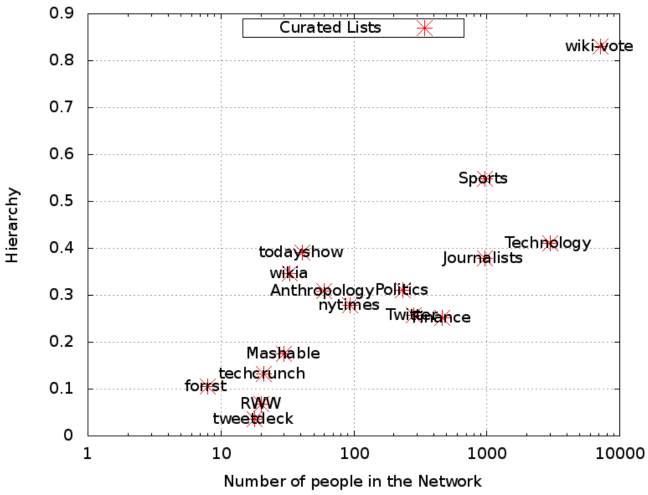


Figure 8: **Hierarchy in social network among famous people.**

Number of strata: Figure 9b plots the number of social strata in these four social networks, as we increase the graph size. We see that the number of strata stabilizes around seven for LiveJournal and around five for Flickr. YouTube has the lowest number of levels, and it also has the lowest hierarchy, while Delicious has the largest number of levels and also has the highest hierarchy. Compared to the number of nodes (100,000), the number of strata (< 15) is very low.

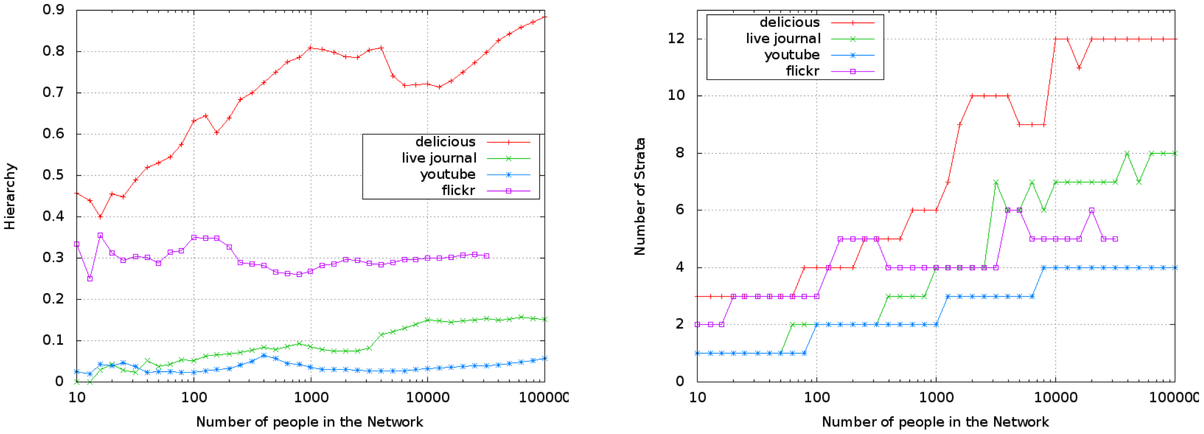
Rank distribution: Figure 10a plots the frequency distribution of people belonging to different social strata in a network, i.e., how many nodes belong to each stratum. We see that, in all the networks, most nodes have a low rank in the hierarchy (between one and three). A very small fraction of the nodes have ranks above four.

The exception to this is Delicious, which has a wider distribution of ranks. We show the exact probability distribution of the Delicious nodes in Figure 10b. The plot shows that a lot of delicious nodes have medium ranks in the hierarchy. But, even in Delicious, very few nodes belong to the highest stratum.

Agony distribution: Our measure of hierarchy is based on the intuition that people prefer to connect to other people who are in the same stratum or higher up. People who connect to others lower in the hierarchy incur agony. Figure 10c plots the distribution of agony among the nodes in the different networks that we study. The figure shows that most people incur very small amount of agony. There are a few people who incur a lot of social agony. These people tend to follow a lot of people who are lower than them in the hierarchy.

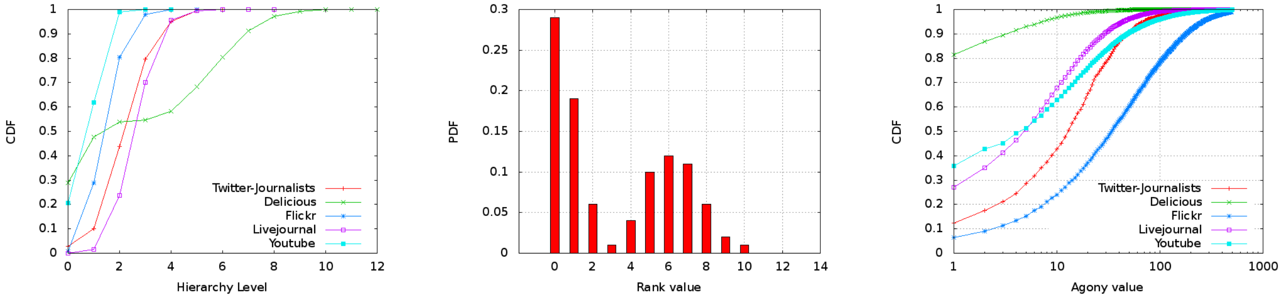
Random redirection: We now study whether the hierarchy for each of these social networks is more or less than that observed in a randomly directed graph with the same underlying structure. To do this, we take each graph and randomly change the direction of each edge. Hence, we keep the undirected graph the same, but change the direction of the edge. In Figure 11, we show the importance of edge directions to hierarchy for these social networks and the effect of randomly redirecting the edges.

Among the social networks we studied, Delicious has the highest hierarchy. The networks starts out with medium hierarchy and it keeps increasing. The Delicious graph has almost perfect hierarchy at size 100,000. The hierarchy in the randomly redirected graph, shows a similar overall pattern but with low hierarchy. Delicious also has the most number of levels in the hierarchy. YouTube, on the other hand, has the lowest hierarchy, which is even lower than the hierarchy observed if the edges were randomly oriented. The likely reason for this is that YouTube has a good search index and the preferred way of getting to videos is through search. Hence, social connections become less important and people do not connect to each other based on rank. In Flickr, the hierarchy largely remains the same even as the graph becomes large. However, the hierarchy in the redirected graph decreases sharply. In LiveJournal, the hierarchy starts out being very low and increases slowly with graph size. The randomly redirected graph on the other hand shows exactly the opposite behavior, consistent with the behavior of random graphs that we saw earlier.



(a) **Effect of network size on value and variance of hierarchy** (b) **Effect of network size on number of strata in the hierarchy**

Figure 9: **Effect of network size on hierarchy.**



(a) **Cumulative distribution of ranks** (b) **Probability distribution for the Delicious graph** (c) **Distribution of Agony among nodes**

Figure 10: **Distribution of ranks among nodes.**

5. RELATED WORK

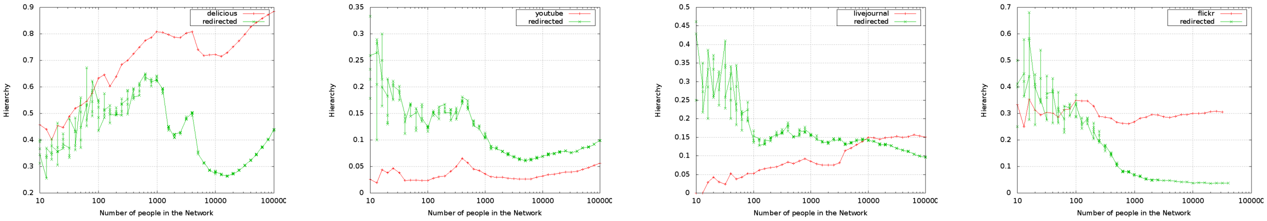
Early efforts to find the hierarchy underlying social interactions followed from observations of dominance relationships among animals. Landau [17] and Kendall [15] devised statistical tests of hierarchy for a society, but with the necessary assumption that there exists a strict dominance relation between all pairs of individuals, and that the relations are transitive (i.e. no cycles). Although de Vries [5, 6] expanded the Landau and Kendall measures by allowing ties or missing relationships, his algorithms are feasible only on small graphs.

The hierarchy underlying a social network can be used in recommending friends (the link prediction problem [20]) and in providing better query results [16]. There exist link-based methods of ranking web pages [11]. Maiya and Berger-Wolf [21] begin from the assumption that social interactions are guided by the underlying hierarchy, and they present a maximum likelihood approach to find the best interaction model out of a range of models defined by the authors. In the same vein, Clauset, Moore, and Newman [4] use Markov Chain Monte Carlo sampling to estimate the hierarchical structure in a network. Rowe et. al. [22] defined a weighted centrality measure for email networks based on factors such as response time and total number of messages, and tested their algorithm on the Enron email corpus. Leskovec, Huttenlocher, and Kleinberg recently brought attention to signed network relationships (e.g. "friend" or "foe" in the Epinions online social network) [19] and presented a way to predict whether a link in a signed social network is positive or negative [18].

The closest to our problem in the computer science literature is the minimum feedback arc set problem. In the minimum feedback arc set problem, we are given a directed graph G and we want to find the smallest set of edges whose removal make the remaining graph acyclic. This is a well known NP-hard problem and is in fact NP-hard to approximate beyond 1.36 [14]. Poly-logarithmic approximation algorithms are known for this problem [8].

6. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we introduced a measure of hierarchy in directed social networks. We gave an efficient algorithm to find the optimal hierarchy given just the network. We also showed the emergence of hierarchy in multiple online social networks: in contrast to random networks, social networks have low hierarchy when they are small and the hierarchy



(a) **Delicious** (b) **YouTube** (c) **LiveJournal** (d) **Flickr**

Figure 11: **Effect of directed edges.**

increases as the network grows. We showed that there are a small number of strata, and this number does not grow significantly as the network grows.

An interesting future direction is to study the emergence of hierarchy over time in different social networks. Another direction of future work is to use our measure of hierarchy to develop better ranking algorithms.

Acknowledgments

We would like to thank Alantha Newman and Michael Saks for helpful discussions. We will also like to thank the anonymous referees for their suggestions. This work was supported in part by the NSF under grants CCF-0728937, CCF-0832787, CCF 0832795, CNS-0831268, IIS-0414852 and by the Summer 2010 DIMACS REU program.

7. REFERENCES

[1] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabási. The diameter of the world wide web. *Nature*, 401:130–131, 1990.

[2] Michael C. Appleby. The probability of linearity in hierarchies. *Animal Behavior*, 31(2):600–608, 1983.

[3] Albert-Laszlo Barabási. The origin of bursts and heavy tails in humans dynamics. *Nature 435, 207*, 2005.

[4] Aaron Clauset, Cristopher Moore, and Mark Newman. Structural inference of hierarchies in networks. In *International Conference on Machine Learning, Workshop on Social Network Analysis*, June 2006.

[5] Han de Vries. An improved test of linearity in dominance hierarchies containing unknown or tied relationships. *Animal Behavior*, 50:1375–1389, 1995.

[6] Han de Vries. Finding a dominance order most consistent with a linear hierarchy: A new procedure and review. *Animal Behavior*, 55(4):827–843, 1998.

[7] Paul Erdös and Alfréd Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5, 1960.

[8] Guy Even, Joseph (Seffi) Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multi-cuts in directed graphs. *Integer Programming and Combinatorial Optimization*, pages 14–28, 1995.

[9] Eugene F. Fama and Kenneth R. French. Testing trade-off and pecking order predictions about dividends and debt. *Review of Financial Studies 15, 1-33*, 2002.

[10] Murray Z. Frank and Vidhan K. Goyal. Testing the pecking order theory of capital structure. *Journal of Financial Economics 67, 217-248*, 2003.

[11] Lise Getoor and Christopher P. Diehl. Link mining: A survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, December 2005.

[12] Mark Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.

[13] http://www.answers.com/topic/social-stratification-1.

[14] Vigo Kann. *On the approximability of NP-complete optimization problems.* PhD thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, May 1992.

[15] M. G. Kendall. *Rank correlation methods.* Charles Griffin, London, 1962.

[16] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 1999.

[17] H. G. Landau. On dominance relations and the structure of animal societies: I. effect of inherent characteristics. *Bulletin of Mathematical Biophysics*, 13(1):1–19, 1951.

[18] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *ACM International Conference on World Wide Web (WWW)*, 2010.

[19] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *ACM SIGCHI Conference on Human factors in computing systems*, 2010.

[20] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *International Conference on Information and Knowledge Management*, 2003.

[21] Arun S. Maiya and Tanya Y. Berger-Wolf. Inferring the maximum likelihood hierarchy in social networks. In *Computational Science and Engineering*, August 2009.

[22] Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J. Stolfo. Automated social hierarchy detection through email network analysis. In *Joint 9th WEBKDD and 1st SNA-KDD Workshop*, 2007.

[23] Schjelderup-Ebbe T. Contributions to the social psychology of the domestic chicken. *Reprinted from Zeitschrift fuer Psychologie, 1922, 88:225-252.*, 1975.

[24] Travers and Milgram. An experimental study of the small world problem. *sociometry*, 32:425–443, 1969.

[25] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature 393, 440-442*, 1998.

APPENDIX

A. PROOFS

We shall now prove Theorem 1 and 2. We start with proving that Algorithm 1 produces a feasible dual solution.

Lemma 1. *Let H be the subgraph of G that contains the reverse of all (and only those) edges labeled $+1$ by Algorithm 1. Then, for each vertex v : $indeg_H(v) = outdeg_H(v)$*

Proof. Let H be the subgraph of G consisting of all the $+1$ edges. Initially, H is the empty graph. We establish the following loop invariants.

- All edges with label -1 belong to G. The reverse of all edges labeled +1 belong to G.

- $\forall v \in V : indeg_H(v) = outdeg_H(v)$.

These are true at the start. If we prove these for each iteration of the loop, they will imply the lemma.

The first assertion is true, since we initialize the label all edges to -1 and whenever we reverse an edge, we also change its sign.

Now, we shall prove the second assertion. Suppose this is true at some middle state. Algorithm 1 finds a directed cycle C in G, removes edges with label +1 from H and adds edges with label -1 to H. For any vertex v, the edges e_1, e_2 adjoining it in C can have any of the four ±1 label combinations. When they have labels +1,+1, the indegree and outdegree both decrease by 1 and when they have labels -1,-1, both the the indegree and outdegree increase by 1. When the labels are -1,+1, we remove edge e_2 from H, which was pointing into v and add edge e_1, which now points into v. Similarly, if the labels were +1,-1 then we remove edge e_2, which was pointing out of v in H and add edge e_1, which now points out of v. So, the indegree or outdegree does not change in these cases. This proves the lemma. □

Lemma 2. *H is the maximal such subgraph.*

Proof. Let T be another subgraph, such that number of edges of T is greater than number of edges of H. Let $rev(H)$ be the graph with edges of H reversed. Consider the graph P obtained by taking the disjoint union of edges of $rev(H)$ and T and removing cycles of length two with one edge from H and the other from T. Set the label of edges in $rev(H \setminus T)$ to 1, and the label of edges in $T \setminus H$ to -1. The edges in $T \cap H$ become cycles of length two in $rev(H) \cup T$ and are removed from P. Observe that P occurs as a subgraph (along with the labels) of G at the termination of Algorithm 1.

P is Eulerian since both $rev(H)$ and T are Eulerian and we only remove cycles from their disjoint union. Hence, we can construct a cycle cover of the edges of P. But the total number of negative edges of P is greater than the number of positive edges. Hence, there exists a negative cycle in this cover. Since P is a subgraph of G, this also implies that there exists a negative cycle in G at the end of the Algorithm 1, which is a contradiction. □

Lemma 3. *Algorithm 1 terminates in $O\left(m^2 n\right)$ time.*

Proof. In each iteration of the loop, the number of edges with label +1 increases by at least 1. The total number of edges is upper bounded by m. Hence, there are at most m iterations. Each iteration calculates a negative cycle detection algorithm, which can be done by Bellman-Ford and takes time $O\left(mn\right)$. Hence, the total time is at most $O\left(m^2 n\right)$. □

Hence, we have proved Theorem 1.

Theorem 1. *Let H be the subgraph of G that contains all (and only those) edges labeled +1 by Algorithm 1. Then, for each vertex $v : indeg_H(v) = outdeg_H(v)$. Also, for every subgraph T of G with the property that $v : indeg_T(v) = outdeg_T(v)$, number of edges in H is greater than the number of edges in T.*

Theorem 1 shows that Algorithm 1 calculates the optimal integral dual solution. We now prove properties of Algorithm 2. First we prove that Algorithm 2 terminates.

Lemma 4. *If the input graph to Algorithm 2 does not contain negative cycles, then Algorithm 2 terminates.*

Proof. All nodes have label 0 at the start of the algorithm. Consider the shortest paths between all pairs of vertices. Since there are no negative cycles, these are well defined. Let m be the minimum length among all shortest paths. Note that m will be negative, since the graph contains negative edges. We claim that $-l$ is an upper bound on the label that any vertex can get. If any vertex gets a higher label, we can trace the set of edges that were used to get to that label, and these would give a shorter path than l, which is a contradiction. □

The next lemma helps us prove Theorem 2.

Lemma 5. *For each edge $(u, v) \in DAG$, $l(v) \geq l(u)+1$. For each edge $(u, v) \in$ the Eulerian subgraph H, $l(u)-l(v)+1 \geq 0$*

Proof. Suppose $(u, v) \in DAG$. Then, $w(u, v) = -1$. Hence, at the end of Algorithm 2, the condition $l(v) \geq l(u) - (-1)$ is satisfied. Similarly, for edge (u, v) in H, $w(v, u) = 1$. Hence, at the end of Algorithm 2, the condition $l(u) \geq l(v) - 1$ is satisfied. □

The above lemma shows that for an edge (u, v) in the DAG, we can set the primal variables $x(u, v) = 0$ and for an edge (u, v) in the Eulerian subgraph we set $x(u, v) = l(u) - l(v) + 1 \geq 0$ by Lemma 5.

Theorem 2. *x, l is a feasible solution to the primal. z is a feasible solution to the dual problem. Further,*

$$\sum_{(u.v)\in E} x(u,v) = \sum_{(u,v)\in E} z(u,v)$$

Proof. Lemma 5 proves that x, l is a feasible primal solution. Theorem 1 shows that z is a feasible dual solution. Now, we show that the value of the primal solution is equal to the value of a dual solution, which shows that both are optimal.

$$\text{Value of the primal solution} = \sum_{(u,v)\in E} x_{(u,v)}$$

$$= \sum_{(u,v)\in E} \max\{0, l(u) - l(v) + 1\}$$

$$= \sum_{(u,v)\in DAG} \max\{0, l(u) - l(v) + 1\} +$$

$$\sum_{(u,v)\in \mathrm{H}} \max\{0, l(u) - l(v) + 1\}$$

$$= 0 + \sum_{(u,v)\in \mathrm{H}} l(u) - l(v) + 1 \qquad \text{(By Lemmas 5)}$$

$$= \sum_{C\in\mathcal{C}} \sum_{(u,v)\in C} l(u) - l(v) + 1$$

(where $\mathcal{C}$ is some cycle cover of the Eulerian subgraph)

$$= \sum_{C\in\mathcal{C}} |C| \left(\text{For any cycle } C, \sum_{(u,v)\in C} l(v) - l(u) + 1 = |C| \right)$$

= number of edges in the Eulerian subgraph

= Value of the dual solution

This proves that x, l is an optimal primal solution. □

This shows that the linear program has an integral optimal solution and that Algorithms 1, 2 calculate the optimal solution to the integer program we started out with.