# Emergence of hierarchy in directed online social networks

Mangesh Gupte
Dept. of Computer Science
Rutgers University

Pravin Shankar
Dept. of Computer Science
Rutgers University

Jing Li
Dept. of Economics
MIT

S. Muthukrishnan
Dept. of Computer Science
Rutgers University

Liviu Iftode
Dept. of Computer Science
Rutgers University

## 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 people, and at different scales. We conjecture that people form connections in 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 social network, and present an efficient algorithm to compute this measure. We validate our measure using ground truths including Wikipedia notability score. We use this measure to study hierarchy in several directed online social networks including Twitter, Delicious, Flickr, and curated lists of several types of people based on different occupations, and different organizations. Our experiments how hierarchy emerges as the networks grows on different online social networks. We show that the degree of stratification in a network is bounded and does not increase after the graph has become large.

## 1. INTRODUCTION

Social stratification has existed for as long as humans have existed. 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 [12]. Social hierarchy 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 discernable.

The popularity of online social networks has created the opportunity to study sociological phenomenon at scales that were earlier unfathomable. Phenomenon such as small diameter in social networks [23] and strength of weak ties [11] have been revisited in the light of the large data now available about people and their connections [1, 24, 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 [22] where each hen is aware of its place among the hierarchy and there have been various papers which investigate the importance of such a hierarchy [9, 8]. 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 the social hierarchy. Most of the popular online social networks today, such as Twitter, Flickr, Youtube, Delicious and Livejournal contain directed edges. In fact the only popular social network where edges are undirected is Facebook. We conjecture that there is an inherent "social rank" that each person enjoys and that, when making connections, everyone in the network is aware of their rank as well as the rank of people they connect to.

Given a social graph, we do not know the ranks of people in the network, we can only observe the links. The existence of a link indicates a social rank recommendation; a link $u \rightarrow v$ ($u$ is a follower of $v$) indicates a social recommendation of $v$ from $u$. If there is no reverse link from $v$ to $u$, it indicates that $v$ is higher up in the hierarchy than $u$. We assume that in social networks, when people connect to other people with lower hierarchy, this causes them *social agony*. To infer the 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, and curated lists of several types of people based on different occupations, and different organizations.

Our experiments show how hierarchy emerges in online social networks. For small networks, people know each other and consequently hierarchy is not observed. The network needs to grow for hierarchy to emerge. However, this does not continue indefinitely, and after a certain point, the hierarchy stabilizes. We observe that there is a small number of levels that users are assigned 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 networks grows on different online social networks.
4. We show that the degree of stratification in a network is bounded and does not increase after the graph has become large.

## 2. HIERARCHY IN DIRECTED SOCIAL NETWORKS

There are several definitions of hierarchy like tree organizations and chains. We take hierarchy to mean a partially ordered set. This is a very general definition that includes chains as well as trees. We can view a partially ordered set as a graph, where each element of the set is a node and the partial ordering ($u \geq v$) 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 an examples of perfect hierarchy. We now define a measure of hierarchy for directed graphs that might contain cycles.

Given a network $G = (V, E)$ 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. For social networks, this could represent the reputation or social status of the person.

In social networks, where nodes are aware of their ranks, we expect that higher rank nodes do not 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. In fact, we shall assume that the agony caused to $u$ by each such reverse edge is equal to $r(u) - r(v) + 1$[1] [2]. Hence, the agony to $u$ caused by edge $(u, v)$ relative to the rank $r$ is $\max(r(u) - r(v) + 1, 0)$

[1]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.

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

We define the agony in the network relative to the rank $r$ 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 or 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 the ranking $r$ that minimizes the total agony in the graph. We define the agony of $G$ as the minimum agony over all possible rankings $r$.

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

We prove in Section 3.1 (Equation 1) that the minimum agony in the graph is bounded by the number of edges in the graph, $m$. This motivates our definition of hierarchy in the graph.

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

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

We make a few observations about the amount of agony in the graph.

1. For any graph $G$, the hierarchy $h(G)$ lies in $[0, 1]$. This follow from the fact that $A(G)$ lies in $[0, m]$. (Equation 1).
2. DAGs have perfect hierarchy. For any DAG $G$, $h(G) = 1$. This is achieved by setting $r(v) > r(u) + 1$ for each edges $(u, v)$ in the DAG.
3. Directed cycles have no hierarchy. If $G$ is a collection of edge disjoint directed cycles, then $h(G) = 0$. We show this in Section 3.1.

## 3. EFFICIENTLY MEASURING HIERARCHY

Note that the number of rankings $r$ is exponentially large, so we need an efficient way to search among them. In Section 3.1 we present an efficient algorithm to find a ranking which gives the highest hierarchy for any directed graph $G$.[3]

### 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

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

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

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

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

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

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

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:

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

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

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

We can reinterpret the dual program as finding an Eulerian subgraph of the original graph when we restrict the dual variables to be 0 or 1 instead of in the range $[0,1]$ . We say that a graph is Eulerian if indegree of each vertex is equal to the outdegree. (There is also the requirement that the graph is connected which we ignore.)

The reinterpretation gives us insight into the primal solution. By weak duality, the value of the primal is lower bounded by the value of the dual, hence the primal value cannot become smaller than the size of the maximum Eulerian subgraph of the given graph (in terms of number of edges). In fact, if the original graph $G$ is Eulerian, this gives a lower bound of $m$. Equation 1 demonstrated 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 which 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. In the next section, we give a combinatorial algorithm that find the best ranking. In fact, we shall prove that the linear program has an integral optimal solution.

Algorithm 1 gives a way to construct an Eulerian subgraph of $G$. Theorem 1 prove the correctness of Algorithm 1 (that the subgraph it constructs is Eulerian) and also shows that the subgraph is optimal. We give a proof of Theorem 1 in Section A.

Algorithm 2 constructs an optimal integral solution to the dual. We then use the dual solution to come up with a primal solution of the same value which, by duality, proves that both are optimal.

**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(e) = -1
**while** $\exists$ *a negative cycle* $C$ **do**
  Reverse all edges in $C$
  Reverse the sign of all edges in $C$
**end**
DAG $\leftarrow$ All edges labeled -1
Eulerian graph $\leftarrow$ All edges labeled +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 $T$.*

To find the optimal value of hierarchy in the graph $G$, we need to find a ranking $r$ of the nodes and calculate the agony values on each edge $x(i,j)$. Algorithm 2 shows how we can labels the nodes.

Observe that the input graph to Algorithm 2 is the graph output by Algorithm 1. Hence, even though the graph has negative edges, it does not have any negative cycles and so

**Algorithm 2**: Label the graph given as a decomposition of the Eulerian graph and a DAG

**Input**: A Graph $G = (V, E)$ with weights +1, -1 such that the edges with label -1 form a DAG and Edges with label +1 form an Eulerian Graph.[a].

**Output**: A labeling of all vertices of $G$, such that the measure on the $G$ with the given labels, is equal to the size of the Eulerian Graph.

Set label $l'(v) \leftarrow 0$, for each vertex $v \in V$,
**while** *there exists an edge* $(u, v)$ *such that* $l'(v) > l'(u) + w(u, v)$ **do**
  $l'(v) \leftarrow l'(u) + w(u, v)$
**end**
$L \leftarrow$ the largest vertex label in $G$
Set $l(v) = L - l'(v)$

[a]This is the graph output by Algorithm 1 and so the edges labeled +1 are reverse of those in the original graph

the algorithm terminates. Theorem 2 proves that the labels produced by this algorithm are the 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 use this algorithm to find the hierarchy in various social networks.

# 4. EXPERIMENTS

In this section, we present the results of our experiments evaluating the measure of hierarchy that we introduced. Following are the goals of our evaluation:

- Validate that the notion of hierarchy we propose does correspond to real hierarchy based on ground truths.
- 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.

## 4.1 Validation of Hierarchy Measure

We want to validate that our measure of hierarchy corresponds with real hierarchy that is 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; and the computed hierarchy measure is 0.38. This indicates that there is a medium hierarchy in this graph. Specifically, there are seven levels of hierarchy. A higher hierarchy 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 1 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.

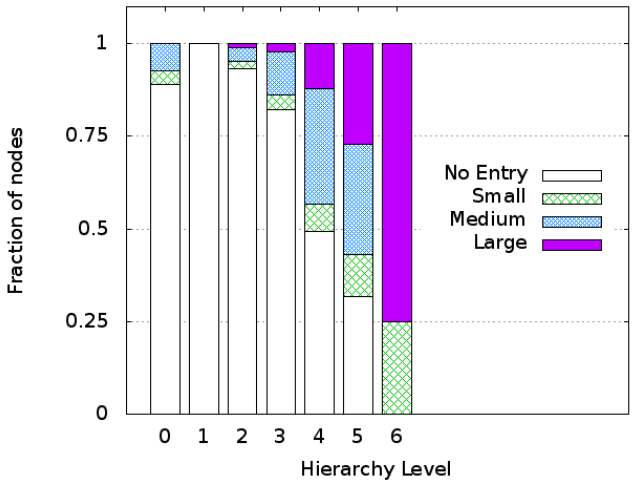



**Figure 1: Correlation with Wikipedia Notability Score.**

*Agony distribution* Our measure of hierarchy is based on the intuition that people prefer to connect to other people who are in the same hierarchy or higher up. People who connect to lower hierarchy incur agony. Figure 2 plots the distribution of agony among all the journalists in our network. The figure shows 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 hierarchy.

*Correlation with well known measures* We further measure the correlation of our computed hierarchy level for the different journalists in this graph with other well known measures of social networks, to get more insight into the factors that contribute to a node's hierarchy level.

Figure 3(a) 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 3(b) 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

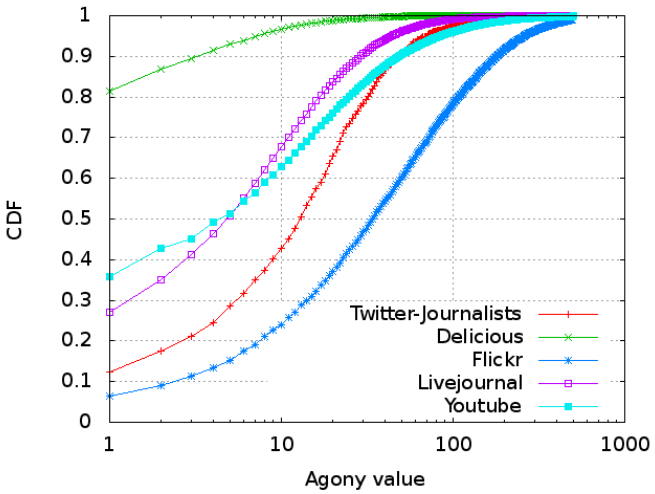



**Figure 2: Distribution of Agony among nodes**

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 3(c). 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

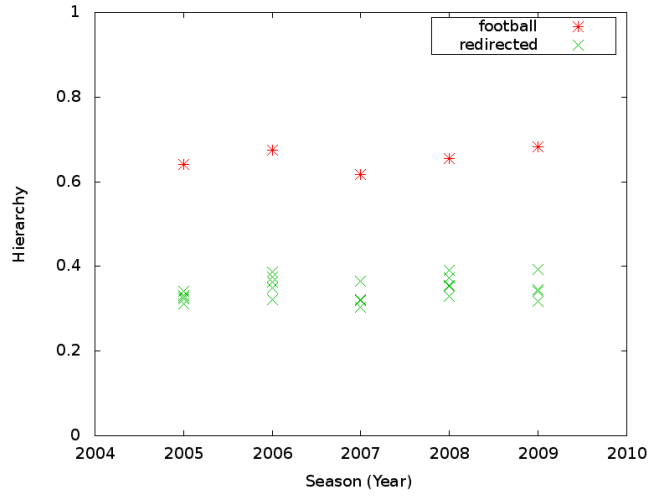



**Figure 3: Hierarchy in the College Football Network**

College Football Dataset: For this experiment, we look at 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 $u \to v$ if $v$ played and defeated $u$ during the season. We only consider the win-loss records and do not consider the margin of victory.[4] 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 lot of variation between the quality of college football teams and we expect to see high hierarchy as observed in Figure 3.

*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 5 times. The hierarchy for these randomly redirected graphs is also shown in Figure 3. We see that the five randomly redirected graphs have very similar hierarchy, which is significantly lower than the real graph, hence 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 first look at the behavior of random graphs to establish the baseline. We generate a random directed graph as follows. 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 $np$.

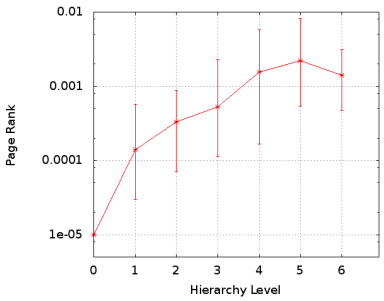
Figure 4(a) shows that for random graphs, the hierarchy starts out being large, and monotonically decreases as the size of the graph increases. We can also see that for small graph sizes, the variance is high, but as the graph size increases, the variance become very small and all random graphs behave essentially the same. This describes the change in hierarchy with size.

We also conduct this experiment for different values of density, $p$. Figure 4(b) 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.

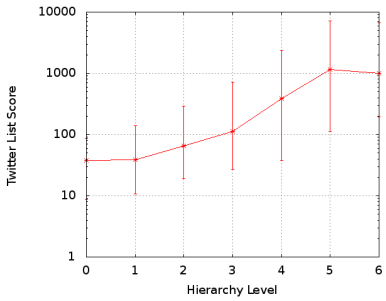
Curated Lists on Twitter: We now measure hierarchy for different online social networks. For this experiment, we collect different curated lists on Twitter that correspond to different types of users.

*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

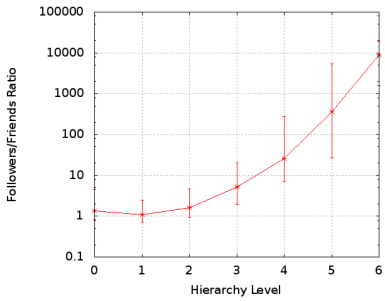
[4]The margin of victory is not considered even in the official BCS computer rankings, since "running up the scoreboard" is considered bad form and is discouraged.



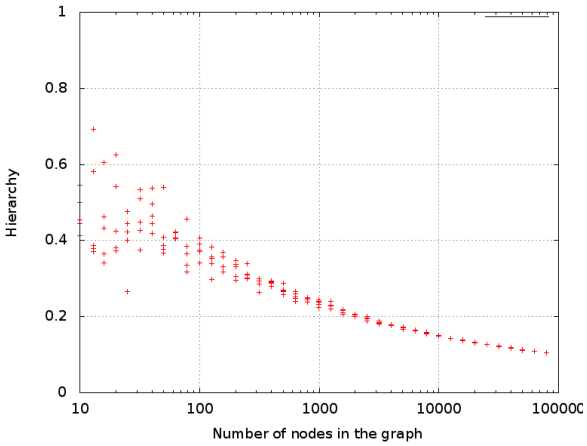

(a) **Correlation with Page Rank**



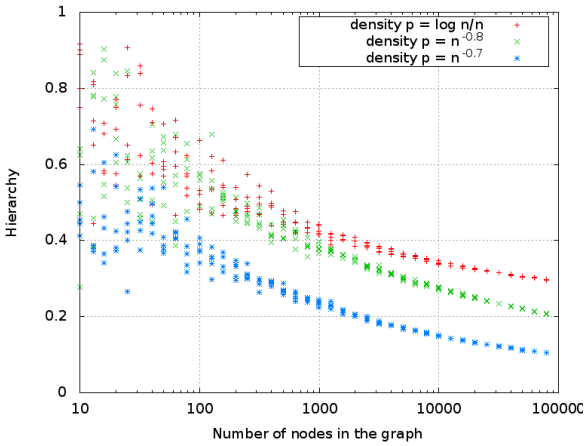

(b) **Correlation with Twitter List Score**




(c) **Correlation with Twitter Follower/Friend Ratio**




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




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

**Figure 4: Hierarchy in random graphs**

fifty nine people and the largest is Technology with almost three thousand people.

*Organizations* We also look at lists of employees of different organizations which 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 is 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 plot the hierarchy which is shown in Figure 5. 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.

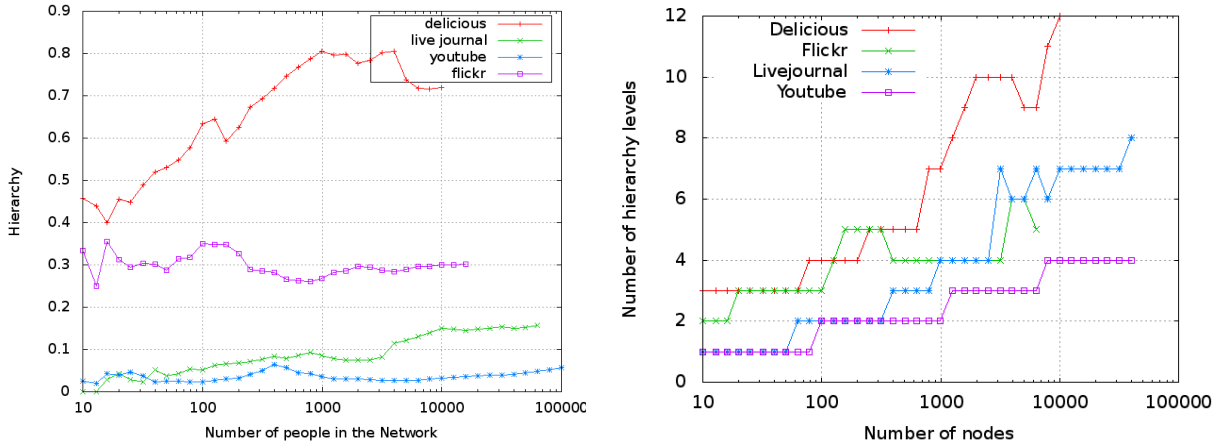
*Wikipedia administrator voting* Leskovec, Huttenlocher and Kleinberg. [17, 18] collected and analyzed votes for electing administrators in wikipedia. We use the voting dataset they collected and observe a very strong hierarchy in this dataset. This is consistent with the finding in [18] that "status" 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 in 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 indicates that $u$ follows $v$. 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 6(a).

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




(a) **Effect of network size on value and variance of hierarchy**

(b) **Effect of network size on number of levels in the hierarchy**

**Figure 6: Effect of network size on hierarchy**

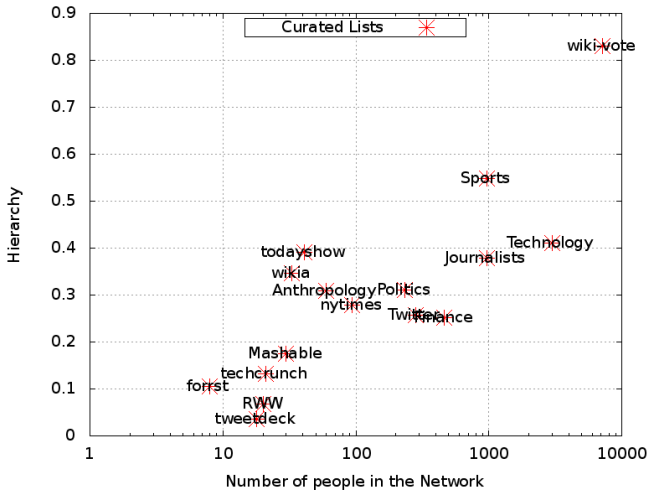



**Figure 5: Hierarchy in Social Network among Famous People**

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.

*Number of levels* Figure 6(b) plots the number of levels in the hierarchy (number of social strata) in these four social networks, for different graph size. We see that the number of levels stabilizes around seven for Livejournal and around five for Flickr. Youtube has the lowest number of levels as it also has the lowest hierarchy while delicious has the largest number of levels and also has the most hierarchy.

*Rank distribution* Figure 7(a) plots the distribution of social strata in different networks, i.e., how many nodes belong to each strata. We see that, in all the networks, most nodes have a low hierarchy level (between one and three). A very small fraction of the nodes belong to the highest hierarchy.

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 7(b). 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 hierarchy.

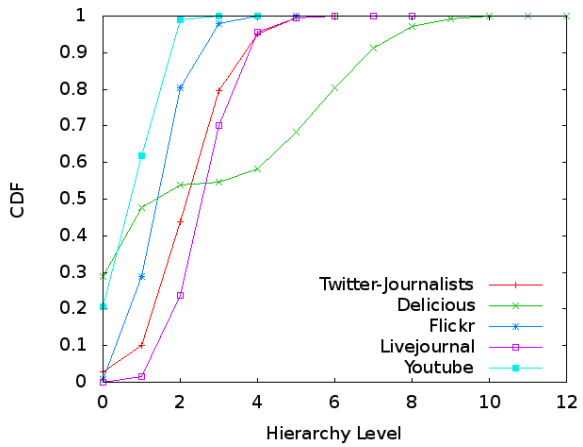
*Random redirection* We now study whether the hierarchy for each of these social networks is more or less than that observed in a randomly directed graph with the same underlying structure. To do this, we take each graph and randomly change the direction of each edges. Hence, we keep the undirected graph the same, but change the direction of the edge. In Figures 4.4 we show the effect of edge directions on hierarchy for these social networks, as they grow in size.

*Delicious* Among the social networks we studied, delicious has the highest hierarchy. The hierarchy increases until a few thousand nodes and then remains constant. Delicious also has the most number of levels in the hierarchy.

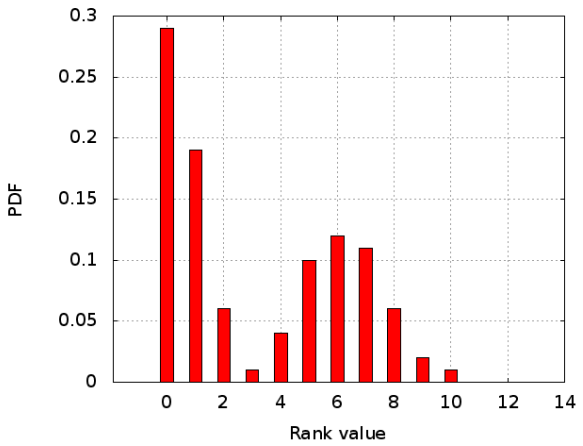
*Youtube* Youtube, on the other hand, has the lowest hierarchy, which is even lower than the hierarchy observed in a random graph. The likely reason for this is that, in Youtube, users subscribe to other users mostly based on the content and not the identity of the person. Since a good search index is available on youtube, the social connections become less important, which manifests as low social hierarchy.

## 5. RELATED WORK

Early efforts to find the hierarchy underlying social interactions followed from observations of dominance relationships among animals. Landau [16] and Kendall [14] 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 miss-

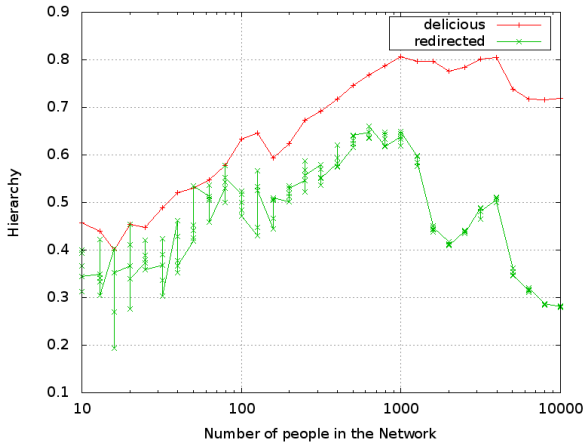



(a) **Cumulative distribution for all graph**

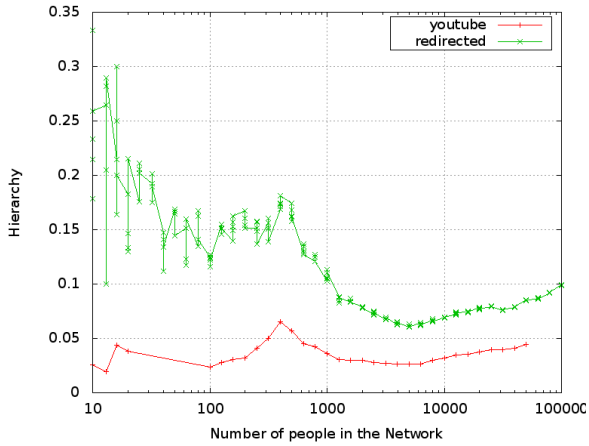



(b) **Probability distribution for the Delicious graph**

**Figure 7: Distribution of ranks among nodes**




(a) **Delicious has the highest hierarchy**




(b) **Youtube has the lowest hierarchy**

**Figure 8: Effect of directed edges on Delicious and Youtube**

ing 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 [19]) and in providing better query results [15]. There exist link-based methods of ranking web pages [10]. Maiya and Berger-Wolf [20] 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. [21] 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 bring attention to signed network relationships (e.g. "friend" or "foe" in the Epinions online social network) [18] and present a way to predict whether a link in a signed social network is positive or negative [17].

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 [13]. Polylogarithmic approximation algorithms are known for this problem [7].

## 6. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we introduced 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

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 evolution 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.

# 7. REFERENCES


[1] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. 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, May 1983.

[3] A.-L. BarabÃąsi. 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 *Proc. 23rd Intl. 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] Guy Even, Joseph (Seffi) Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multi-cuts in directed graphs, extended summary. *Integer Programming and Combinatorial Optimization*, pages 14–28, 1995.

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

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

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

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

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

[13] 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.

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

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

[16] 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.

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

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

[19] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proc. 12th Intl. Conference on Information and Knowledge Management*, 2003.

[20] Arun S. Maiya and Tanya Y. Berger-Wolf. Inferring the maximum likelihood hierarchy in social networks. In *Proc. 12th IEEE Intl. Conference on Computational Science and Engineering*, August 2009.

[21] 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.

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

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

[24] 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 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 whenever we reverse an edges, 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 $v$, if the edges $e_1, e_2$ adjoining it in $C$ can have any of the four $\pm 1$ label combinations. Suppose they have labels +1,+1. Then the indegree and outdegee both decrease by 1 and when they have labels -1,-1, the the indegree and outdegee increase by 1. If the labels were -1,+1 then we remove edge $e_2$ from $H$ which was pointing into $v$ in $H$ graph and add edge $e_1$ which is now also pointing 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 also pointing out of $v$. So, the indegree or outdegree do not change in these case. This proves the lemma. □

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

*Proof.* Let $T$ is 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 with edges $rev(H) \cup T$ obtained by removing cycles of length 2. At the end of Algorithm 1 the labels of edges in $H \setminus T$ is 1, edges in $T \setminus H$ have label $-1$ and those in $T \cup H$ have weight 0. Since, both $rev(H)$ and $T$ are Eulerian, we can construct a cycle cover of the edges. But the total number of negative edges is greater than the number of positive edges. Hence, there exists a negative cycle in this cover. But, this also implies that there exists a negative cycle 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 while loop, the number of edges with label $+1$ increases by at least 1. But 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 $O(mn)$. 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. To find the ranking, we need to get the optimal integral primal solution. We now give an algorithm to find the labels for each vertex, from the $\pm 1$ edge labels given by Algorithm 1.

Observe that the input graph to Algorithm 2 is the one output by Algorithm 1. Hence, even though it has negative edges, it does not have negative cycles, and so the Algorithm 2 will terminate. The next two lemmas help us prove Theorem 2.

**Lemma 4.** *For each edge $(u, v) \in DAG$, $l(v) \geq l(u) + 1$*

*Proof.* $w(u, v) = -1$ and $l'$ gives shortest path labels. Hence,

$$\begin{aligned} & l'(v) \leq l'(u) - 1 \\ \implies & L - l'(v) \geq L - l'(u) + 1 \\ \implies & l(v) \geq l(u) + 1 \end{aligned}$$

□

**Lemma 5.** *For each edge $(u, v) \in G$, where $(u, v) \in$ Eulerian Graph, $l(u) - l(v) + 1 \geq 0$*

*Proof.* $w(v, u) = 1$ and $l'$ gives shortest path labels. Hence,

$$\begin{aligned} & l'(u) \leq l'(v) + 1 \\ \implies & L - l'(u) \geq L - l'(v) - 1 \\ \implies & l(u) \geq l(v) - 1 \end{aligned}$$

□

The above two lemmas show 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 $x(u, v) = l(v) - l(u) + 1 \geq 0$ 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.* Lemmas 4, 5 prove that $x, l$ is a feasible solution. Theorem 1 shows that $z$ is feasible. 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.

Value of the primal solution

$$\begin{aligned} &= \sum_{(u,v) \in E} x_{(u,v)} \\ &= \sum_{(u,v) \in E} \max\{0, l(u) - l(v) + 1\} \\ &= \sum_{(u,v) \in DAG} \max\{0, l(u) - l(v) + 1\} + \\ &\quad \sum_{(u,v) \in \text{EG}} \max\{0, l(u) - l(v) + 1\} \\ &= 0 + \sum_{(u,v) \in \text{EG}} l(u) - l(v) + 1 \qquad \text{(By Lemmas 4, 5)} \\ &= \sum_{C \in \mathcal{C}} \sum_{(u,v) \in C} l(u) - l(v) + 1 \\ &\quad \text{(where } \mathcal{C} \text{ is some cycle cover of the Eulerian subgraph)} \\ &= \sum_{C \in \mathcal{C}} \sum_{(u,v) \in C} l(u) - l(v) + 1 \\ &= \sum_{C \in \mathcal{C}} |C| \left( \text{For any cycle } C, \sum_{(u,v) \in C} l(v) - l(u) + 1 = |C| \right) \\ &= \text{number of edges in the Eulerian subgraph} \\ &= \text{Value of the dual solution} \end{aligned}$$

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

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