

The Layered Structure of Company Share Networks

Andrea Romei
Dipartimento di Informatica
Università di Pisa, Italy
andrearomei74@gmail.com

Salvatore Ruggieri
Dipartimento di Informatica
Università di Pisa, Italy
ruggieri@di.unipi.it

Franco Turini
Dipartimento di Informatica
Università di Pisa, Italy
turini@di.unipi.it

Abstract—We present a framework for the analysis of corporate governance problems using network science and graph algorithms on ownership networks. In such networks, nodes model companies/shareholders and edges model shares owned. Inspired by the widespread pyramidal organization of corporate groups of companies, we model ownership networks as layered graphs, and exploit the layered structure to design feasible and efficient solutions to three key problems of corporate governance. The first one is the long-standing problem of computing direct and indirect ownership (*integrated ownership problem*). The other two problems are introduced here: computing direct and indirect dividends (*dividend problem*), and computing the group of companies controlled by a parent shareholder (*corporate group problem*). We conduct an extensive empirical analysis of the Italian ownership network, which, with its 3.9M nodes, is $30\times$ the largest network studied so far.

I. INTRODUCTION

Corporate finance is the study of the financing patterns of companies. There are several agents in the market, with different objectives. On the one side, shareholders invest their funds in a company to generate returns. They collectively own the company and receive annual dividends in proportion to the shares owned. On the other side, business decisions are made by the board of directors of the company. The board is nominated at shareholder meetings with a voting mechanism typically following the *one-share-one-vote* rule: voting rights are proportional to ownership rights (i.e., to shares owned). Thus, a shareholder owning 51% of shares controls the business decisions. The larger the shares owned by the controller the more sane is the governance: the controller has a clear interest in that the company generates returns. This guarantees returns for minority shareholders as well. However, there exist schemes, such as pyramidal chains of companies [1], which lead to controlling *de-facto* a company through minority shareholdings. This may have negative impacts because the controller has incentives to divert resources from the company instead of generating returns for all shareholders [15]. As an extreme example, a controller which owns only 1% of shares of a company, but is able to nominate the board of directors may affect strategic decisions of the business (partnerships, supply/sell prices, etc.) in favor of other companies owned 100% by the controller, with the effect that the company will have less profits and then lower dividends. The controller will lose 1% of the missing returns, but will gain 100% of them through the other companies owned. This diversion of resources is at the expenses of the other shareholders, who will lose 99% of the missing returns. A key element of corporate governance is then to effectively understand and manage the separation between ownership, returns and control [12].

The analysis of financing patterns can be conducted using tools from network science [8]. An ownership network is a graph where nodes model shareholders and companies, and an edge between a shareholder and a company is weighed by the shares of the company owned by the shareholder. We extend the state-of-the-art of advanced data analysis on ownership networks both in methods and in empirical investigations. Regarding methods, we recognize a layered structure of the ownership network, which follows from the pyramidal organization of groups of companies. On this ground, we design fast algorithms able to uncover sub-structures of control and ownership. We tackle the long-standing *integrated ownership problem* (how much shares of a company a shareholder owns either directly or indirectly), a new model of determining the group of companies controlled by a same shareholder (the *corporate group problem*), and a new problem that we call the *dividend problem* (how much yearly dividends of a company a shareholder receives either directly or indirectly). Solving these problems is of primary importance for economists, market control authorities, and policy makers, who are also interested in (suspicious) outlier cases such as:

integrated ownership vs. dividends: which shareholders receive from companies lower dividends than their integrated ownership?

integrated ownership vs. control: which companies are controlled by a shareholder with a minority integrated ownership?

Regarding empirical investigation, we report on the analysis of a large and complete dataset: the network of *all* Italian companies, shareholders and shareholdings (1.5M companies, 3.9M nodes, 3.87M edges). Existing empirical studies focused, in fact, on small and incomplete datasets. Datasets were small because they were restricted to corporations listed in stock markets. A fortiori, datasets were also incomplete. Since not all companies/shareholders were in those datasets, the chain of shareholders of a company was broken at a certain point. The empirical investigation confirms a layered structure for the Italian ownership network, with a small difference between integrated ownership and dividends, and with a low percentage of companies controlled by minority shareholders. Summing up, the interplay among methods, algorithms and empirical study of a significant case is the main contribution of the paper.

The paper is organized as follows. Section II introduces ownership graphs and groups of companies. Sections III and IV recall existing problems and introduce new ones, providing flat solutions. Section V formalizes the notion of layers and exploits layers for optimizing flat solutions. Section VI provides an extensive data analysis of the network of Italian companies. Section VII summarizes related work and concludes.

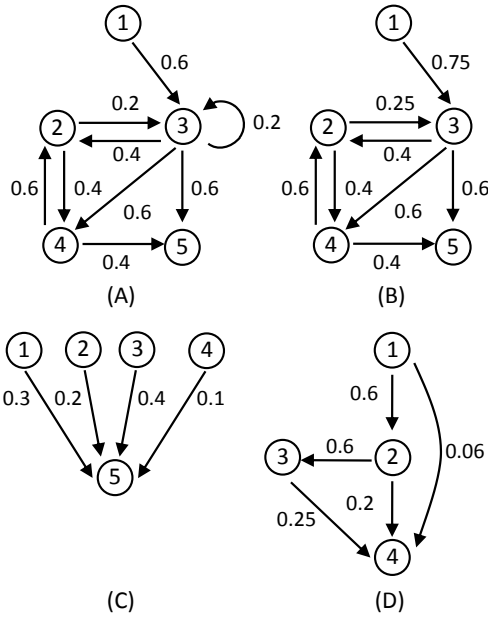


Fig. 1. Ownership graphs: (A) non-canonical; (B) canonical; (C) partnership; (D) pyramidal.

II. OWNERSHIP GRAPHS AND GROUPS

Companies are owned by shareholders through stocks or other legal terms of ownership of the company capital. The percentage of capital of a company owned by a shareholder is the *share* owned. Shareholders can be individuals, other companies, or public institutions.

A. Ownership graphs

Let $\mathcal{N} = \{1, \dots, N\}$ be the set of IDs of shareholders and companies. An *ownership graph* is a weighted directed graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ where a directed edge $e = (i, j, w)$ is in $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \times (0, 1]$ iff shareholder i owns $w > 0$ shares of company j . We write $e_{ij} = 1$ if there exists $(i, j, w) \in \mathcal{E}$, and $e_{ij} = 0$ otherwise. The square matrix E such that $E_{ij} = e_{ij}$ for $i, j \in \mathcal{N}$ is the *adjacency matrix* of \mathcal{G} . Also, we write $w_{ij} = w$ if $(i, j, w) \in \mathcal{E}$, and $w_{ij} = 0$ otherwise. The square matrix W such that $W_{ij} = w_{ij}$ is the *weighted adjacency matrix* of \mathcal{G} . The set of incoming neighbors of node j is $N_{in}(j) = \{i \in \mathcal{N} \mid e_{ij} = 1\}$. Analogously, $N_{out}(i) = \{j \in \mathcal{N} \mid e_{ij} = 1\}$ is the set of outgoing neighbors of i . For any node $j \in \mathcal{N}$, it turns out $S = \sum_{i \in N_{in}(j)} w_{ij} \leq 1$. In particular, natural persons and public institutions are not owned by anybody, hence $S = 0$. Moreover, if the ownership graph regards a restricted set of companies (e.g., for a specific country or industry sector), shareholders that are not in the set are not part of the graph, hence $S < 1$ for the companies in the set which are owned by those shareholders. Finally, missing data on shareholdings may also lead to $S < 1$. Nevertheless, ownership graphs are among the most accurate datasets for data science analysis, since data has legal and business value. A *path* $p = \langle i_1, \dots, i_k \rangle$ is a non-empty sequence of nodes such that there is an edge $(i_h, i_{h+1}, w_{i_h i_{h+1}}) \in \mathcal{E}$ for every $h \in [1, k-1]$. The path is *simple* if i_1, \dots, i_k are distinct nodes. The *weight* of the path is $weight(p) = \prod_{h=1}^{k-1} w_{i_h i_{h+1}}$.

B. Canonical ownership graphs

Treasury shares are shares owned by the issuing company. They originate either from stocks withheld by the company or from buybacks from the market. Treasury shares do not pay a dividend, have no voting rights, and cannot exceed a maximum percentage of total capitalization. For instance, if a company owns 20% of its own shares, the actual value and decision power of a shareholder owning 60% of the shares is $60\%/80\% = 75\%$. In an ownership graph, edges from a node to itself (*self-links*) model treasury shares. We exploit the above argument to remove those links. Let $w_{jj} > 0$ be the weight of a self-link for a node $j \in \mathcal{N}$. We re-weight edges $e = (i, j, w)$ for $i \neq j$ as $e' = (i, j, w')$ where $w' = w/(1 - w_{jj})$. We call the ownership graph obtained by removing self-links $(j, j, w_{jj}) \in \mathcal{E}$ and by re-weighting the other incoming edges a *canonical ownership graph*. Fig. 1 (A) shows an example of an ownership graph with a self-link on node 3. Fig. 1 (B) is its canonical form.

C. Groups

Groups of companies and their shareholders in an ownership network are typically organized in chains of ownership, e.g., as in corporate groups. The simplest organization is a partnership, where a company is owned by shareholders that own nothing else – see e.g., Fig. 1 (C). When all shareholders are natural persons associated in doing business, this boils down to the legal form of partnership businesses. Groups of companies can be structured in more complex forms than partnerships. This raises a number of challenging problems for corporate governance, some of which are introduced in the next section.

III. PROBLEM STATEMENTS

From a corporate governance perspective, it is important to know: (*integrated ownership problem*) how much shares of a company are owned by a shareholder either directly or indirectly through other companies; (*dividend problem*) how much yearly dividends of a company are received by a shareholder either directly or indirectly through other companies; (*corporate group problem*) what are the groups of companies controlled by a common parent shareholders.

A. The integrated ownership problem

The total amount of shares o_{ij} of a company j owned by a shareholder i , either directly or indirectly, is essential to determine the value of j in the portfolio of i . As an example, node 3 in Fig. 1 (B) directly owns 60% of shares of node 5 and indirectly owns some additional shares, e.g., $0.6 \cdot 0.4 = 24\%$ through path $\langle 3, 4, 5 \rangle$. o_{ij} is called the *integrated ownership* of j by i , and the matrix O with elements $O_{ij} = o_{ij}$ is called the *integrated ownership matrix* ([3], [8]). Formally, o_{ij} is defined recursively as the value for which:

$$o_{ij} = w_{ij} + \sum_{k \neq i} o_{ik} w_{kj}$$

namely, o_{ij} is the sum of direct ownership (w_{ij}) and indirect ownership through a node $k \neq i$ ($o_{ik} w_{kj}$). Indirect ownership through i itself is not considered, since this amounts at cycles of self-ownership. Cycles through other companies are instead possible because national legislations admit *cross ownership*¹

¹Contrarily to treasury shares, cross-owned shares pay a dividend and have voting rights.

between companies. Consider again Fig. 1 (B). Paths from node 3 to node 5 that pass twice or more from node 3 do not count, since they involve self-owned shares, e.g., as in $\langle 3, 2, 3, 5 \rangle$. The number of paths that count is, however, infinite, because the loop between nodes 2 and 4 *can* be travelled, e.g., in paths $\langle 3, (2, 4)^n, 5 \rangle$ and $\langle 3, 4, (2, 4)^n, 5 \rangle$ for $n \geq 1$. Using matrix notation, the integrated ownership matrix O is the solution of the recurrence relation:

$$O = (I - \text{diag}(O))W + OW \quad (1)$$

Starting from a canonical ownership graph, the *integrated ownership problem* consists of computing, if it exists, the integrated ownership matrix O .

B. The dividend problem

We introduce here a new problem regarding the calculation of dividends. Dividends are distributed yearly to shareholders in proportion to the shares they own. In turn, shareholders that are companies distribute their dividends to their own shareholders, and so on. The amount of dividends of a company j that reaches some shareholder i in the chain, however, is not necessarily proportional to the integrated ownership o_{ij} . The point here is that loops in the dividend payout are broken by temporal considerations, hence paths cannot contain any cycle. Consider again Fig. 1 (B). Node 3 receives dividends of node 5 directly (a quota of 0.6) and indirectly through paths $\langle 3, 4, 5 \rangle$ (a quota of 0.24) and $\langle 3, 2, 4, 5 \rangle$ (a quota of 0.064), for a total of 0.904. Consider paths with loops, e.g., $p = \langle 3, 4, 2, 4, 5 \rangle$. Such path provides $\text{weight}(p) = 0.0384 = 3.84\%$ of the dividends of 5 to node 3, but it requires two fiscal years to be traversed. In fact, dividends of 5 that reach 4 are partly distributed to node 2. Part of such dividends is returned back to 4 (and then to 3) only in the next fiscal year. However, for a shareholder i it is useful to know the quota d_{ij} of dividends of j that may reach it in *one* fiscal year. Let δ_j be the monetary amount of dividends of company j distributed to shareholders, and ν_j be the company value, e.g., its stock market value. The *dividend-price ratio* ν_j/δ_j is widely adopted by economic analysts as a measure of the earnings on direct investment of a shareholder. The extended ratio $(o_{ij} \nu_j)/(d_{ij} \delta_j)$ of integrated value $(o_{ij} \nu_j)$ over dividend pro-quota $(d_{ij} \delta_j)$ is the generalization of the dividend-price ratio to both direct and indirect investments.

We call the matrix D with elements $D_{ij} = d_{ij}$ the *dividend matrix*. Formally,

$$d_{ij} = \sum_{p \in \{\langle i, \dots, j \rangle \mid \langle i, \dots, j \rangle \text{ simple path}\}} \text{weight}(p) \quad (2)$$

i.e., d_{ij} is the sum of weights of all simple paths from i to j .

Summarizing, starting from a canonical ownership graph, the *dividend ownership problem* consists of computing the dividend matrix D . Such a matrix always exists since the number of simple paths is bounded by $N!$. Moreover, for an acyclic ownership graph, it is readily checked that $D = O$.

C. The corporate group problem

In the control problem, one is interested in the capacity of a shareholder of affecting directly or indirectly the decisions of a company. Assuming that voting rights are proportional to ownership rights (the so-called *one-share-one-vote* rule), a linear threshold model estimates that a shareholder i *directly*

control a company j if $w_{ij} > 0.5$. i is called the controller (or the controlling shareholder), and j the controlled company. The controller has the majority of votes in any decision of the shareholder meeting of the controlled company. Hence, it controls the company. Various models have been proposed for the notion of indirect control (weakest link [7], integrated control [6], relative majority [8]). We introduce here a new 0/1 model, where i *indirectly controls* j if it controls, either directly or indirectly, other companies and altogether they own more than 50% of the company shares of j . The rationale is that, in any shareholder meeting of company j , the votes of i and of the intermediate companies controlled by i converge to the decision wanted by i . The proposed model of propagation of control is an instance of the Linear Threshold Model (LTM) propagation [9] over the ownership network. Starting from such a model, we introduce the problem of discovering corporate groups. Consider the ownership graph in Fig. 1 (D). Node 1 directly controls 2, which, in turn, directly controls node 3. Although node 4 is not directly controlled, the sum of the shares owned by 1, 2 and 3 is 0.51, thus it is indirectly controlled by node 1 (but not by node 2 nor 3). The integrated ownership of node 4 by 1 is $o_{14} = 0.06 + 0.6 \cdot 0.2 + 0.6 \cdot 0.6 \cdot 0.25 = 0.27$. Thus, indirect control can be achieved even without an integrated ownership of the majority of shares. This is an example of a pyramidal chain of companies (or “Chinese boxes”) used to keep control of a (large) group through minority shareholdings.

A *corporate group* is a group of companies controlled by a same controller, called the *parent shareholder*. Corporate groups must be explicitly declared only for specific cases stated by the law, e.g., for listed companies. Starting from a canonical shares graph, the *corporate group problem* consists of computing all pairs (i, \mathcal{G}_i) of parent shareholders $i \in \mathcal{N}$ and corporate group \mathcal{G}_i of companies controlled by i .

IV. FLAT SOLUTIONS

We will now provide solutions to the three introduced problem that do not take advantage of any structure of share graphs. To contrast the approach with the one exploiting a layered structure, we call the solutions introduced here “flat”.

A. The integrated ownership problem

A solution of the recurrence equation (1) was first provided by [3] (see also the presentation in [8]) as:

$$O = \text{diag}(V)^{-1}(V - I) \quad \text{where } V = (I - W)^{-1}$$

The matrix $V - I$ can be better understood when stated by exploiting the Neumann series as:

$$V - I = (I - W)^{-1} - I = \left(\sum_{n \geq 0} W^n \right) - I = \sum_{n \geq 1} W^n$$

i.e., $V - I$ is the transitive closure of the weighted adjacency matrix W . Thus, V_{ij} , for $i \neq j$, is the sum of the weights of all paths from i to j , and V_{ii} is the sum of the weights of all paths from i to i plus 1. The matrix $I - W$ is invertible under a condition generally satisfied by ownership graphs, namely that there exist no subset of nodes $\mathcal{S} \subseteq \mathcal{N}$ totally owned by

Algorithm 1 *ShareO()*

```

// transitive closure of W through matrix inversion
1:  $V \leftarrow (I - W)^{-1}$ 
// integrated ownership matrix
2: for  $i \in [1, N]$  do
3:    $\mathcal{J} \leftarrow \{j \in [1, N] \mid V_{ij} > 0\}$ 
4:   for  $j \in \mathcal{J}$  do
5:     if  $i \neq j$  then
6:        $O_{ij} \leftarrow V_{ij}/V_{ii}$ 
7:     else
8:        $O_{ii} \leftarrow (V_{ii} - 1)/V_{ii}$ 

```

themselves, i.e., such that for every $j \in \mathcal{S}$, $\sum_{i \in \mathcal{S}} w_{ij} = 1$ (see [5]). The elements of O can then be stated as:

$$O_{ij} = \begin{cases} V_{ij}/V_{ii} & \text{if } i \neq j \\ (V_{ii} - 1)/V_{ii} & \text{otherwise.} \end{cases}$$

Let us justify this result. For $i \neq j$, the sum of weights of all paths from i to j can be split into the sum of those that do not cycle through i plus those that first cycle on i and then do not. Hence, $V_{ij} = O_{ij} + (V_{ii} - 1)O_{ij}$, which once solved, yields $O_{ij} = V_{ij}/V_{ii}$. For $i = j$, the sum of all paths is $V_{ii} - 1 = O_{ii} + (V_{ii} - 1)O_{ii}$, which once solved, yields $O_{ii} = (V_{ii} - 1)/V_{ii}$ – this quantity is the amount of shares self-owned by i either directly (if the graph is not canonical²) or indirectly. *ShareO()* shown as Alg. 1 formalizes the computation of the non-zero cells of the integrated ownership matrix.

B. The dividend problem

Since cycles are not to be taken into account in computing indirect dividends, the dividend problem can be reduced to computing all simple paths between pairs of nodes in a graph. The all-pairs simple paths problem is solved by the Rubin’s algorithm [14] by computing a matrix R such that R_{ij} is the set of all simple paths from i to j . The dividend problem can then be solved by aggregating weights along all simple paths from i to j . According to equation (2):

$$D_{ij} = \sum_{p \in R_{ij}} \text{weight}(p)$$

Rubin’s algorithm is a smart generalization of the Floyd-Warshall algorithm for the all-pairs shortest path problem. We instantiate it in the procedure *ShareD()* shown as Alg. 2. R_{ij} actually stores pairs (p, w) where p is a simple path and $w = \text{weight}(p)$ is the weight of the simple path. The algorithm initializes R_{ij} with paths of length 1, namely the ownership graph edges (lines 1-3). Then it constructs simple paths from i to j , passing from an intermediate node k (lines 3-8). Given two simple paths (p_1, w_1) from i to k , and (p_2, w_2) from k to j , they can be appended to form a simple path $p_1 \cdot p_2$ from i to j when k is the only shared node, namely if $p_1 \cap p_2 = \{k\}$ (line 7). The weight of the new path is $w_1 \cdot w_2$ (line 8). The dividend matrix D_{ij} is computed at lines 9-12 by summing up all weights of simple paths in R_{ij} . For the sample graph of Fig. 1 (B), it turns out that:

$$O = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0.43 & 0.25 & 0.55 & 0.37 \\ 0 & 1 & 0.25 & 1 & 1 \\ 0 & 0.67 & 0.17 & 0.37 & 0.50 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} D = \begin{pmatrix} 0 & 0.57 & 0.75 & 0.57 & 0.68 \\ 0 & 0 & 0.25 & 0.55 & 0.37 \\ 0 & 0.76 & 0 & 0.76 & 0.90 \\ 0 & 0.60 & 0.15 & 0 & 0.49 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

²Strictly speaking, the solution does not require the ownership graph to be canonical. However, canonization will be useful later on when considering graph layers.

Algorithm 2 *ShareD()*

```

// initialization
1: for  $i, j \in [1, N]$ ,  $w_{ij} > 0$  do
2:    $R_{ij} \leftarrow \{(i, j), w_{ij}\}$ 
// Rubin’s algorithm
3: for  $k \in [1, N]$  do
4:    $\mathcal{I} \leftarrow \{i \in [1, N] \mid R_{ik} \neq \emptyset\}$ 
5:    $\mathcal{J} \leftarrow \{j \in [1, N] \mid R_{kj} \neq \emptyset\}$ 
6:   for  $i \in \mathcal{I}$ ,  $j \in \mathcal{J}$  do
7:     for  $(p_1, w_1) \in R_{ik}$ ,  $(p_2, w_2) \in R_{kj}$ ,  $p_1 \cap p_2 = \{k\}$  do
8:        $R_{ij} \leftarrow R_{ij} \cup \{(p_1 \cdot p_2, w_1 \cdot w_2)\}$ 
// dividend matrix
9: for  $i \in [1, N]$  do
10:   $\mathcal{J} \leftarrow \{j \in [1, N] \mid R_{ij} \neq \emptyset\}$ 
11:  for  $j \in \mathcal{J}$  do
12:     $D_{ij} \leftarrow \sum_{(p, w) \in R_{ij}} w$ 

```

Company 3 owns directly or indirectly all shares of companies 2, 4 and 5, and 25% of its own shares. This makes the 75% of shares owned by shareholder 1 to actually weigh as 100% of integrated ownership, i.e., integrated ownership ends up with re-weighting edges (as we did in canonization) to take into account cross-ownership. Notice that, due to cycles, dividends are not necessarily proportional to integrated ownership.

C. The corporate group problem

The solution procedure *ShareG()* shown as Alg. 3 adopts a pattern similar to the previous ones. It first computes a control matrix C such that $C_{ij} = 1$ if i controls j (lines 1–4). Parent shareholders i are those not controlled by anyone³ (i.e., such that for all k , $C_{ki} = 0$), and the corporate group \mathcal{G}_i consists of the column indexes j such that $C_{ij} = 1$ (lines 5–6). According to our model of indirect control, the set of companies controlled by i is obtained through an invocation *Control(i)* to the LTM-like algorithm [9] of propagation of control shown in Alg. 4. In such algorithm, \mathcal{C} is the set of controlled nodes (“activated” in the terminology of LTM) and \mathcal{S} is the set of candidate nodes, corresponding to the frontier of a visit of the graph. If the sum of incoming edges from nodes in \mathcal{C} is greater than 0.5 then a candidate node j is added to the control set (lines 5), and its outgoing edges not yet in the control set are added to the candidate set (line 6). The visit of the graph continues while there are candidate nodes (line 3).

D. Computational complexities

The procedure *ShareD()* has the same complexity of matrix inversion, which is $O(N^3)$ using a naïve approach based on Gauss-Jordan elimination.

The procedure *ShareD()* requires up to $O(N^3)$ operations. An operation (lines 7–8) consists of $O(P)$ intersections of simple paths, where P is the total number of simple paths between any pair of nodes. Each intersection is $O(N)$, since a simple path has length at most N . Summarizing, *ShareD()* is $O(N^4 P)$. The number P of simple paths can be exponential in N , e.g., for a clique of N nodes it is $N!$. This makes *ShareD()* exponential in N in the worst case.

³We assume there exists no pair of companies that control each other.

Algorithm 3 *ShareG()*

```
// control matrix
1: for  $i \in [1, N]$  do
2:    $C \leftarrow \text{Control}(i)$ 
3:   for  $j \in C$  do
4:      $C_{ij} \leftarrow 1$ 
// corporate groups
5: for  $i \in [1, N]$  such that  $C_{ki} = 0$  for all  $k \in [1, N]$  do
6:    $G_i \leftarrow \{j \in [1, N] \mid C_{ij} = 1\}$ 
```

Algorithm 4 *Control(i)*

```
1:  $C \leftarrow \{i\}$ 
2:  $S \leftarrow N_{out}(i)$ 
3: while  $S \neq \emptyset$  do
4:    $j \leftarrow \text{pop } S$ 
5:   if  $\sum_{k \in N_{in}(j) \cap C} w_{kj} > 0.5$  then
6:      $C \leftarrow C \cup \{j\}$ 
7:      $S \leftarrow S \cup N_{out}(j) \setminus C$ 
8: return  $C \setminus \{i\}$ 
```

The procedure *ShareG()* is $O(N^4)$. In fact, there are N LTM visits. A call *Control(i)* amounts to a loop of $O(N)$ steps. Each step performs the summation at line 5, which takes $O(N)$, for $O(N)$ nodes in the set S .

The worst case complexities of the three procedures would prevent us from using them for generic graphs. However, on the one side the worst cases regard dense graphs, whilst ownership networks are very sparse. Memory occupation can take advantage of such sparsity by using sparse data structures. On the other side, we will discuss below how to exploit the layered structure of ownership graphs to further restrict the usage of the flat algorithms to a tiny sub-graph of the ownership graph.

V. LAYERS IN OWNERSHIP GRAPHS

We present a structural model of ownership graphs based on layers, and exploit it to revise the algorithms of the previous section. The model builds on a pervasive pattern observed in corporate groups of companies, namely pyramidal organization [1]. At top level, there is a parent shareholder owning a some companies in the group, which in turn own other companies at deeper layers and so on.

A. Layers

Consider Fig. 2 (left), where nodes are assigned to layers. Layer $k = 1$ consists of nodes with no incoming edge, such as node 1. These nodes are either individuals (owned by nobody), public institutions (owned solely by the State), or companies whose shareholders are not part of the data (e.g., companies established abroad). At layers $k \leq 2$, there are nodes having all their incoming edges from nodes at layers $< k$, such as node 2 in the example. Then there is a layer $\kappa = 3$ which includes nodes that may be involved in loops within the layer, as the one between 3 and 4. Finally, layer $k = 4$ again satisfies the property that incoming edges are from layers $< k$. Let us formalize the definition of layers.

A partition of the nodes \mathcal{N} into K sets $\mathcal{L}_1, \dots, \mathcal{L}_K$, called *layers*, is a layered partition if, for $k \in [1, K]$, all nodes

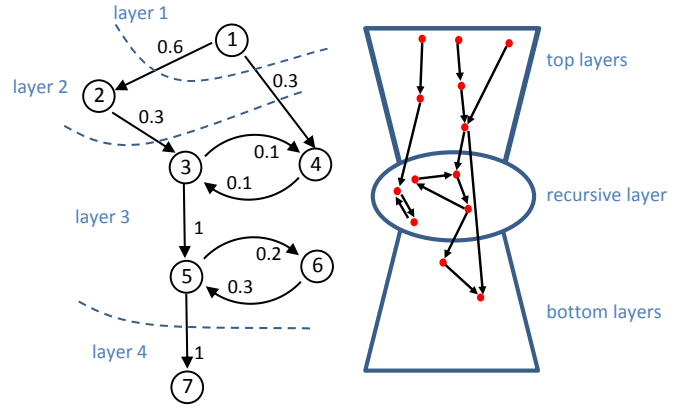


Fig. 2. Layers in ownership graphs.

in \mathcal{L}_k have all incoming edges from nodes in lower layers $\mathcal{L}_1, \dots, \mathcal{L}_{k-1}$ except for at most one $\kappa \in [1, K]$ for which all nodes in \mathcal{L}_κ have all incoming edges from nodes in layers $\mathcal{L}_1, \dots, \mathcal{L}_\kappa$. Layer κ , if it exists, is called the *recursive layer*.

The definition is closely related to bow-tie structures, already observed in ownership graphs [8]. See Fig. 2 (right). Layers $1, \dots, \kappa - 1$ (top layers) resemble the IN component of the bow-tie, layer κ is the analogous of the strongly connected component (SCC), and layers $\kappa + 1, \dots, K$ (bottom layers) resemble the OUT component. Notice that \mathcal{L}_κ is not necessarily strongly connected, as e.g., in Fig. 2 (left).

The procedure *ExtractLayers()*, shown in Alg. 5, computes a layered partition through two topological sort algorithms. The first one (lines 1–14) starts from nodes that have no incoming edge, i.e., layer \mathcal{L}_1 . For every node in the layer, it marks as deleted the outgoing edge towards node j (line 11), and adds j to the next layer if it has no unmarked incoming edges. The set \mathcal{V} contains the nodes not yet assigned to layers. At the end of the first topological sort, \mathcal{V} contains the recursive layer and the bottom layers. The second topological sort (lines 16–28) visits such nodes bottom-up starting from those that have no outgoing edges (layer \mathcal{L}_K). At the end of the second topological sort, the unvisited nodes are those of the recursive layer (line 29). The procedure has linear complexity since it visits nodes and edges only once. Moreover, it returns a layered partition with the *smallest* recursive layer.

B. Exploiting layers

The recursive layer is the hard-to-analyse sub-graph of the ownership graph. Later on we will see that, for a real large graph, it is only a small fraction of the overall graph. This makes it worth devising solutions to the problems of this paper that take advantage of the layered structure of ownership graphs. Consider first the dividend problem. The key observation is that D_{ij} for a node i at layer $k \neq \kappa$ can be computed from cells D_{hj} where h, j are at layer $> k$. In fact, for j in a layer $\leq k$, we have that $D_{ij} = 0$ since there is no path from i to j . For j in a layer $> k$, we have that:

$$D_{ij} = w_{ij} + \sum_{h \in N_{out}(i)} w_{ih} \cdot D_{hj} \quad (3)$$

where $h \in N_{out}(i)$ is in a layer $> k$. The equation above can be intuitively read as follows: i owns w_{ij} dividends of j

Algorithm 5 *ExtractLayers()*

```

1:  $\mathcal{V} = \{1 \dots N\}$ 
2: for  $i \in \mathcal{V}$  do
3:    $e[i] = |N_{in}(i)|$ 
4:  $\mathcal{L}_1 \leftarrow \{i \in [1, N] \mid e[i] = 0\}$ 
5:  $k \leftarrow 1$ 
6: while  $\mathcal{L}_k \neq \emptyset$  do
7:    $\mathcal{V} = \mathcal{V} \setminus \mathcal{L}_k$ 
8:    $\mathcal{L}_{k+1} \leftarrow \emptyset$ 
9:   for  $i \in \mathcal{L}_k$  do
10:    for  $j \in N_{out}(i)$  do
11:       $e[j] \leftarrow e[j] - 1$ 
12:      if  $e[j] = 0$  then
13:         $\mathcal{L}_{k+1} \leftarrow \mathcal{L}_{k+1} \cup \{j\}$ 
14:    $k \leftarrow k + 1$ 
15:  $\kappa \leftarrow k$ 
16: for  $i \in \mathcal{V}$  do
17:    $e[i] = |N_{out}(i)|$ 
18:  $\mathcal{M}_1 \leftarrow \{i \in \mathcal{V} \mid e[i] = 0\}$ 
19:  $h \leftarrow 1$ 
20: while  $\mathcal{M}_h \neq \emptyset$  do
21:    $\mathcal{V} = \mathcal{V} \setminus \mathcal{M}_h$ 
22:    $\mathcal{M}_{h+1} \leftarrow \emptyset$ 
23:   for  $j \in \mathcal{M}_h$  do
24:     for  $i \in N_{in}(j)$  do
25:        $e[i] \leftarrow e[i] - 1$ 
26:       if  $e[i] = 0$  then
27:          $\mathcal{M}_{h+1} \leftarrow \mathcal{M}_{h+1} \cup \{i\}$ 
28:    $h \leftarrow h + 1$ 
29:  $\mathcal{L}_\kappa \leftarrow \mathcal{V}$  // recursive layer
30:  $K \leftarrow \kappa + h - 1$ 
31: return  $\mathcal{L}_1, \dots, \mathcal{L}_\kappa, \mathcal{M}_{h-1}, \dots, \mathcal{M}_1$ 

```

directly, and $w_{ih} \cdot D_{hj}$ indirectly through the directly owned company h . Consider now layer κ . First, we have to run the flat procedure *ShareD()* on \mathcal{L}_κ to compute dividends, since cross-ownerships (a.k.a., loops) are present. The resulting matrix D does not include values for cells D_{ij} where $i \in \mathcal{L}_\kappa$ and j is in $\mathcal{L} = \mathcal{L}_{\kappa+1} \cup \dots \cup \mathcal{L}_K$. We first compute the intermediate values:

$$D'_{ij} = w_{ij} + \sum_{h \in N_{out}(i) \cap \mathcal{L}} w_{ih} \cdot D_{hj}$$

namely the dividends as if κ were non-recursive. Then, we calculate for i in layer κ and j in layer $> \kappa$:

$$D_{ij} = D'_{ij} + \sum_{h \in \mathcal{L}_\kappa} D_{ih} \cdot D'_{hj} \quad (4)$$

Intuitively, dividends of j can reach i either through simple paths connecting i to layers $> \kappa$ (D'_{ij}) or through simple paths that first transverse the recursive layer up to a node h (D_{ih}) and then pass to layers $> \kappa$ (D'_{hj}). Summarizing, the layered procedure for solving the dividend problem consists of three steps: (*step 1*) compute layers through *ExtractLayers()*; (*step 2*) run the procedure *ShareD()* for layer κ to compute the dividend matrix of nodes in such a layer; (*step 3*) compute bottom-up the rows of the matrix for nodes in layers $K, \dots, \kappa + 1, \kappa, \kappa - 1, \dots, 1$ by using equations (3–4). For lack of space, we do not report the pseudo-code of the layered algorithm. The worst case complexity is the same of the flat version – this occurs when the recursive layer is the whole graph. As we will see later on, however, the recursive layer is orders of magnitude smaller than the whole ownership graph. Since for non-recursive layers the layered approach has time complexity linear in the number of nodes and edges, and space complexity linear in the number of edges (matrix R from *ShareD()* is computed *only* for the recursive layer), this makes the layered approach efficient for sparse graphs.

Similar arguments can be adopted for integrated ownership and *ShareO()*, where the analogous of the equations (3,4) allow to speed up the computation of the intermediate matrix $V = (I - W)^{-1}$. The layered approach results, in practice, in a (double topological) re-ordering of rows of $I - W$ so that it is upper triangular for all rows except for the nodes in the

Algorithm 6 *LayeredShareG()*

```

1:  $\mathcal{L}_1, \dots, \mathcal{L}_\kappa, \dots, \mathcal{L}_K \leftarrow \text{ExtractLayers}()$ 
2:  $\mathcal{V} \leftarrow \{1 \dots N\}$ 
3: for  $k \in [1, \kappa - 1]$  do
4:   for  $i \in \mathcal{L}_k \cap \mathcal{V}$  do
5:      $\mathcal{G}_i \leftarrow \text{Control}(i)$ 
6:      $\mathcal{V} \leftarrow \mathcal{V} \setminus \mathcal{G}_i$ 
7:   for  $i \in \mathcal{L}_\kappa \cap \mathcal{V}$  do
8:      $\mathcal{C} \leftarrow \text{Control}(i)$ 
9:     for  $j \in \mathcal{C}$  do
10:       $C_{ij} \leftarrow 1$ 
11:   for  $i \in \mathcal{L}_\kappa \cap \mathcal{V}$  such that  $C_{ki} = 0$  for all
       $k \in \mathcal{L}_\kappa \cap \mathcal{V}$  do
12:      $\mathcal{G}_i \leftarrow \{j \in [1, N] \mid C_{ij} = 1\}$ 
13:      $\mathcal{V} \leftarrow \mathcal{V} \setminus \mathcal{G}_i$ 
14:   for  $k \in [\kappa + 1, K]$  do
15:     for  $i \in \mathcal{L}_k \cap \mathcal{V}$  do
16:        $\mathcal{G}_i \leftarrow \text{Control}(i)$ 
17:        $\mathcal{V} \leftarrow \mathcal{V} \setminus \mathcal{G}_i$ 

```

recursive layer. However, the size of this matrix is the same as the output O , hence there is no substantial gain in memory occupation, as for the dividend matrix.

An analogous approach for the *ShareG()* procedure would adopt a bottom-up layered computation of matrix C . We provide instead a top-down solution as the *LayeredShareG()* procedure shown in Alg. 6. It proceeds top-down from layer 1 to K . At each layer $< \kappa$ (lines 1–6), the procedure computes the group controlled by a shareholder i in the layer and it removes the companies of the group from the set \mathcal{V} of nodes to be visited. Thus, shareholder i is necessarily a parent shareholder (otherwise, it is controlled by a company in some previous layer, hence it is not in \mathcal{V}), and the group \mathcal{G}_i is the corporate group of i . Similarly for layers $> \kappa$ (lines 14–17). For layer κ , the approach of *ShareG()* must be necessarily adopted (lines 7–13).

VI. EMPIRICAL ANALYSIS

In this section, we analyse a real ownership graph that is $30\times$ the largest network studied so far in the literature [8]. Our goal is twofold. The first one is a network science goal: studying the characteristics of the network in terms of degree distributions and layered structure. The second one is a data-driven corporate governance goal: investigating the cases of critical conditions of the market where integrated ownerships differ from dividends, and where control is achieved without owning the majority of integrated ownership. Finally, we provide running times of the layered and flat version of algorithms, showing how the former ones run very fast.

A. The Italian ownership graph

The Italian Business Register records information on all Italian companies. Data include legal and financial information, and it is kept up-to-date by the companies themselves, since the register is recognized by law as the official source of information about companies. We had a unique access to a complete 2012 snapshot of the register. We computed the *Italian ownership graph* by considering all legal forms of

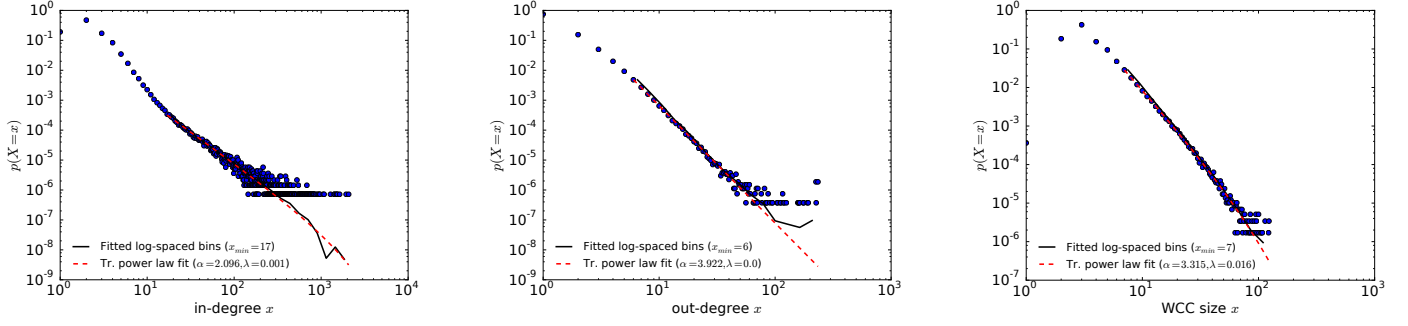


Fig. 3. Degree and WCC size distributions of the Italian canonical ownership graph.

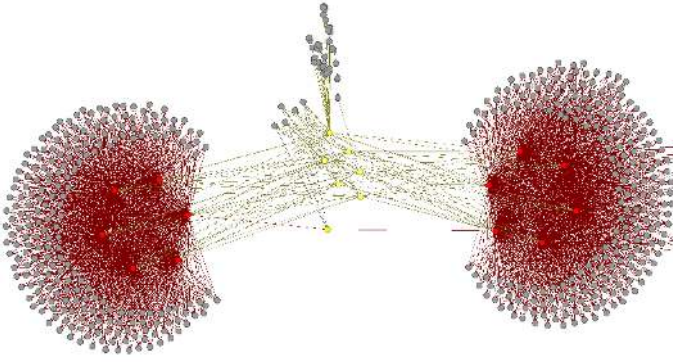


Fig. 4. The “lungs” subgraph.

companies (partnerships, stock companies, and limited companies) apart from sole proprietorship/individual businesses. All shareholders were included, apart from two cases. The first one concerns non-Italian companies owning shares of an Italian company: the non-Italian company is in the register as a shareholder, but its own shareholders are not included (because the company is registered abroad). The second case regards shareholders of floating stocks of companies listed in the Italian stock exchange: they are not managed by the Business Register.

The ownership graph. The ownership graph includes 3.904M nodes, of which 1.496M are companies, and 3.867M edges, of which only 2,768 (less than 0.08%) are self-links. The non-zero in-degree and out-degree distributions of the canonical ownership graph are shown in Fig. 3 (left, center). Data is fitted⁴ by truncated power laws (with parameter λ close to 0). The mean of non-zero in-degrees is 2.78 (less than 3 shareholders per company on average), and the mean of non-zero out-degree is 1.45 (less than one and a half company owned by a shareholder on average).

The company with the highest in-degree has 1524 shareholders. It is a Limited Liability Consortium supplying road haulage services in the business of transportation of goods for third parties. Shareholders of the consortium include truck owners and drivers. The second (resp. the third) topmost in-degree company is a service provider company with 1371 (resp. 1200) shareholders, which are small and medium firms

to which the company provides fiscal assistance services (resp. job training services).

There are 6 shareholders with the highest out-degree: each owns 234 other companies. There are also 6 other shareholders with the second highest out-degree of 225. These two cases are plotted as clear outliers in Fig. 3 (center). A subgraph of the ownership graph containing those 12 companies is shown in Fig. 4. We call it the “lungs” subgraph. Each of 6 highest out-degree companies owns shares of the same 234 companies in the left lung. Similarly, each of the other 6 companies owns shares of the same other 225 companies in the right lung. The 12 companies (nodes in red) are owned by the same 7 companies (nodes in yellow). Such 7 companies are non-Italian. This outlier group (7 foreign companies, 12 parent companies, >460 owned companies) definitely deserves further scrutiny by legal, taxation, and economy experts.

The non-partnership graph. The distribution of weakly connected components (WCCs) in the canonical ownership graph is shown in Fig. 3 (right). There are 586,568 WCCs. 456,521 are partnerships (refer to Sect. II-C). On average, a WCC includes 6.66 nodes, 2.55 of which are companies. The giant component (not shown in Fig. 3 (right)) includes 1.55M nodes, 632K of which are companies. We removed partnership WCCs from the canonical ownership graph, since they represent a common subgraph pattern, whose ownership, dividend and control matrices are trivial to compute and analyse. The resulting graph includes 2.474M nodes and 2.894M edges. We call it the *non-partnership graph*.

The layered structure. There are $K = 29$ layers in the non-partnership graph, with layer $\kappa = 16$ as the recursive layer. The layer sizes, shown in Fig. 5 (left), range from 1.538M nodes of layer 1 (individuals and foreign companies) to 3 nodes of layer 15 and to 1 node of layers 17 and 18. The recursive layer has 2,370 nodes, which is only 0.1% of the total. The size distribution of top layers (1-15) is fitted by an exponential distribution $e^{-\beta}$ with parameter $\beta = 0.84$. This can be interpreted as follows: *the probability of having a chain of company ownership decay exponentially with the length of the chain*. The size of bottom layers (17-29) is fitted by an exponential distribution with $\beta = -0.72$.

The heat map in Fig. 5 (center) shows how edges from a node at layer k are distributed among the layers $\geq k$. 79% of edges are directed towards the next layer, and 89.7% stay within the next two layers. The statement that the recursive layer is the hard-to-analyse sub-graph is supported by Fig. 5

⁴We use the methods and software from [2] for fitting heavily tailed distributions. The distribution with the best loglikelihood ratio is selected among power laws, truncated power laws, exponentials, stretched exponentials, and log-normals.

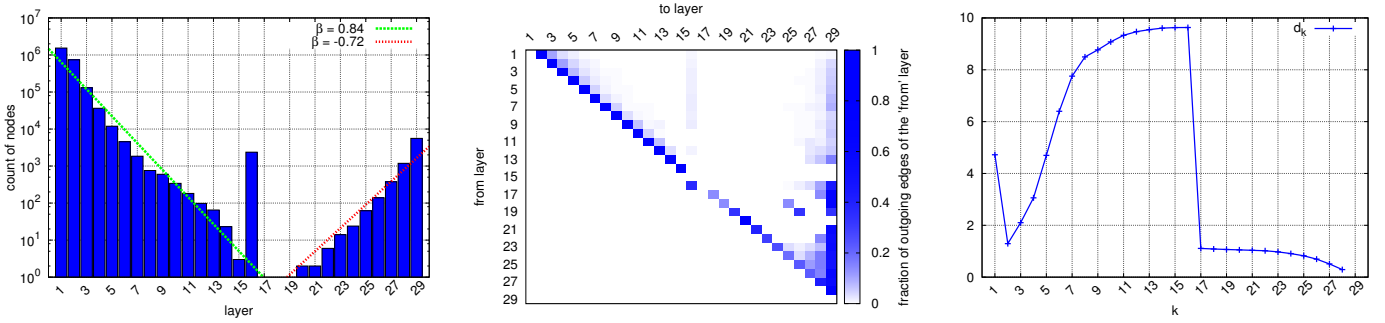


Fig. 5. Non-partnership graph stats. Left: layer sizes. Center: layer heat map. Right: average number of reachable nodes d_k from nodes in layers $\geq k$.

(right), which shows the average number d_k of nodes reachable from any node in the sub-network of layers $\geq k$. Reading the plot from right to left, d_k increases from layer $k = 29$ up to the recursive layer $k = 16$, then it decreases up to layer $k = 2$, and finally increases for layer $k = 1$. The recursive layer includes loops, which cause higher connectedness. Layers 2 to 15 include the ending points of chains of companies, thus reducing the average number of reachable nodes. Finally, nodes in layer 1 are starting points of chains, and they cannot be ending points, hence d_1 is greater than d_2 .

B. Ownership, dividends, and groups

The solutions of the integrated ownership, dividend, and corporate group problems allow for the analysis of the cases of interests for corporate governance discussed in the introduction. In addition, we sketch an approach for the discovery of family business groups.

Integrated ownership vs. dividends. Observe that $d_1 = 4.72$ in Fig. 5 (right): on average, a shareholder reaches/owns less than 5 companies either directly or indirectly. As a result, O and D are sparse matrices, with only 11.683M non-zero entries – less than five times the number of nodes. Let us consider the case when the difference $O_{ij} - D_{ij}$ is large. Fig. 6 (left) shows the distribution of $O_{ij} - D_{ij}$ (notice that such a difference cannot be negative). There are only $42 \cdot 10^3$ pairs (i, j) with non-zero difference, which is 0.36% of the non-zero entries of O , and only 100 pairs with a difference greater than 0.4. The network of Italian companies has, for most of it, a sane structure where shareholders receive dividends proportional to their ownership rights. Pairs with large difference should be subject to further scrutiny. An example sub-graph, shown in Fig. 7 (a), includes shareholder A and company D with a difference of 0.92. In fact, A receives dividends of D from the simple paths $\langle A, B, D \rangle$ and $\langle A, B, C, D \rangle$, for a total of $0.1 \cdot 0.6 + 0.1 \cdot 0.5 \cdot 0.4 = 0.08$. Due to cross-ownerships of B-C and C-D, the three companies B, C, D are totally owned by A, hence integrated ownership of D by A is 1. The difference is then 0.92. It is worth noting that, in theory, A does not control D, which is instead controlled by B which, in turn, is controlled by C. Nevertheless, the structure of the group clearly shows that, due to cross-ownership, A is the *de-facto* parent controller of the group. Our analysis was able to unveil this case. Whether hiding the actual controller is a wanted effect (and why) should be part of further investigation.

Integrated ownership vs. control. 1.96 M corporate groups were found in the non-partnership graph. Only 318,956 groups

are non-empty, i.e., $\mathcal{G}_i \neq \emptyset$. The other groups consists of shareholders not controlled by and not controlling anybody – they can be either individuals or companies that are widely held. The size of a corporate group is $|\mathcal{G}_i \cup \{i\}|$, i.e., the number of controlled companies plus one (the parent shareholder). The size distribution is shown in Fig. 6 (center). A power law fits the log-space binned distribution. The average size of a non-empty corporate group is 1.66, i.e., a parent shareholder that controls at least one company on average actually controls less than two companies. An example corporate group of size 115 is shown in Fig. 7 (b). The parent shareholder, shown in yellow, is a credit securitization company. It owns the majority of shares of 114 controlled companies. The structure of the corporation is thus flat (*horizontal* in the terminology of [1]): one controlling shareholder, many controlled companies. In a sense, the structure is a “reverse partnership” – the visualization in Fig. 7 (b) recalls a “sunburst”. Let us consider now the case of interest for corporate governance in which the parent shareholder controls a company with an integrated ownership lower than the majority of shares. Fig. 6 (right) plots the number of pairs (i, j) such that i is a parent shareholder, $j \in \mathcal{G}_i$ is controlled by i , and the integrated ownership of j by i is at most s , i.e., $O_{ij} \leq s$. There are only 10,590 pairs (equal to 1.98% of the total) such that O_{ij} is lower or equal than 50%. They appear in 6,228 distinct corporate groups. Again, we can conclude that the network of Italian companies has, for most of it, a sane structure regarding the natural requirement that the controller owns, either directly or indirectly, the majority of shares of the controlled company. The pairs for which this does not hold should be subject to further analysis by corporate governance experts. Our analysis can provide them with analytical tools for ranking pairs according to some criteria. For instance, one could look at corporate groups with high percentage of companies controlled via minority shares. The corporate group shown in red in Fig. 7 (c) is one such example. It is structured in 6 layers. The parent shareholder A (an individual) owns the majority of shares of the controlled company C (a financial real estate), but not the majority of shares of the other 7 controlled companies. For instance, A has an integrated ownership of only $0.8 \cdot 0.6 = 48\%$ shares of D, and $0.48 \cdot 0.29 + 0.8 \cdot 0.27 = 35.5\%$ shares of G.

Family business groups. Companies can be organized in different structures than corporate groups. For instance, our definition of corporate groups does not account for *family business groups*. Such groups are not controlled by a single parent shareholder, but rather by a small number of parent shareholders. This typically occurs when those shareholders

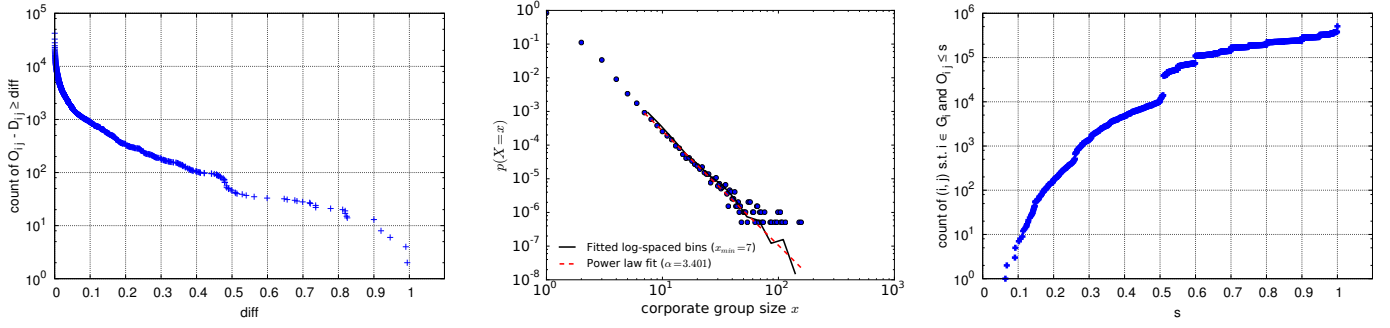


Fig. 6. Left: distribution of non-zero $O - D$. Center: distribution of corporate group size. Right: count of companies controlled via maximum int. ownership.

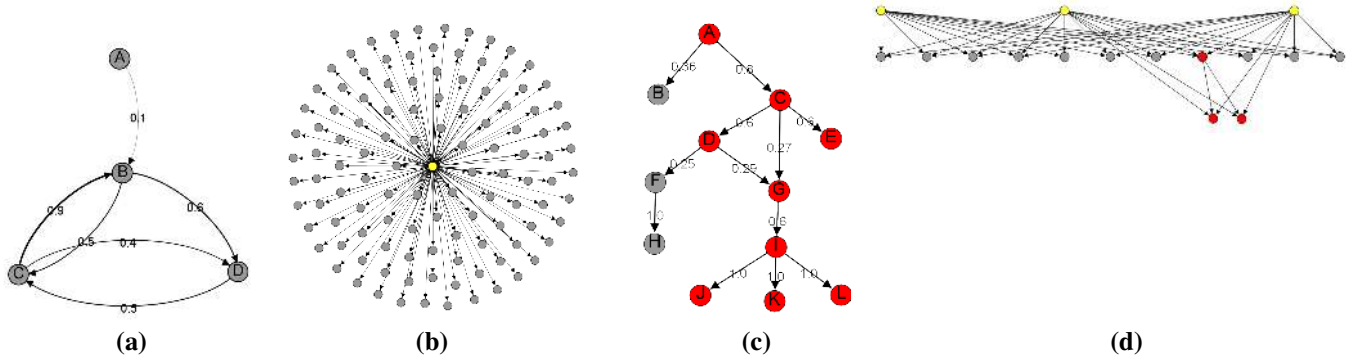


Fig. 7. Sample subgraphs: (a) high ownership-dividend difference; (b) “sunburst” corporate group; (c) a layered corporate group; (d) a family business group.

are family members who inherited their shares from relatives. In order to discover family corporate groups, we proceed with a rather general approach, which may be applied to discover groups organized with *a-priori* unknown schemes. First, a clustering algorithm is run over the graph of the integrated ownership matrix O . The intuition is that companies with similar property of shares should fall within a same cluster. We cluster integrated ownership rather than simply ownership, because the degree of indirect ownership is not explicitly taken into account by a graph clustering algorithm on the input of the ownership graph. In experiments, we adopted the Markov Cluster Algorithm (MCL) [16]. Next, we rank the interestingness of each cluster by the number of parent shareholders in the cluster. Basically, if there is only one parent shareholder, then the cluster is (part of) a corporate group. Otherwise, we consider nodes with no incoming links in the cluster as family members, and run a variant⁵ of Alg. 4 to compute the set of companies of the cluster controlled by the family members.

An example cluster is shown in Fig. 7 (d). It spans over 3 layers, and includes parent shareholder nodes only, i.e. no direct or indirect control exists between any pair of nodes. The three nodes in yellow are family members (they are actually brothers) who own about 1/3 shares respectively of each company at layer 2. There are also two companies at layer 3. They are owned directly by two family members and by the company shown in red at layer 2 (which is owned by all three family members). Notice that such a family group schema may be adopted also when the parent shareholders are not relatives,

⁵ C is initialized to the set of family members and line 2 is $S \leftarrow \cup_{i \in C} N_{out}(i)$.

Elapsed time for layered owner/dividend/corporate

<i>ExtractLayers()</i>	0.91s	0.91s	0.91s
Layered comput. on layers 1-15	32.84s	32.44s	3.32s
Flat computation on layer 16	5.01s	0.15s	0.01s
Layered comput. on layers 17-29	0.28s	0.04s	0.01s
Total elapsed time	39.04s	33.54s	4.25s

Elapsed time for flat owner/dividend/corporate

Flat computation on layers 1-29	> 1h	out-m.	11.34s
---------------------------------	------	--------	--------

TABLE I. ELAPSED TIMES ON THE ITALIAN NON-PARTNERSHIP SHARE GRAPH (2.474M NODES, 2.894M EDGES).

and not even individuals, but rather companies. The “lungs” sub-graph shown in Fig. 4 is such an example. There, the yellow nodes, which play the role of “family” members, are non-Italian companies.

C. Running times

Table. I reports the running times of the three steps of the layered ownership, dividend and corporate group algorithms. The input is the Italian non-partnership share graph. The test machine is a commodity PC with Intel Core i5-2410@2.30GHz with 16 Gb of RAM and Windows 7 OS. All flat and layered algorithms were implemented in Java 8, with no use of multi-threading and with all data structures (ownership graph, matrices O , D , R and C) stored in main memory. Matrix inversion in the integrated ownership solutions is implemented through the Scipy Python library on sparse linear algebra (docs.scipy.org). The flat solutions of the integrated ownership and dividend problems run out of time limit and out

of memory respectively. Regarding integrated ownership, the main problem is the inversion of the (sparse) matrix $2.474M \times 2.474M$. Regarding dividend, the out-of-memory issue is due to the huge number of simple paths to be stored in R . The small size of the recursive layer (2,370 nodes, which is three orders of magnitude smaller than the whole network) makes instead layered computations feasible in terms of memory occupation and efficient in terms of running time. The total running times range from a few seconds to less than 40 seconds. For comparison, the flat computation of the corporate group problem, which is the only flat algorithm which terminates, requires $2.6\times$ the time of the layered computation.

VII. CONCLUSIONS

A. Related work

A thorough analysis of ownership graphs from a network science perspective has been conducted in [8]. Economists are interested in forms of concentration of ownership and control. For instance, the network value of a node $v_i = \sum_j o_{ij} \nu_j$ is defined as the sum of all its integrated ownerships o_{ij} weighted by the intrinsic economic value ν_j of the owned company j . [8] provides a reinterpretation of the network value of a node in terms of flow and centrality of generic networks – and in particular in terms of $c(\alpha, \beta)$ -centrality [4]. Our contribution lies instead in exploiting the layered structure for algorithmic optimizations in the calculation of values o_{ij} . [8] also proposes a bow-tie view of ownership networks, which is compared in Sect. V-A with the layered structure introduced here. The dividend problem is introduced in this paper. Rubin’s algorithm [14], which is used at the core of the solution, has been exploited for finding paths connecting an individual in an online social network with friends of a friend at distance l [11]. However, density of social networks makes the approach unfeasible even for small values of l . On the contrary, sparsity and layered structure of ownership networks, allow for an efficient solution of the dividend problem. Regarding the corporate group problem, existing models of control propagation consider continuous values (vs our 0/1 model). The weakest link of a path $p = \langle i_1, \dots, i_k \rangle$ is $\min_{h=1}^{k-1} w_{i_h i_{h+1}}$. The sum of weakest links connecting i and j has been proposed in [7] as a measure of the control of j by i . Such an approach, however, does not take into account the presence of cycles. Integrated control [6] and relative majority [8] models start from a modified adjacency matrix W and apply the integrated approach of solving equation (1). Finally, a cornerstone empirical investigation of the relations between ownership and control is [12]. The paper covers 540 large firms from 27 countries. The largest dataset considered so far (see [8]) includes data of 24,877 companies and 106,141 shareholders from 47 countries, which is $30\times$ smaller than the Italian ownership graph.

B. Conclusion

The main contribution of this paper was in the interplay among modelling of ownership networks as layered graphs, algorithms exploiting efficiently such structure, and the empirical study of a significant case. We have translated key problems about corporate governance into analytical problems over graphs such as computing transitive closure (integrated ownership), all-pairs simple paths (dividends), and information

diffusion over graphs (control). The analysis of differences in the distributions of integrated ownership, dividends, and control groups reveals outlier (suspicious) behaviors that need further scrutiny by corporate governance experts, including economists, market control authorities, and policy makers.

Several open issues arise for future works, aimed at extending the approach to consider: multiple classes of stocks with different voting rights, different types of controlling shareholder (individuals, State, financial institutions, widely held companies), different minimum thresholds for controlling a company, integration of economy expert background knowledge for ranking those clusters of nodes not explainable as corporate or family business groups. The availability of data about the boards of directors of companies would raise additional related issues, such as relating control with sharing of directors between the controlled and the controlling company (interlocking [10]), and studying glass-ceiling or other forms of discrimination in groups of companies [13].

REFERENCES

- [1] H. V. Almeida and D. Wolfenzon, “A theory of pyramidal ownership and family business groups,” *The Journal of Finance*, vol. 61, no. 6, pp. 2637–2680, 2006.
- [2] J. Alstott, E. Bullmore, and D. Plenz, “powerlaw: A Python package for analysis of heavy-tailed distributions,” *PLoS ONE*, vol. 9, no. 1, p. e85777, 2004.
- [3] S. Baldone, F. Briochi, and S. Paleari, “Ownership measures among firms connected by cross-shareholdings and a further analogy with input-output theory,” in *Proc. of the 4th JAFEE International Conference on Investment and Derivatives*, 1998.
- [4] P. Bonacich, “Power and centrality: A family of measures,” *The American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.
- [5] F. Briochi, L. Buzzacchi, and M. G. Colombo, “Risk capital financing and the separation of ownership and control in business groups,” *Journal of Banking & Finance*, vol. 13, no. 4, pp. 747–772, 1989.
- [6] A. Chapelle and A. Szafarz, “Controlling firms through the majority voting rule,” *Physica A*, vol. 355, pp. 509–529, 2005.
- [7] S. Claessens, S. Djankov, and L. H. P. Lang, “The separation of ownership and control in East Asian corporations,” *Journal of Financial Economics*, vol. 58, no. 1–2, pp. 81–112, 2000.
- [8] J. B. Glattfelder, *Decoding Complexity: Uncovering Patterns in Economic Networks*, ser. Springer Theses. Springer, 2013.
- [9] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network,” in *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2003)*. ACM, 2003, pp. 137–146.
- [10] M. S. Mizruchi, “What do interlocks do? An analysis, critique, and assessment of research on interlocking directorates,” *Annual Review of Sociology*, vol. 22, no. 1, pp. 271–298, 1996.
- [11] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, “Scalable link prediction in social networks based on local graph characteristics,” in *Proc. of the Int. Conf. on Information Technology: New Generations (ITNG 2012)*. IEEE Computer Society, 2012, pp. 738–743.
- [12] R. L. Porta, F. Lopez-De-Silanes, and A. Shleifer, “Corporate ownership around the world,” *The Journal of Finance*, vol. 54, no. 2, pp. 471–517, 1999.
- [13] A. Romei and S. Ruggieri, “A multidisciplinary survey on discrimination analysis,” *The Knowledge Engineering Review*, vol. 29, no. 5, pp. 582–638, 2014.
- [14] F. Rubin, “Enumerating all simple paths in a graph,” *IEEE Transactions on Circuits and Systems*, vol. 25, no. 8, pp. 641–642, 1978.
- [15] A. Shleifer and R. Vishny, “A survey of corporate governance,” *The Journal of Finance*, vol. 52, no. 2, pp. 737–783, 1997.
- [16] S. van Dongen, “Graph clustering by flow simulation,” Ph.D. dissertation, University of Utrecht, 2000, <http://micans.org/mcl/>.