2010 International Conference on Advances in Social Networks Analysis and Mining

Detecting Social Positions using Simulation

Joel Brynielsson
FOI
SE 164 90 Stockholm, Sweden
Email: joel.brynielsson@foi.se

Johanna Högberg
Umeå University
SE 901 87 Umeå, Sweden
email: johanna@cs.umu.se

Lisa Kaati, Christian Mårtenson, Pontus Svenson
FOI
SE 164 90 Stockholm, Sweden
Email: lisa.kaati,christian.martenson,pontus.svenson@foi.se

***Abstract*—Describing social positions and roles is an important topic within social network analysis. One approach is to compute a suitable equivalence relation on the nodes of the target network. One relation that is often used for this purpose is *regular equivalence*, or *bisimulation*, as it is known within the field of computer science. In this paper we consider a relation from computer science called *simulation relation*. Simulation creates a partial order on the set of actors in a network and we can use this order to identify actors that have characteristic properties. The simulation relation can also be used to compute *simulation equivalence* which is a less restrictive equivalence relation than regular equivalence but is still computable in polynomial time. This paper primarily considers weighted directed networks and we present definitions of both weighted simulation equivalence and weighted regular equivalence. Weighted networks can be used to model a number of network domains, including information flow, trust propagation, and communication channels. Many of these domains have applications within homeland security and in the military, where one wants to survey and elicit key roles within an organization. Identifying social positions can be difficult when the target organization lacks a formal structure or is partially hidden.**

I. Introduction

Social network analysis (SNA) [1], [2], [3] is a set of powerful techniques to identify social roles, important groups and hidden organization structures. While SNA has a long and successful history within sociology, networks are everywhere in nature, and SNA and related methodology can be used to analyze a wide variety of different problems [4].

One particular application that has gained interest lately is for military and security purposes: can we use SNA to find criminals or terrorists? While it is easier to use SNA in forensic analyses after a crime or terrorist act has been committed, it is sometimes still hoped that SNA in conjunction with other techniques could allow us to obtain early warnings about future incidents. Correlation of observed data about individuals, things, places, memberships, etc., may be used to detect organized crime or terrorist cells and networks through the observation of hidden relations and co-occurrences. This methodology assumes that the ways the members of a group can and do communicate with each other are correlated with important properties of that group. There are a number of different measures that could be calculated for a given network and that tell us details about it.

Contact author: Pontus Svenson, ponsve@foi.se

A network consists of a set of actors and a set of binary relations between the actors that describe their communication patterns. Most networks of interest are very large, making it difficult to visualize them to a human user. In order to facilitate easier visualization and analysis, we want to construct a smaller network that still has the most important characteristics of the original one. One way of doing this is to discover the community structure of the network. Another is to look for actors in the network that are equivalent.

In this paper, we follow the latter route and investigate how different equivalence relations from computer science could be used to construct abstractions of large networks. The equivalence relations used could be used to distinguish between different social positions and roles in the network. The notions of social position and social role are standard concepts when analyzing social networks. The standard view is that a social position is a property of actors in a network. For a given social network a social position is defined as a subset of actors, namely those who have that property [5]. Positions are determined by an equivalence relation over the set of actors and the equivalence classes represent the different positions in a network. There are several different equivalence relations that can be used to describe social positions in a network: automorphic equivalence, structural equivalence and reqular equivalence are three of the most well-known equivalences in the literature [1]. Of these three relations, regular equivalence is the relation that is least restrictive. However, it is more common to approximate the relations to get a better result.

In this paper we use simulation equivalence to create equivalence classes that form positions in the network. The reduced network that is formed can be used as an abstraction of the larger network, and is easier to use for explorative visual analysis.

Finding such equivalence classes is interesting for many different applications. Our interest lies mainly in using the equivalence classes to produce abstractions of large networks that are easier for military intelligence analysts to study and visualise. An obvious application is for recommendation systems. Another application is in enterprise incentive management, where company bonuses should be divided as fairly as possible among all employees. In a large corporation, employees who cannot be simulated (i.e., replaced) by other employees could be regarded as more valuable than others, and thus receive larger bonuses. The lack of redundancy could also

978-0-7695-4138-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ASONAM.2010.52

be considered as a risk to the system. In such cases, measures should be taken to make sure that a backup is created. In a similar way, the relation could be used to find weak points in other networks, for instance transport networks or information processing networks.

The techniques that we present in this paper can be used to identify social roles and social relations on actors in a network. First, *regular equivalence* is used to group actors into equivalence classes based on their similarity in the network. For each actor in the network, the similarity measure is obtained by looking at the network properties which is hopefully a good approximation of the person's actual skill or rank. That is, the underlying assumption is that a person's communicative behavior reveals whether he/she is, e.g., a formal or informal leader at some level. This is an important part of military intelligence analysis when studying, e.g., terrorist organizations: to cluster people depending on their formal or informal status. Second, *simulation equivalence* is a somewhat less strict similarity measure that says that two persons are simulation equivalent if they simulate each other. Lastly, the partial order that the *simulation relation* produces is used to identify key actors in the network. Hence, the simulation relation can be used to produce a network illustrating how the equivalence classes are related to each other, e.g., identifying leaders as opposed to subordinates and so forth.

The paper is outlined as follows. Section II presents an example aimed at providing an intuitive understanding of the mathematics presented in Sections III and IV, where relevant definitions are given. After that, Section V presents an example in the form of a communication scenario that puts the presented abstraction techniques into context, while Section VI briefly describes military intelligence work and how it could benefit from abstraction techniques such as the one presented. The paper concludes with a summary and some suggestions for future work in Section VII.

II. An Intuitive Example

Analyzing equivalence classes is one way of making sense of the patterns of relations among actors in a network. The ability to define, theorize about, and analyze data in terms of equivalence is important since we want to be able to make generalizations about social behavior and social structure. In order to do this we think about actors as examples of categories. That is, sets of actors who are in some way defined as "equivalent."

To define regular equivalence, [1] uses an example where neighborhood bullies occupy the same social position, though in different neighborhoods, because they beat up some kid(s) and are scolded by some irate parent(s), but they do not necessarily beat up the same kid(s) nor are they scolded by the same parent(s).

The network in Figure 1 shows a set of actors and a set of ties representing the different relations mentioned above. Using regular equivalence, we get three different equivalence classes each containing two actors: $\{1, 2\}$, $\{3, 4\}$ and $\{5, 6\}$. The social position that the equivalence classes represents are: irate parents $\{1, 2\}$, bullies $\{3, 4\}$ and children $\{5, 6\}$ (irate parents scolds at neighborhood bullies that beats up their kids).

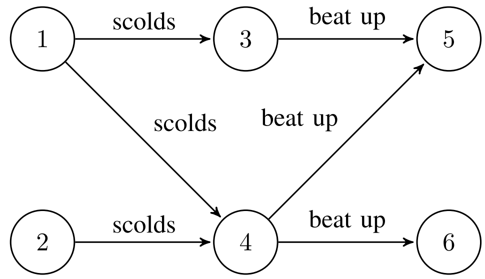

Fig. 1. A network representing parents who scolds at bullies beating up kids.

To illustrate the difference between simulation equivalence and regular equivalence we use a similar network as in Figure 1 with the modification that one of the bullies decides to start robbing some of his victims as well as beating them up. The network is shown in Figure 2. Using regular equivalence on the network we get the following equivalence classes : $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$ and $\{5, 6\}$. This means that the children are still equivalent and hold a common social positions, but the parents and the bullies do not.

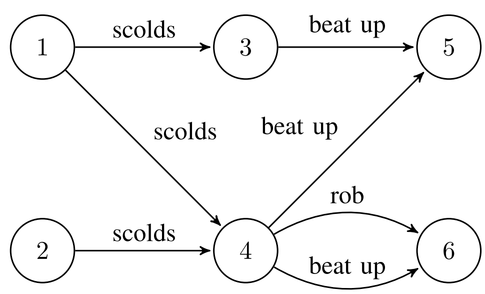

Fig. 2. A network representing parents who scolds at bullies beating up and robbing kids.

Using simulation equivalence we obtain the equivalence classes $\{1, 2\}$, $\{3\}$, $\{4\}$ and $\{5, 6\}$. The children $\{5, 6\}$ maintain their social position as *children*, the bullies (3 and 4) are now divided into two social positions: one containing the bully that only beats up his victims and the other containing the bully that beats up and sometimes also robs his victims. The parents, consisting of the equivalence class $\{1, 2\}$ form a social position of *irate parents that scolds bullies that are beating up and/or robbing their kids*. The fact that the social position of the parents are preserved although one of the bullies now robs some of his victims depends on the underlying properties of the simulation equivalence. Simulation equivalence can be desribed more formally as follows: an actor a simulates an actor b if a has *at least* ties to the same actors as a *or* to actors that simulate these. We say that two actors are simulation equivalent if they simulate each other.

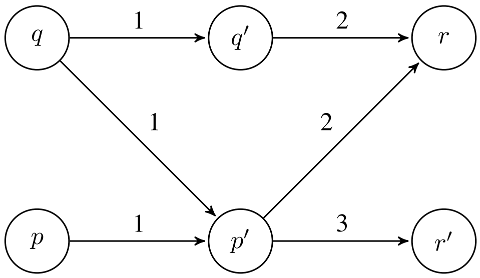

Fig. 3. The directed network G with weighted edges.

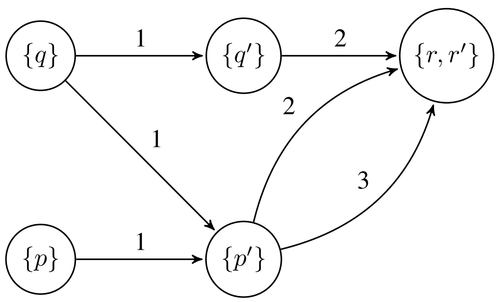

Fig. 4. The equivalence classes of network G using regular equivalence.

A. Weighted Directed Networks

In a more advanced approach we can add weights to the ties in the network. Using the same network structure as in the previous example but adding weights to the ties we obtain a network as shown in Figure 3.

Regular equivalence for weighted networks is a quite restrictive relation since two states are considered equivalent only if they have edges with similar weights to equivalent states. In this example we use weights from the semiring of integers. This means that two edges can be combined using addition of their weights. The network from Figure 3 where the regular equivalence classes are computed is shown in Figure 4. In this example the actors 5 and 6 form an equivalence class.

Using simulation equivalence, we obtain four equivalence classes as depicted in Figure 5. The fact that actor q and p form an equivalence class comes from the fact that actor p' simulates actor q', which means that every tie that actor q has to some equivalence class, actor p' also has (and possible more ties). Simulation equivalence is a less restrictive equivalence relation than regular equivalence, especially in the case with weights that are integers. Using regular equivalence, two states are considered equivalent only if they have similar ties *and* weights to other equivalence classes while in the case of simulation, one actor simulates another actor if the ties are similar and have *at least* the same weights (possibly more).

When computing simulation equivalence, a preorder is obtained. This preorder divides the states into a partial order and can be used to create an abstraction of a network. The abstraction is created by removing all actors that can be simulated by some other actor. In the network in Figure 3, the actors r and r' can be simulated by all other actors in the network, and can be removed from the resulting abstraction. Actor q' can be simulated by actor p' and q' can be replaced by p'. Actor p' cannot be simulated by any other actor in the network. Actors q and p can be simulate each other (that is, they are simulation equivalent).

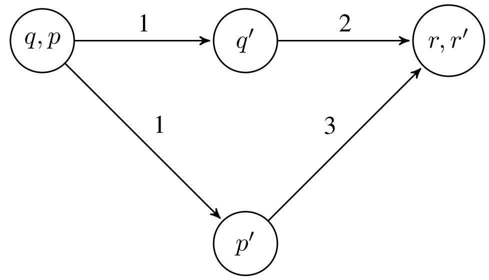

Fig. 5. The equivalence classes of network G using simulation equivalence.

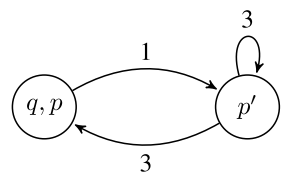

Fig. 6. An abstraction of the network G using simulation relation.

Figure 6 shows an abstraction of the network in Figure 3 where only the actors that cannot be simulated by other actors are represented.

III. Preliminaries

In this section we present some preliminaries on networks, relations and equivalences.

Numbers and relations Let S be a set, and let $\simeq$ and $\cong$ be binary relations on S, i.e., $\simeq$ and $\cong$ are both subsets of $S \times S$. We abbreviate the Cartesian product $S \times S$ by S^2. The relation $\cong$ is said to be *coarser* than $\simeq$ (or equivalently: $\simeq$ is a *refinement* of $\cong$), if $\simeq$ is a subset of $\cong$.

Let $\preceq$ be a binary relation on the set S. The relation $\preceq$ is a *preorder* (or *quasi-order*) if it is reflexive and transitive. It is an *equivalence relation* if it is a symmetric preorder. We write $\uparrow(s)$, where $s \in S$, for the set of elements $\{s' \mid s \preceq s'\}$ of elements in S that dominate s. We denote by $max_{\preceq}(S)$ the subset $\{s \mid \nexists s' \in S \colon s \preceq s' \wedge s' \not\preceq s\}$. That is, $max_{\preceq}(S)$ is the set of *maximal* elements of S with respect to $\preceq$. We denote by $=_{\preceq}$ the coarsest equivalence relation that is a subset of $\preceq$, i.e., $=_{\preceq}$ is equal to $\preceq \cap \preceq^{-1}$.

Example: Preorder Let $S = \{a, b, c\}$. The relation $\{(a,a),(b,b),(c,c),(a,b),(b,c),(a,c)\}$ is a preorder on S, but neither

$$\{(a,a),(a,b),(b,c),(a,c)\}$$

nor

$$\{(a,a),(b,b),(c,c),(a,b),(b,c)\}$$

are, as the first relation is not reflexive and the second is not transitive.

The *equivalence class* of an element $s \in S$ with respect to an equivalence relation $\simeq$ is the set $[s]_{\simeq} = \{s' \mid s \simeq s'\}$. Whenever $\simeq$ is obvious from the context, we simply write $[s]$ instead of $[s]_{\simeq}$. It should be clear that $[s]$ and $[s']$ are equal if s and s' are in relation $\simeq$, and disjoint otherwise. The equivalence relation $\simeq$ thus induces a partition $(S/\simeq) = \{[s] \mid s \in S\}$ of S.

Networks and Graphs An *alphabet* is a finite set of symbols. We write ε for the empty word.

Definition 1 (Directed graph): A *directed graph* (or network) is a tuple (V, E) where

- V is a finite set of vertices (or nodes), and
- $E \subseteq V \times V$ is a finite set of edges.

If G has vertices with labels in the alphabet Σ, then V can be partitioned into a Σ-indexed family $(V_a)_{a\in\Sigma}$ of sets of vertices. If G has edge labels in Σ, then E is a Σ-indexed family $(E_a)_{a\in\Sigma}$ of sets of edges, such that $E_a \subseteq V \times V$, for each $a \in \Sigma$. We thus only allow parallel edges if they are labeled with different symbols.

For each symbol $a \in \Sigma$, we denote by $E_a(u)$ the set of vertices $\{v \mid (u, v) \in E_a\}$ that can be reached from u along a-labeled edges. Similarly, for each $w \in \Sigma^*$, we have

$$E_w(u) = \begin{cases} \bigcup_{v\in E_a(u)} E_{w'}(v) & \text{if } w = aw', \text{ for some } a \in \Sigma \\ & \text{and } w' \in \Sigma^*, \text{ and} \\ \{u\} & \text{if } w = \epsilon. \end{cases}$$

Let Σ be an alphabet and $G = (V, E)$ a directed graph with edge-labels in Σ. The graph G has a *trace* $w \in \Sigma^*$ if there is a $v \in V$ such that $E_w(v) \neq \emptyset$. The *trace behavior* of a graph G is given by

$$L(G) = \bigcup_{v\in V} \{w \in \Sigma^* \mid E_w(v) \neq \emptyset\} \ .$$

Algebraic structures Towards the end of Section IV, we recall a number of results concerning weighted simulation, and for this we need the following algebraic concepts.

A *monoid* is a set A together with a binary operation $\cdot$ from $A \times A$ to A and an element 1 in A that satisfy the following two axioms:

- The operation $\cdot$ is *associative* in that for every three elements a, b, and c in A, it holds that $(a \cdot b) \cdot c$ is equal to $a \cdot (b \cdot c)$.
- The element 1 is the *neutral* element with respect to $\cdot$; i.e., we have $a \cdot 1 = 1 \cdot a = a$, for all $a \in A$.

A monoid $(A, \cdot, 1)$ is *commutative* if $a \cdot b = b \cdot a$, for all a, b in A.

Moreover, a *semiring* is a tuple $(A, +, \cdot, 0, 1)$ where

- $(A, +, 0)$ is a commutative monoid and
- $(A, \cdot, 1)$ is a monoid.
- For every a, b, and c in A, it holds that $(a + b) \cdot c$ equals $(a \cdot c) + (b \cdot c)$ and $a \cdot (b + c)$ equals $(a \cdot b) + (a \cdot c)$, i.e. the multiplicative operation *distributes* over the additive.
- For every a in A, we have that $a \cdot 0 = 0 \cdot a = 0$. In other words, the element 0 is *absorptive*.
- Finally, 0 and 1 are distinct elements.

IV. Simulation Relations

Before we delve into the formal definitions, let us start with an intuitive description of simulation relation and simulation equivalence. A simulation on a graph G is an ordering (more precisely, a preorder) of the vertices of G, such that if there is an edge $v \to v'$ in G, then for every vertice u that dominates v, there is also an edge $u \to u'$ to some vertex u' that dominates v'. In other words, if a vertex u simulates a vertex v, then the freedom of movement at u is at least as great as it is at v.

A maximal simulation equivalence is the coarsest equivalence relation contained in $\preceq$, i.e., $=_{\preceq}$. In a social network the equivalence classes corresponds to different social roles.

The simulation preorder $\preceq$ can also be used to gain information about the actors in a network. The preorder contains information on the relations between the actors and if they simulate each other. A subnetwork containing only actors from the maximal vertices of $\preceq$ is an over approximation of the original network. This technique normally produces a small network but it does not preserve the trace behavior of the graph: although every trace that can be found in the original graph can also be found in the reduced graph, the converse is not true in general.

A. Relations

In this paper we consider the problem of finding the coarsest possible equivalence relation $\simeq$ on the set of vertices of a graph. We can use this information to collapsing each equivalence class of $\simeq$ into a single vertex and produce a (hopefully) smaller graph.

Definition 2 (Aggregated graph): Let $G = (V, E)$ be a directed edge-labeled graph, and let $\simeq$ be an equivalence relation on the vertex-space V. The *reduced graph* $(G/\simeq)$ is the system $((V/\simeq), E')$ where

$$E' = \{([u]_{\simeq}, [u']_{\simeq}) \mid (u, u') \in E\} \ .$$

Note that E' is well-defined because $\simeq$ is an equivalence relation.

Definition 3 (Regular equivalence): An equivalence relation $\simeq \subseteq V \times V$ is a *regular equivlence* or *bisimulation* if $u \simeq v$, $a \in \Sigma$, and $v' \in E_a(v)$, implies that there is a $u' \in E_a(u)$ such that $u' \simeq v'$, and vice versa.

The coarsest regular equivalence can be computed in time $O(m \log n)$, where m is the number of edges and n the number of vertices of the input graph, using a divide-and-conquer technique by Hopcroft [6] that was generalized to the nondeterministic case by Paige and Tarjan [7]. The major drawback with regular equivalence is that it is unnecessarily strict, and thus provides an unnecessarily weak reduction of the vertex space [8].

A less restrivtive relation is *simulation preorder*.A vertex u *simulates* a vertex v if, for every symbol $a \in \Sigma$ and edge $(v, v') \in E_a$, there is an edge $(u, u') \in E_a$ such that u' simulates v'.

Definition 4 (Simulation preorder): A preorder relation $\preceq \subseteq V \times V$ is a *simulation* if the fact that $v \preceq u$, $a \in \Sigma$, and $v' \in E_a(v)$, implies that there is a $u' \in E_a(u)$ such that $v' \preceq u$.

A pair of vertices $u, v \in V$ are *simulation equivalent* if $\preceq$ is a simulation and both $u \preceq v$ and $v \preceq u$ hold.

If the graph G under consideration has has vertex-labels in the alphabet Σ, i.e. $V = (V_a)_{a \in \Sigma}$, then require that every simulation on V refines the partitioning $(V_a)_{a \in \Sigma}$.

A *regular equivalence* is thus a symmetric simulation preorder, i.e. a simulation relation that is also an equivalence relation. Since the two relations are so closely related, it should come as no surprise that simulation can be computed in time $O(mn)$ [9].

A simulation preorder is in general not an equivalence relation. Instead, we can use the coarsest equivalence contained in $\preceq$, namely $=_{\preceq}$. Simulation equivalence $=_{\preceq}$ produces, in general, larger equivalence classes than regular equivelence.

If we are willing to settle for an over-approximation of the trace behavior, we can use the simulation preorder and create an aggregated abstraction of the graph. We first re-route all edges to and from each non-maximal vertex v to each of the vertices in $\uparrow(v)$. We then drop all vertices that are not in $max_{\preceq}(\preceq)$ together with the edges connected to them.

Definition 5 (Simulation abstraction): Let $G = (V, E)$ be a directed edge-labeled graph, and let $\preceq$ be a simulation relation on the vertex-space V. The *reduced graph* $(G/\preceq)$ with respect to $\preceq$ is the system (V', E'), and where $V' = (max_{\preceq}(V)/\preceq)$

$$E' = \{([u]_{\simeq}, [u']_{\simeq}) \mid u \in \uparrow(v) \cap max_{\preceq}(V) \wedge u' \in \uparrow(v') \cap max_{\preceq}(V) \wedge (v, v') \in E\} \ .$$

a) **Example:** Consider the directed network G of Figure 7.

Computing regular equivalence over G gives us the equivalence classes {0},{1},{2},{3},{4},{5},$\{6, 7, 8, 9\}$, the aggregated network is shown in Figure 8. Using simulation equivalence we obtain the equivalence classes {0},{1,2}, {3},{4}, {5},$\{6, 7, 8, 9\}$, the aggregated network is depicted in Figure 9. Finally, using the simulation preorder abstraction, we obtain an aggregated network as depicted in

Figures 8, 9 and 10 show the graphs that arise after minimization with respect to each of these equivalence relations.

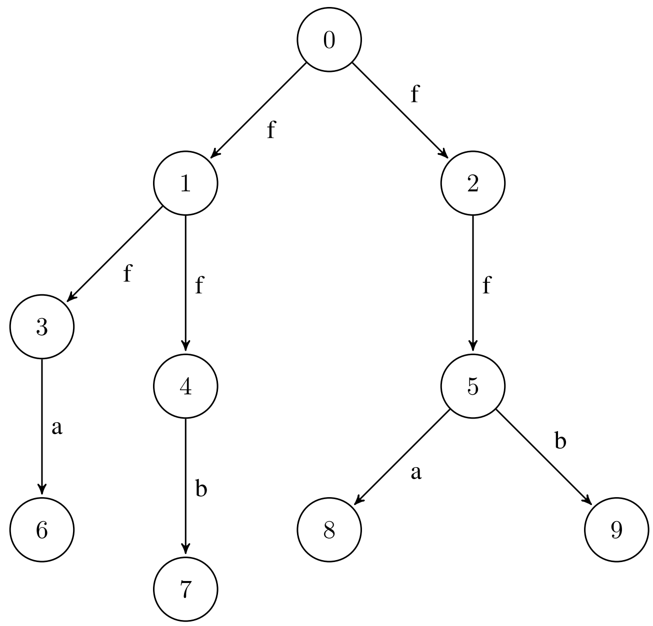

Fig. 7. The directed labelled graph G

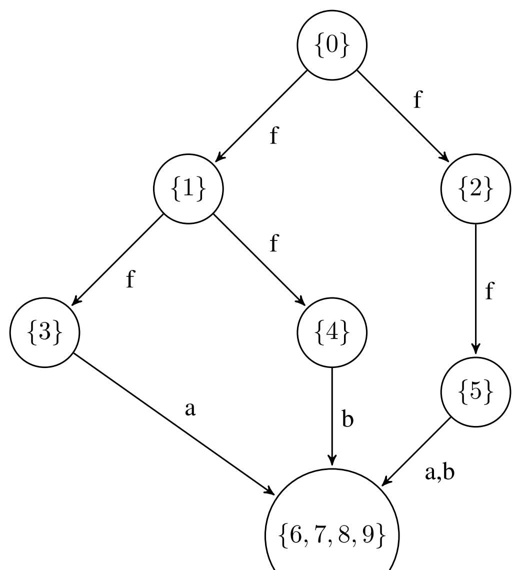

Fig. 8. The graph of Figure 7 after computing regular equivalence.

B. Weighted Relations

It is often useful to model quantitative as well as qualitative relations between vertices in a social network, such as trust, affection, or level of communication. The typical approach is to assign weights taken from some semiring A to the edges of the network, turning the edge relation into a (total) mapping that takes pairs of vertices into A.

Definition 6: A *weighted directed graph* or *weighted directed network* (over a semiring A) is a tuple $G = (V, E)$, where V is a finite set of vertices and $E \colon V \times V \to A$ is a mapping

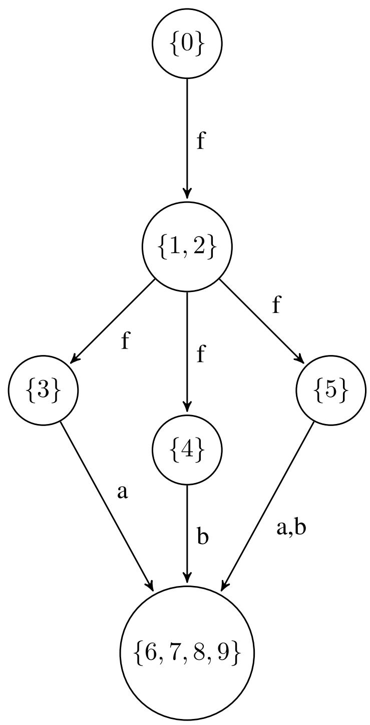

Fig. 9. The graph of Figure 7 after computing simulation equivalence.

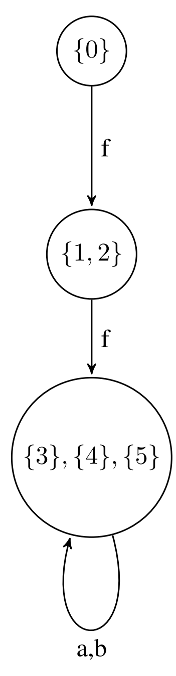

Fig. 10. The graph of Figure 7 after using simulation preorder abstraction. The nodes $\{6, 7, 8, 9\}$ did not contain maximal vertices and was consequently dropped.

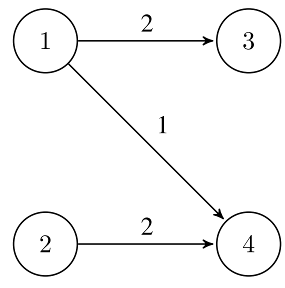

Fig. 11. A network with weighted edges.

that assigns to every ordered pair of vertices in V a weight in A.

If G has edge labels in Σ, the we consider E to be a family $(E_a)_{a\in\Sigma}$ of mappings such that $E_a\colon V \times V \rightarrow A$, for every $a \in \Sigma$.

Regular equivalence for weighted directed networks is defined as follows.

Definition 7 (Weighted regular equivalence): An equivalence relation $\simeq \subseteq V \times V$ is a *regular equivalence* if $u \simeq v$ and $v' \in E(v)$, implies that there is a $u' \in E(u)$ such that $E(u, u') = E(v, v')$ and $u' \simeq v'$, and vice versa.

The definition of weighted simulation is similar. Definitions for simulation relations for *weighted tree automata* (since a string is a special kind of tree, the definitions can be directly applicable to graphs) can be found in [10].

Definition 8 (Weighted simulation): The preorder $\preceq$ is a *weighted simulation* if $v \preceq u$ implies that, for each $v' \in V$ and $a \in \Sigma$, there is an $u' \in V$ such that $E(u, u') \geq E(v, v')$ and $v' \preceq u'$.

As one could notice, the difference between regular equivalence and simulation equivalence is more observable in the weighted setting. This is because the weights of the edges has to be strictly equal to fulfil the requirements in the case of regular equivalence while in the case of weighted simulation, the weight of an edge has to be equal or greater than to satisfy the required conditions. The difference can be illustrated using the network depicted in Figure 11. Using regular equivalence the nodes of the network can be divided into three equivalence classes: $\{1\}, \{2\}$ and $\{3, 4\}$. Using simulation equivalence the nodes of network can be divided into two equivalence classes $\{1, 2\}$ and $\{3, 4\}$.

V. EXAMPLE

Social networks play fundamental roles as mediums for spreading information, ideas and influence among their members. In the following example, we test our ability to find social positions in a network where each tie represents the possibility to communicate.

In the experiments reported on here, we used a network obtained from the communication pattern within a department consisting of 20 researchers, see Figure 12. The network is weighted and directed. A directed tie between two actors a and b represents the fact that actor a communicates with actor b. Each tie is also assigned a weight representing the possibilities that actor a has to communicate with actor b. In this experiment we assign a link between two actors the weight one if their regular communication is done using e-mail and the weight two if the communication is done using phone. This means that communicating via phone is worth twice as much as communicating using e-mail, indicating that real-time talking signifies a closer social relation than that of virtual relationships.

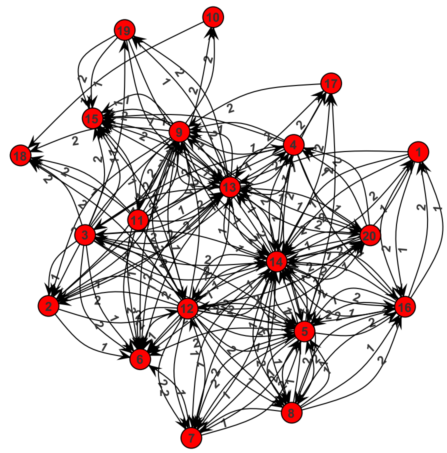

Fig. 12. The original network of communication.

Using regular equivalence for weighted networks according to Def. 7 we obtain 18 different equivalence classes, depicted in Figure 13, divided into 17 classes containing one actor and one class containing 3 actors. Further, computing simulation equivalence on the same graph results in a smaller graph with 5 different equivalence classes, depicted in Figure 14. Although these two graphs differ in size, they are still similar in that they contain *clusters of people* that are, to some extent, similar. What differs is the similarity measure: a less strict measure results in a smaller graph.

Using the preorder that was obtained when computing the simulation relation, we notice that one of the equivalence classes simulates all the other equivalence classes according to Table I. Therefore the resulting graph, see Figure 15, only contains one node. Theoretically, this indicates that we can replace the other equivalence classes with this dominating equivalence class and still preserve all communication that was present in the original network. Hence, it should be noted that the graph obtained in Figure 15 must be thought of in terms of *clusters of equivalence classes* rather than the clusters of individuals that are depicted in the graphs in Figures 13–14. Of course, this total reduction is due to the investigated social scenario (a research department) where all departmental members are, in some sense, comparable with respect to their duties. That is, since all members in the investigated organization are of the same type the graph collapses into one single node. In a more heterogeneous organization, the number of equivalence classes would of course correspond to the number of different types of personnel in the organization, e.g., a scientist cannot be assumed to be able to simulate an officer and vice versa since the jobs are so different.

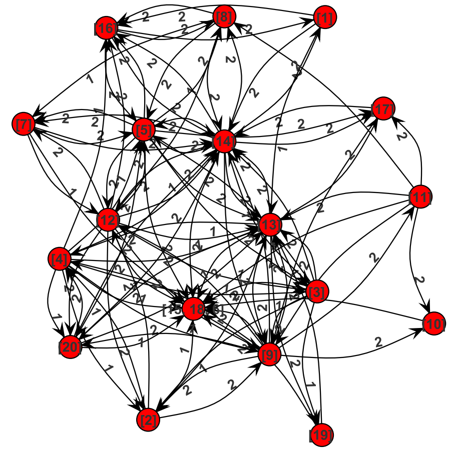

Fig. 13. Regular equivalence computed on the network of communication.

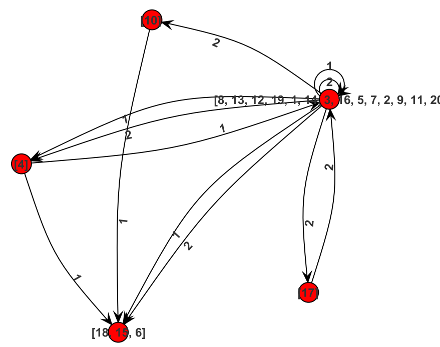

Fig. 14. Simulation equivalence computed on the network of communication.

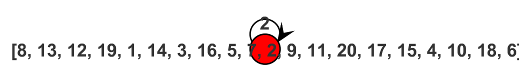

Fig. 15. Simulation preorder abstraction on the network of communication.

VI. Military Intelligence by Graph Reduction

Our application of interest is social network analysis within the military intelligence domain. Here, one wishes to consider and model a network of interesting people, e.g., a terrorist cell, that are connected to each other in various ways.

Military intelligence is largely an unknown business: the nature of the work and the resulting intelligence products make it vital to keep current practices, methods and techniques secret. However, on a more generic level it is apparent that a shift is currently taking place. Traditional intelligence work has been closely related to the so-called intelligence cycle

Actor	Simulated by actors
10	20, 17, 4, 16, 7, 12, 9, 10, 2, 14, 3, 11, 19, 1,5, 8, 13
4	20, 17, 4, 16, 7, 12, 9, 2, 14, 3, 11, 19, 1, 5, 8, 13
17	20, 17, 16, 7, 12, 9, 2, 14, 3, 11, 19, 1, 5, 8, 13
6,15,18	20, 15, 4, 16, 7, 12, 9, 10, 2, 14, 18, 3, 11, 19, 1, 5, 8, 13, 6
1,2,3,5,7,8,9,11, 12,13,14,16,19,20	1,2,3,5,7,8,9,11,12,13,14,16, 19,20

TABLE I
THE SIMULATION PREORDER. LEFT COLUMN CONTAINS ACTORS THAT ARE SIMULATION EQUIVALENT, RIGHT COLUMN ALL ACTORS THAT CAN SIMULATE THE ACTORS IN THE LEFT COLUMN.

where one plans, gathers documents, analyses these documents and delivers a report. This iterative process is ill-suited to the modern information age and therefore new computerized methods making use of continuous updating, multiple sources and automation changes the very foundations of military intelligence work and turns the traditional way that analysts' work into a target-centric intelligence loop where several sources contribute in parallel to a continuously updated situation picture [11]. For example, the combination of manual social network analysis, live data from a mobile phone communication network and field observations can yield new insight and better intelligence products. One consequence of this shift is that the analyst is faced with networks containing large numbers of vertices and edges that need to be analyzed quickly and continuously. Hence, efficient graph reduction techniques and tools for graph mining are foreseen to be vital ingredients in tomorrow's computer support for intelligence analysts. Given a large social network depicting a dark organization of some kind, the intelligence analyst could gain insight by finding graph patterns in many ways. The graph reduction techniques that we present in this paper help the analyst to discover important patterns within graph data that, in turn, give insights regarding important intelligence aspects regarding social roles and positions.

VII. SUMMARY AND FUTURE WORK

In this paper we present a relation from computer science called *simulation relation* that can be used to distinguish between different social positions and roles in a social network.

A simulation relation computed on the nework N is an ordering (a preorder) of the nodes of N. The ordering is such that if an actor a simulates an actor b then actor a has at least the same relations as b to other actors (or actors that simulate these). Actors in the network that simulate each other are considered to be simulation equivalent and each such equivalence class represents a social position.

We use simulation equivalence to describe social positions within a social network since simulation equivalence is a less restrictive equivalence relation than regular equivalence, automorphic equivalence and structural equivalence.

We use the simulation preorder to create an abstraction of a given social network. The abstraction contains every trace that is found in the original network but the size may be significantly smaller than that of the original network. The abstraction can be used to perform a worst case scenario analysis of the network.

The social positions and the ordering that are obtained using simulation equivalence are particulary interesting when we look at social networks describing different competences since we can use the positions to get an understanding of how the groups of the competences are composed.

We see many possibilities for future work based on the algorithm and ideas presented in this paper. First of all, it would be interesting to investigate if it is possible to define approximately simulation equivalence (as in the case of regular equivalence). Secondly, it would be interesting to conduct experiments on other, publically-available, data sources. In order to validate the usefulness of the relation for military intelligence analysis, it is necessary to implement the functionality into an SNA tool (such as the one used in [12]) and conduct user experiments. Finally, it would be interesting to use the ideas in this paper to consider uncertain data. In this case, we might not know for sure that there is an edge between two actors, instead we only have a probability that there exists a link.

This work was supported by the FOI research project "Tools for information management and analysis", which is funded by the R&D programme of the Swedish Armed Forces.

REFERENCES

[1] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
[2] J. P. Scott, *Social Network Analysis: A Handbook*. Sage Publications, 2000.
[3] P. J. Carrington, J. Scott, and S. Wasserman, Eds., *Models and Methods in Social Network Analysis*. Cambridge University Press, 2005.
[4] M. Newman, A. Barabasi, and D. J. Watts, Eds., *The Structure and Dynamics of Networks*. Princeton University Press, 2006.
[5] M. Marx and M. Masuch, "Regular equivalence and dynamic logic," *Social Networks*, vol. 25, 2003.
[6] J. Hopcroft, "An $n \log n$ algorithm for minimizing states in a finite automaton," in *Theory of Machines and Computations*, Z. Kohavi, Ed. Academic Press, 1971.
[7] R. Paige and R. Tarjan, "Three partition refinement algorithms," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 973–989, 1987.
[8] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke, "Computing simulations on finite and infinite graphs," in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 1995, pp. 453–462.
[9] P. A. Abdulla, A. Bouajjani, L. Holík, L. Kaati, and T. Vojnar, "Composed bisimulation for tree automata," in *CIAA*, 2008, pp. 212–222.
[10] A. Maletti, "A backward and a forward simulation for weighted tree automata," in *Proc. 3rd Int. Conf. Algebraic Informatics*, 2009, submitted.
[11] R. M. Clark, *Intelligence Analysis: A Target-Centric Approach*. Washington, District of Columbia: CQ Press, 2004.
[12] L. Ferrara, C. Mårtenson, P. Svenson, P. Svensson, J. Hidalgo, A. Molano, and A. Madsen, "Integrating data sources and network analysis tools to support the fight against organized crime," ser. Lecture Notes in Computer Science, vol. 5075. Springer Verlag, 2008, pp. 171–182.