9 Role Assignments

Jürgen Lerner

Classification is the key to understand large and complex systems that are made up of many individual parts. For example in the study of food webs (networks that consist of living organisms and predator-prey relationships, flow of protein, etc.) it is, even for moderately small ecosystems, impossible to understand the relationship between each pair of individual organisms. Nevertheless, we can understand the system – to a certain extent – by classifying individuals and describing relationships on the class level. Classification in networks aims to describe regular patterns of interaction and to highlight essential structure, which remains stable over long periods of time.

In this chapter we formalize the classification of vertices in a graph, such that vertices in the same class can be considered to occupy the same *position*, or play the same *role* in the network. This idea of network position or role, see e. g., Nadel [436], has been formalized first by Lorrain and White [394] by a special type of vertex partition. They proposed that vertices play the same role if they have identical neighborhoods. Subsequent work like Sailer [501] and White and Reitz [579] generalized this early definition, weakening it sufficiently to make it more appropriate for modeling social roles. All these definitions have in common that vertices which are claimed to play the same role must have something in common w. r. t. the relations they have with other vertices, i. e., a generic problem definition for this chapter can be given by

given a graph $G = (V, E)$,
find a partition of V that is *compatible* with E.

The generic part here is the term 'compatible with E'. In this chapter, we present definitions for such compatibility requirements, and properties of the resulting classes of vertex-partitions.

Outline of this chapter. The remainder of this section treats preliminary notation. In Sections 9.1 through 9.3, different types of role assignments are introduced and investigated. In Section 9.4 definitions are adapted to graphs with multiple relations (see Definition 9.4.1) and in Section 9.5 composition of relations is introduced and its relationship to role assignments is investigated.

Sections 9.1 through 9.3 follow loosely a common pattern: After defining a compatibility requirement, some elementary properties of the so-defined set of role assignment are mentioned. Then, we investigate a partial ordering on this set, present an algorithm for computing specific elements, and treat the

U. Brandes and T. Erlebach (Eds.): Network Analysis, LNCS 3418, pp. 216–252, 2005.

complexity of some decision problems. We provide a short conclusion for each type of vertex partition, where we dwell on the applicability for defining role assignments in empirical networks.

The most complete investigation is for regular equivalences in Section 9.2. Although there is some scepticism as to whether regular equivalences are a good formalization of role assignments in real social networks, we have chosen to treat them prominently in this chapter, since their investigation is exemplary for the investigation of types of role assignments. The results for regular equivalences are often translatable to other types of equivalences, often becoming easier or even trivial. We emphasize this generality when appropriate.

Graph model of this chapter. In this chapter, *graph* usually means directed graph, possibly with loops. Except for Sections 9.2.4 and 9.2.5, where graph means undirected graph, Section 9.3.1, where results are for undirected multigraphs, and Sections 9.4 and 9.5, where we consider graphs with multiple relations (see Definition 9.4.1).

9.0.1 Preliminaries

In the following, we will often switch between vertex partitions, equivalence relations on the vertex set, or role assignments, since, depending on the context, some point of view will be more intuitive than the other. Here we establish that these are just three different formulations for the same underlying concept.

Let V be a set. An *equivalence relation* $\sim$ is a binary relation on V that is *reflexive*, *symmetric*, and *transitive*, i. e., $v \sim v$, $u \sim v$ implies $v \sim u$, and $u \sim v \wedge v \sim w$ implies $u \sim w$, for all $u, v, w \in V$. If $v \in V$ then $[v] := \{u\,;\ u \sim v\}$ is its *equivalence class*.

A *partition* $\mathcal{P} = \{C_1, \ldots, C_k\}$ of V is a set of non-empty, disjoint subsets $C_i \subseteq V$, called *classes* or *blocks*, such that $V = \bigcup_{i=1}^{k} C_i$. That is, each vertex $v \in V$ is in exactly one class.

If $\sim$ is an equivalence relation on V, then the set of its equivalence classes is a partition of V. Conversely, a partition $\mathcal{P}$ induces an equivalence relation by defining that two vertices are equivalent iff they belong to the same class in $\mathcal{P}$. These two mappings are mutually inverse.

Definition 9.0.1. *A* role assignment *for V is a surjective mapping $r \colon V \to W$ onto some set W of* roles*.*

The requirement *surjective* is no big loss of generality since we can always restrict a mapping to its image set. One could also think of role assignments as vertex-colorings, but note that we do not require that adjacent vertices must have different colors. We use the terms role and position synonymously.

A role assignment defines a partition of V by taking the inverse-images $r^{-1}(w) := \{v \in V\,;\ r(v) = w\}$, $w \in W$ as classes. Conversely an equivalence relation induces a role assignment for V by the class mapping $v \mapsto [v]$. These two mappings are mutually inverse, up to isomorphism of the set of roles.

We summarize this in the following remark.

Remark 9.0.2. For each partition there is a unique associated equivalence relation and a unique associated role assignment and the same holds for all other combinations.

For the remainder of this chapter, definitions for vertex partitions translate to associated equivalence relations and role assignments.

9.0.2 Role Graph

The image set of a role assignment can be supplied naturally with a graph structure. We define that roles are adjacent if there are adjacent vertices playing these roles:

Definition 9.0.3. *Let $G = (V, E)$ be a graph and $r \colon V \to W$ a role assignment. The* role graph $R = (W, F)$ *is the graph with vertex set W (the set of roles) and edge set $F \subseteq W \times W$ defined by*

$$F := \{(r(u), r(v))\,;\ \exists u, v \in V \text{ such that } (u, v) \in E\}\ .$$

R is also called quotient *of G over r.*

The role graph R models roles and their relations. It can also be seen as a smaller model for the original graph G. Thus, a role assignment can be seen as some form of network compression. Necessarily, some information will get lost by such a compression. The goal of role analysis is to find role assignments such that the resulting role graph displays essential structural network properties, i. e., that not too much information will get lost.

Thus we have two different motivations for finding good role assignments. First to know which individuals (vertices) are ‘similar’. Second to reduce network complexity: If a network is very large or irregular, we can't capture its structure on the individual (vertex) level but perhaps on an aggregated (role) level. The hope is that the role graph highlights essential and more persistent network structure. While individuals come and go, and behave rather irregularly, roles are expected to remain stable (at least for a longer period of time) and to display a more regular pattern of interaction.

9.1 Structural Equivalence

As mentioned in the introduction, the goal of role analysis is to find meaningful vertex partitions, where ‘meaningful’ is up to some notion of compatibility with the edges of the graph. In this section the most simple, but also most restrictive requirement of compatibility is defined and investigated. Lorrain and White [394] proposed that individuals are role equivalent if they are related to the same individuals.

Definition 9.1.1. *Let $G = (V, E)$ be a graph, and $r \colon V \to W$ a role assignment. Then, r is called* strong structural *if equivalent vertices have the same (out- and in-)neighborhoods, i. e., if for all $u, v \in V$*

$$r(u) = r(v) \Longrightarrow N^+(u) = N^+(v) \text{ and } N^-(u) = N^-(v)\ .$$

Remember Remark 9.0.2: Definitions for role assignments translate to associated partitions and equivalence relations.

Remark 9.1.2. By Definition 9.0.3 it holds for any role assignment r that, if (u, v) is an edge in the graph, then $(r(u), r(v))$ is an edge in the role graph. If r is strong structural, then the converse is also true. This is even an equivalent condition for a role assignment to be strong structural [579]. That is, a role assignment r is strong structural if and only if for all $u, v \in V$, it holds that $(r(u), r(v))$ is an edge in the role graph if and only if (u, v) is an edge in the graph.

We present some examples for strong structural equivalences. The identity mapping $\mathrm{id}\colon V \to V;\ v \mapsto v$ is strong structural for each graph $G = (V, E)$ independent of E. Some slightly less trivial examples are shown in Figure 9.1. For the star, the role assignment that maps the central vertex onto one role and all other vertices onto another, is strong structural. The bipartition of a complete bipartite graph is strong structural. The complete graph without loops has no strong structural role assignment besides id, since the neighborhood of each vertex v is the only one which does not contain v.

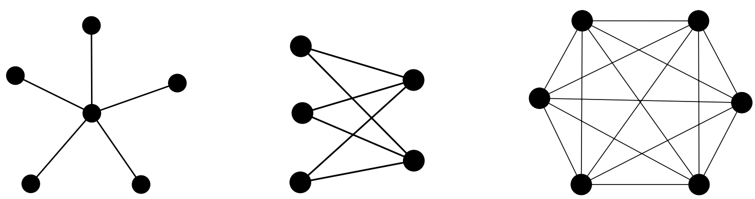

Fig. 9.1. Star (left), complete bipartite graph (middle) and complete graph (right)

We note some elementary properties. A class of strong structurally equivalent vertices is either an independent set (induces a subgraph without edges) for the graph or a clique with all loops. In particular, if two adjacent vertices u, v are strong structurally equivalent, then both (u, v) and (v, u) are edges of the graph, and both u and v have a loop.

The undirected distance of two structurally equivalent (non-isolated) vertices is at most 2. For if u and v are structurally equivalent and u has a neighbor w then w is also a neighbor of v. Thus, structural equivalence can only identify vertices that are near each other.

Although in most irregular graph there won't be any non-trivial structural equivalence, the set of structural equivalences might be huge. For the complete graph with loops, every equivalence is structural. In Section 9.1.2, we investigate a partial order on this set.

Variations of structural equivalence. The requirement that strong structurally equivalent adjacent vertices must have loops has been relaxed by some authors.

Definition 9.1.3 ([191]). *An equivalence $\sim$ on the vertex set of a graph is called* structural *if for all vertices $u \sim v$ the transposition of u and v is an automorphism of the graph.*

White and Reitz [579] gave a slightly different definition, which coincides with Definition 9.1.3 on loopless graphs.

9.1.1 Lattice of Equivalence Relations

The set of equivalence relations on a set V is huge. Here we show that this set naturally admits a partial order, which turns out to be a lattice. (For more on lattice theory, see e. g., [261].) This section is preliminary for Sections 9.1.2 and 9.2.2.

Equivalence relations on a set V are subsets of $V \times V$, thus they can be partially ordered by set-inclusion ($\sim_1 \leq \sim_2$ iff $\sim_1 \subseteq \sim_2$). The equivalence relation $\sim_1$ is then called *finer* than $\sim_2$ and $\sim_2$ is called *coarser* than $\sim_1$. This partial order for equivalences translates to associated partitions and role assignments (see remark 9.0.2).

In partially ordered sets, two elements are not necessarily comparable. In some cases we can at least guarantee the existence of lower and upper bounds.

Definition 9.1.4. *Let X be a set that is partially ordered by $\leq$ and $Y \subseteq X$.*

$y^* \in X$ is called an upper bound *(a* lower bound*) for Y if for all $y \in Y$, $y \leq y^*$ ($y^* \leq y$).*

$y^* \in X$ is called the supremum *(*infimum*) of Y, if it is an upper bound (lower bound) and for each $y' \in X$ that is an upper bound (lower bound) for Y, it follows $y^* \leq y'$ ($y' \leq y^*$). The second condition ensures that suprema and infima (if they exist) are unique.*

The supremum of Y is denoted by $\sup(Y)$ *the infimum by* $\inf(Y)$*. We also write* $\sup(x, y)$ *or* $\inf(x, y)$ *instead of* $\sup(\{x, y\})$ *or* $\inf(\{x, y\})$*, respectively.*

A lattice *is a partially ordered set L, such that for all $a, b \in L$,* $\sup(a, b)$ *and* $\inf(a, b)$ *exist.* $\sup(a, b)$ *is also called the* join *of a and b and denoted by $a \vee b$.* $\inf(a, b)$ *is also called the* meet *of a and b and denoted by $a \wedge b$.*

If $\sim_1$ and $\sim_2$ are two equivalence relations on V, then their intersection (as sets) is the infimum of $\sim_1$ and $\sim_2$. The supremum is slightly more complicated. It must contain all pairs of vertices that are equivalent in either $\sim_1$ or $\sim_2$, but also vertices that are related by a chain of such pairs: The *transitive closure* of a relation $R \subseteq V \times V$ is defined to be the relation $S \subseteq V \times V$, where for all $u, v \in V$

$$\begin{aligned} uSv \Leftrightarrow\ & \exists k \in \mathbb{N},\ \exists w_1, \ldots, w_k \in V \text{ such that} \\ & u = w_1,\ v = w_k,\ \text{ and } \forall i = 1, \ldots, k-1 \text{ it is } w_i R w_{i+1} \ . \end{aligned}$$

The transitive closure of a symmetric relation is symmetric, the transitive closure of a reflexive relation is reflexive and the transitive closure of any relation is transitive.

It follows that, if $\sim_1$ and $\sim_2$ are two equivalence relations on V, then the transitive closure of their union is the supremum of $\sim_1$ and $\sim_2$.

We summarize this in the following theorem.

Theorem 9.1.5. *The set of equivalence relations is a lattice.*

The interpretation in our context is the following: Given two equivalence relations identifying vertices that play the same role, there exists a uniquely defined smallest equivalence identifying all vertices which play the same role in either one of the two original equivalences. Moreover, there exists a uniquely defined greatest equivalence distinguishing between actors which play a different role in either one of the two original equivalences.

9.1.2 Lattice of Structural Equivalences

It can easily be verified that if $\sim_1$ and $\sim_2$ are two strong structural equivalences for a graph, then so are their intersection and the transitive closure of their union.

Proposition 9.1.6. *The set of strong structural equivalences of a graph is a sublattice of the lattice of all equivalence relations.*

In particular there exist always a maximum structural equivalence (MSE) for a graph.

The property of being strong structural is preserved under refinement:

Proposition 9.1.7. *If $\sim_1 \leq \sim_2$ and $\sim_2$ is a strong structural equivalence, then so is $\sim_1$.*

Although the above proposition is very simple to prove, it is very useful, since it implies that the set of all structural equivalences of a graph is completely described by the MSE. In the next section we present a linear time algorithm for computing the MSE of a graph.

9.1.3 Computation of Structural Equivalences

Computing the maximal strong structural equivalence for a graph $G = (V, E)$ is rather straight-forward. Each vertex $v \in V$ partitions V into 4 classes (some of which may be empty): Vertices which are in $N^+(v)$, in $N^-(v)$, in both, or in none.

The basic idea of the following algorithm 21 is to compute the intersection of all these partitions by looking at each edge at most twice. This algorithm is an adaption of the algorithm of Paige and Tarjan [459, Paragraph 3] (see Section 9.2.3) for the computation of the regular interior, to the much simpler problem of computing the MSE.

The correctness of algorithm 21 follows from the fact that it divides exactly the pairs of vertices with non-identical neighborhoods.

An efficient implementation requires some datastructures, which will be presented in detail since this is a good exercise for understanding the much more complicated algorithm in Section 9.2.3.

Algorithm 21: Computation of the maximal strong structural equivalence (MSE) of a graph

```
Input: a graph G = (V, E)
begin
    maintain a partition P = {C_1, ..., C_k} of V, which initially is the complete
    partition P = {V}
    // at the end, P will be the MSE of G
    foreach v ∈ V do
        foreach class C to which a vertex u ∈ N^+(v) belongs to do
            create a new class C' of P
            move all vertices in N^+(v) ∩ C from C to C'
            if C has become empty then
                remove C from P
        foreach class C to which a vertex u ∈ N^-(v) belongs to do
            create a new class C' of P
            move all vertices in N^-(v) ∩ C from C to C'
            if C has become empty then
                remove C from P
end
```

– A graph $G = (V, E)$ must permit access to the (out-/in-)incidence list of a vertex v in time proportional to the size of this list.
– Scanning all elements of a list must be possible in linear time.
– An edge must permit access to its source and its target in constant time.
– A partition must allow insertion and deletion of classes in constant time.
– A class must allow insertion and deletion of vertices in constant time.
– A vertex must permit access to its class in constant time.

The requirements on partitions and classes are achieved if a partition is represented by a doubly linked list of its classes and a class by a doubly linked list of its vertices.

One refinement step (the outer loop) for a given vertex v is performed as follows.

1. Scan the outgoing edges of v. For each such edge (v, u), determine the class C of u and create an associated block C' if one does not already exist. Move u from C to C'.
2. During the scanning, create a list of those classes C that are split. After the scanning process the list of split classes. For each such class C mark C' as no longer being associated with C and eliminate C if C is now empty.
3. Scan the incoming edges of v and perform the same steps as above.

A loop for a given v runs in time proportional to the degree of v, if v is non-isolated and in constant time else. An overall running time of $\mathcal{O}(|V| + |E|)$ follows, which is also an asymptotic bound for the space requirement.

Conclusion. Structural equivalence is theoretically and computationally very simple. It is much too strict to be applied to irregular networks and only vertices that have distance at most 2, can be identified by a structural equivalence. Nevertheless, structural equivalence is the starting point for many relaxations (see Chapter 10).

9.2 Regular Equivalence

Regular equivalence goes back to the idea of *structural relatedness* of Sailer [501], who proposed that actors play the same role if they are connected to role-equivalent actors – in contrast to structural equivalence, where they have to be connected to identical actors. Regular equivalence has first been defined precisely by White and Reitz in [579]. Borgatti and Everett (e. g., [191]) gave an equivalent definition in terms of colorings (here called role assignments). A coloring is regular if vertices that are colored the same, have the same colors in their neighborhoods. If $r\colon V \to W$ is a role assignment and $U \subseteq V$ then $r(U) := \{r(u)\,;\ u \in U\}$ is called the *role set* of U.

Definition 9.2.1. *A role assignment* $r\colon V \to W$ *is called* regular *if for all* $u, v \in V$

$$r(u) = r(v) \quad \Longrightarrow \quad r(N^+(u)) = r(N^+(v)) \text{ and } r(N^-(u)) = r(N^-(v)) \ .$$

The righthand side equations are equations of sets. There are many more equivalent definitions, (see e. g., [579, 90]).

Regular role assignments are often considered as *the* class of role assignments. The term *regular* is often omitted in literature.

Regular equivalence and bisimulation. Marx and Masuch [408] pointed out the close relationship between regular equivalence, bisimulation, and dynamic logic. A fruitful approach to find good algorithms for regular equivalence is to have a look at the bisimulation literature.

9.2.1 Elementary Properties

In this section we note some properties of regular equivalence relations.

The *identity* mapping $\mathrm{id}\colon V \to V;\ v \mapsto v$ is regular for all graphs. More generally, every structural role assignment is regular.

The next proposition characterizes when the complete partition, which is induced by the constant role assignment $J\colon V \to 1$ is regular. A *sink* is a vertex with zero outdegree, a *source* is one with zero indegree.

Proposition 9.2.2 ([82]). *The complete partition of a graph* $G = (V, E)$ *is regular if and only if* G *contains neither sinks nor sources or* $E = \emptyset$.

Proof. *If*: If $E = \emptyset$ then the righthand side in definition 9.2.1 is simply $\emptyset = \emptyset$, thus each role assignment is regular. If G has neither sinks nor sources, then, for all $v \in V$, $J(N^+(v)) = J(N^-(v)) = \{1\}$ and the equations in Definition 9.2.1 are satisfied for all $u, v \in V$.

Only if: Suppose $E \neq \emptyset$ and let $v \in V$ be a sink. Since $E \neq \emptyset$ there exists $u \in V$ with non-zero outdegree. But then

$$J(N^+(v)) = \emptyset \neq \{1\} = J(N^+(u)) \ ,$$

but $J(u) = 1 = J(v)$, thus J is not regular. The case of G containing a source is treated analogously. □

The identity and the complete partition are called *trivial* role assignments. The next lemma is formulated in [190] for undirected connected graphs, but it has a generalization to strongly connected (directed) graphs.

Lemma 9.2.3. *Let G be a strongly connected graph. Then in any non-trivial role assignment r of G, neither $\{r(v)\} = r(N^+(v))$ nor $\{r(v)\} = r(N^-(v))$ holds for any vertex v.*

Proof. If for some vertex v it is $\{r(v)\} = r(N^+(v))$, then the same would need to be true for each vertex in $N^+(v)$. Hence each vertex in successive out-neighborhoods would be assigned the same role and since G is strongly connected it follows that $r(V) = \{r(v)\}$ contradicting the fact that the role assignment is non-trivial. The case of $\{r(v)\} = r(N^-(v))$ for some vertex v is handled equally. □

A graph with at least 3 vertices whose only regular role assignments are trivial is called *role primitive*. The existence of directed role primitive graphs is trivial: For every directed path only the identity partition is regular. Directed graphs which have exactly the identity and the complete partition as regular partitions are for example directed cycles of prime length, since every non-trivial regular equivalence induces a non-trivial divisor of the cycle length.

The existence of undirected role primitive graphs is non-trivial.

Theorem 9.2.4 ([190]). *The graph in Figure 9.2 is role primitive.*

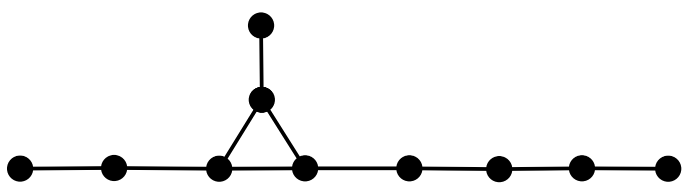

Fig. 9.2. A role-primitive undirected graph

The proof goes by checking that all possible role assignments are either non regular or trivial, where one can make use of the fact that the pending paths of the graph in Figure 9.2 largely diminish the possibilities one has to follow. The proof is omitted here.

A graph in which any role assignment is regular is called *arbitrarily role-assignable*. The next lemma is formulated in [190] for undirected connected graphs.

Lemma 9.2.5. *A strongly connected graph $G = (V, E)$ is arbitrarily role-assignable if and only if it is a complete graph, possibly with some but not necessarily all loops.*

Proof. Let $G = (V, E)$ be a graph satisfying the condition of the lemma and let r be any role assignment. We have to show that for all vertices $u, v \in V$

$$r(u) = r(v) \Longrightarrow r(N^+(u)) = r(N^+(v)) \text{ and } r(N^-(u)) = r(N^-(v)) \ .$$

If $u = v$ this is trivial. Otherwise u and v are connected by a bidirected edge, i. e., the role sets of their in- and out- neighborhoods contain $r(u)$. These role sets also contain all other roles since u and v are connected to all other vertices. So the role sets of the in- and out- neighborhoods of both vertices contain all roles, whence they are equal.

Conversely, let $G = (V, E)$ be a graph with two vertices u and v, such that $u \neq v$ and $(u, v) \notin E$. We assign $V \setminus \{v\}$ one role and v a different one. This is a non-trivial role assignment (note that $n > 2$, since G is connected) with $r(u) = r(N^+(u))$. So by Lemma 9.2.3 this role assignment can't be regular. □

9.2.2 Lattice Structure and Regular Interior

We have seen that the set of regular equivalences of a graph might be huge. In this section we prove that it is a lattice. See the definition of a lattice in Section 9.1.1.

Theorem 9.2.6 ([82]). *The set of all regular equivalences of a graph G forms a lattice, where the supremum is a restriction of the supremum in the lattice of all equivalences.*[1]

Proof. By Lemma 9.2.7, which will be shown after the proof of this theorem, it suffices to show the existence of suprema of arbitrary subsets. The identity partition is the minimal element in the set of regular equivalences, thus it is the supremum for the empty set. Hence we need only to consider the supremum for non-empty collections of regular role assignments. Since the set of all equivalences of a graph is finite, it even suffices to show the existence of the supremum of two regular equivalences.

So let $\sim_1$ and $\sim_2$ be two regular equivalences on G. Define $\equiv$ to be the transitive closure of the union of $\sim_1$ and $\sim_2$.

As mentioned in Section 9.1.1, $\equiv$ is the supremum of $\sim_1$ and $\sim_2$ in the lattice of all equivalences, so it is an equivalence relation and it is a supremum of $\sim_1$

[1] For the infimum see proposition (9.2.9).

and $\sim_2$ with respect to the partial order (which is the same in the lattice of all equivalences and in the lattice of regular equivalences). Therefore it remains to show that $\equiv$ is regular.

For this suppose that $u \equiv v$ and let $x \in N^+(u)$ for $u, v, x \in V$. Since $u \equiv v$ there exists a sequence $u, w_2, \ldots, w_{k-1}, v \in V$ where $u \sim_{j_1} w_2$, $j_1 \in \{1, 2\}$. Since $\sim_{j_1}$ is regular and $x \in N^+(u)$, there exists an $x_2 \in V$ such that $x_2 \in N^+(w_2)$ and $x_2 \sim_{j_1} x$. Iterating this will finally produce an x_k such that $x_k \in N^+(v)$ and $x \equiv x_k$, which shows the condition for the out-neighborhood. The case $x \in N^-(u)$ is handled analogously. □

For the proof of Theorem 9.2.6 we need the following lemma (see e. g., [261]).

Lemma 9.2.7. *Let $(X, \leq)$ be a partially ordered set. If $\sup H$ exists for any subset $H \subseteq X$, then $(X, \leq)$ is a lattice.*

Proof. All we have to show is that for $x, y \in X$ there exists $\inf(x, y)$. Let $H := \{z \in X\ ;\ z \leq x \text{ and } z \leq y\}$. Then one can easily verify that $\sup H$ is the infimum of $\{x, y\}$. □

Corollary 9.2.8. *If G is a graph then there exists a maximum regular equivalence and there exists a minimum regular equivalence for G.*

Proof. The maximum is simply the supremum over all regular equivalences. Dually, the minimum is the infimum over all regular equivalences. Or easier: The minimum is the identity partition which is always regular and minimal. □

Although the supremum in the lattice of regular equivalences is a restriction of the supremum in the lattice of all equivalences, the infimum is not.

Proposition 9.2.9 ([82]). *The lattice of regular equivalences is not a sublattice of the lattice of all equivalences.*

Proof. We show that the infimum is not a restriction of the infimum in the lattice of all equivalences (which is simply intersection). Consider the graph in Figure 9.3 and the two regular partitions $\mathcal{P}_1 := \{\,\{A, C, E\}, \{B, D\}\,\}$ and $\mathcal{P}_2 := \{\,\{A, C\}, \{B, D, E\}\,\}$. The intersection of $\mathcal{P}_1$ and $\mathcal{P}_2$ is $\mathcal{P} = \{\,\{A, C\}, \{B, D\}, \{E\}\,\}$, which is not regular. □

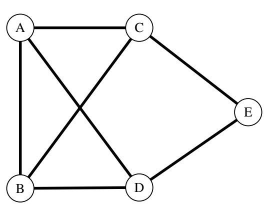

Fig. 9.3. Meet is not intersection

The fact that the supremum in the lattice of regular equivalences is a restriction of the supremum in the lattice of all equivalences implies the existence of a maximum regular equivalence which lies below a given (arbitrary) equivalence.

Definition 9.2.10. *Let G be a graph and $\sim$ an equivalence relation on its vertex set. An equivalence relation $\sim_1$ is called the* regular interior *of $\sim$ if it satisfies the following three conditions.*

1. *$\sim_1$ is regular,*
2. *$\sim_1 \leq \sim$, and*
3. *for all $\sim_2$ satisfying the above two conditions it holds $\sim_2 \leq \sim_1$.*

Corollary 9.2.11. *Let G be a graph and $\sim$ an equivalence relation on its vertex set. Then the regular interior of $\sim$ exists.*

On the other hand there is no minimum regular equivalence above a given equivalence in general (which would have been called a regular closure *or* regular hull*).*

Proof. For the first part, let $G = (V, E)$ be a graph and $\sim$ be an (arbitrary) equivalence on the node set. Then the supremum over the set of all regular equivalence relations that are finer than $\sim$ is the regular interior of $\sim$.

For the second part recall the example in the proof of Prop. 9.2.9 shown in Figure 9.3). It is easy to verify that the regular partitions $\mathcal{P}_1 := \{\{A, C, E\}, \{B, D\}\}$ and $\mathcal{P}_2 := \{\{A, C\}, \{B, D, E\}\}$ are both above the (non-regular) partition $\mathcal{P} := \{\{A, C\}, \{B, D\}, \{E\}\}$ and are both minimal with this property. $\square$

The regular interior is described in more detail in [90]; its computation is treated in Section 9.2.3. The infimum (in the lattice of regular equivalence relations) of two regular equivalence relations $\sim_1$ and $\sim_2$ is given by the regular interior of the intersection of $\sim_1$ and $\sim_2$.

9.2.3 Computation of Regular Interior

The regular interior (see Definition 9.2.10) of an equivalence relation $\sim$ is the coarsest regular refinement of $\sim$. It can be computed, starting with $\sim$, by a number of refinement steps in each of which currently equivalent vertices with non-equivalent neighborhoods are split, until all equivalent vertices have equivalent neighborhoods. For an example of such a computation see Figure 9.4. The running time of this computation depends heavily on how these refinement steps are organized.

In this section we present two algorithms for the computation of the regular interior. CATREGE [83] is the most well-known algorithm in the social network literature. It runs in time $\mathcal{O}(n^3)$. Tarjan and Paige [459] presented a sophisticated algorithm for the *relational coarsest partition problem*, which is essentially equivalent to computing the regular interior. Their algorithm runs in $\mathcal{O}(m \log n)$ time and is well-known in the bisimulation literature. See [408] for the relationship between bisimulation and regular equivalence.

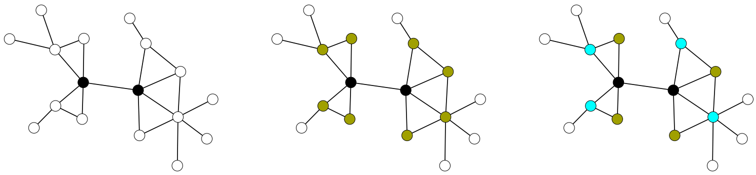

Fig. 9.4. Computation of the regular interior: initial partition (left), first step (middle) second and final step (right)

CATREGE. In [83], Borgatti and Everett proposed CATREGE as an algorithm for computing the maximal regular equivalence of a graph, or more generally for computing the regular interior of an equivalence relation. CATREGE runs in $\mathcal{O}(n^3)$. On a high-level view CATREGE proceeds as follows:

– CATREGE maintains in each refinement step a current partition $\mathcal{P}$, which is initially set to the complete partition (or alternatively to an arbitrary input partition).
– In each refinement step it tests, for each pair of equivalent vertices (w. r. t. $\mathcal{P}$), whether their neighborhoods are equivalent (w. r. t. $\mathcal{P}$). If so, then these vertices remain equivalent, otherwise they will be non-equivalent after this refinement step.
– The algorithm terminates if no changes happen.

The number of refinement steps is bounded by n, since in each refinement step (except the last) the number of equivalence classes grows by at least one. The running time of one refinement step is in $\mathcal{O}(n^2)$.

The Relational Coarsest Partition Problem. This section is taken from [459], although we translate the notation into the context of graphs.

Problem definition. The RELATIONAL COARSEST PARTITION PROBLEM (RCPP) has as input a (directed) graph $G = (V, E)$ and a partition $\mathcal{P}$ of the vertex set V.

For a subset $S \subseteq V$ we write $E(S) := \{v \in V\,;\ \exists u \in S \text{ such that } uEy\}$ and $E^{-1}(S) := \{u \in V\,;\ \exists v \in S \text{ such that } uEy\}$. For two subsets $B \subseteq V$ and $S \subseteq V$, B is called *stable* with respect to S if either $B \subseteq E^{-1}(S)$, or $B \cap E^{-1}(S) = \emptyset$. If $\mathcal{P}$ is a partition of V, $\mathcal{P}$ is called *stable* with respect to S if all of its blocks are stable with respect to S. $\mathcal{P}$ is called *stable* if it is stable with respect to each of its own blocks.

The RCPP is the problem of finding the coarsest stable refinement for the initial partition $\mathcal{P}$.

In the language of role assignments this condition means that for each two roles, say r_1 and r_2, either no vertex, or all vertices assigned r_1 has/have an out-going edge to a vertex assigned r_2. This is the 'out-part' in Definition 9.2.1.

The algorithm of Paige and Tarjan [459] runs in time $\mathcal{O}(m \log n)$ and space $\mathcal{O}(m + n)$. Especially for sparse graphs this is a significant improvement over CATREGE.

Paige and Tarjan already pointed out that it is possible to generalize their algorithm to handle a bounded number of relations. This generalization can be realized in such a way that it yields asymptotically the same running time (see e. g., [207]). Having done this one can apply the algorithm to compute the coarsest stable refinement with respect to E and E^{T} to obtain the regular interior (see Definition 9.2.10).

The Split *function.* The algorithm uses a primitive refinement operation. For each partition $\mathcal{Q}$ of V and subset $S \subseteq V$, let $\textsc{Split}(S, \mathcal{Q})$ be the refinement of $\mathcal{Q}$ obtained by replacing each block B of $\mathcal{Q}$ such that $B \cap E^{-1}(S) \neq \emptyset$ and $B \setminus E^{-1}(S) \neq \emptyset$ by the two blocks $B' := B \cap E^{-1}(S)$ and $B'' := B \setminus E^{-1}(S)$. We call S a *splitter* of $\mathcal{Q}$ if $\textsc{Split}(S, \mathcal{Q}) \neq \mathcal{Q}$. Note that $\mathcal{Q}$ is unstable with respect to S if and only if S is a splitter of $\mathcal{Q}$.

We note the following properties of Split and consequences of stability. Let S and Q be two subsets of V, and let $\mathcal{P}$ and $\mathcal{R}$ be two partitions of V. The following elementary properties are stated without proof.

Property 9.2.12.
1. Stability is *inherited* under refinement; that is, if $\mathcal{R}$ is a refinement of $\mathcal{P}$ and $\mathcal{P}$ is stable with respect to a set S, then so is $\mathcal{R}$.
2. Stability is *inherited* under union; that is, a partition that is stable with respect to two sets is also stable with respect to their union.
3. Function Split is *monotone* in its second argument; that is, if $\mathcal{P}$ is a refinement of $\mathcal{R}$ then $\textsc{Split}(S, \mathcal{P})$ is a refinement of $\textsc{Split}(S, \mathcal{R})$.
4. Function Split is *commutative* in the sense that the coarsest refinement of $\mathcal{P}$ stable with respect to both S and Q is

$$\textsc{Split}(S, \textsc{Split}(Q, \mathcal{P})) = \textsc{Split}(Q, \textsc{Split}(S, \mathcal{P})) \ .$$

Basic algorithm. We begin by describing a naive algorithm for the problem. The algorithm maintains a partition $\mathcal{Q}$ that is initially $\mathcal{P}$ and is refined until it is the coarsest stable refinement. The algorithm consists of repeating the following step until $\mathcal{Q}$ is stable:

> Refine: Find a set S that is a union of some of the blocks of $\mathcal{Q}$ and is a splitter of $\mathcal{Q}$; replace $\mathcal{Q}$ by $\textsc{Split}(S, \mathcal{Q})$.

Some observations. Since stability is inherited under refinement, a given set S can be used as a splitter in the algorithm only once. Since stability is inherited under the union of splitters, after sets are used as splitters their unions cannot be used as splitters. In particular, a stable partition is stable with respect to the union of any subset of its blocks.

Lemma 9.2.13. *The algorithm maintains the invariant that any stable refinement of $\mathcal{P}$ is also a refinement of the current partition $\mathcal{Q}$.*

Proof. By induction on the number of refinement steps. The lemma is true initially by definition. Suppose it is true before a refinement step that refines partition $\mathcal{Q}$ using a splitter S. Let $\mathcal{R}$ be any stable refinement of $\mathcal{P}$. Since S is a union of blocks of $\mathcal{Q}$ and $\mathcal{R}$ is a refinement of $\mathcal{Q}$ by the induction hypothesis, S is a union of blocks of $\mathcal{R}$. Hence $\mathcal{R}$ is stable with respect to S. Since SPLIT is monotone, $\mathcal{R} = \text{SPLIT}(S, \mathcal{R})$ is a refinement of $\text{SPLIT}(S, \mathcal{Q})$. □

The following theorem gives another proof for the existence of the regular interior (see Corollary 9.2.11).

Theorem 9.2.14. *The refinement algorithm is correct and terminates after at most $n-1$ steps, having computed the unique coarsest stable refinement.*

Proof. The assertion on the number of steps follows from the fact that the number of blocks is between 1 and n. Once no more refinement steps are possible, $\mathcal{Q}$ is stable, and by Lemma 9.2.13 any stable refinement is a refinement of $\mathcal{Q}$. It follows that $\mathcal{Q}$ is the unique coarsest stable refinement. □

The above algorithm is more general than is necessary to solve the problem: There is no need to use unions of blocks as splitters. Restricting splitters to blocks of $\mathcal{Q}$ will also suffice. However, the freedom to split using unions of blocks is one of the crucial ideas needed in developing a fast version of the algorithm.

Preprocessing. In an efficient implementation of the algorithm it it useful to reduce the problem instance to one in which $|E(\{v\})| \geq 1$ for all $v \in V$ (that is only to vertices having out-going edges). To do this we preprocess the partition $\mathcal{P}$ by splitting each block B into $B' := B \cap E^{-1}(V)$ and $B'' := B \setminus E^{-1}(V)$. The blocks B'' will never be split by the refinement algorithm; thus we can run the refinement algorithm on the partition $\mathcal{P}'$ consisting of the set of blocks B'. $\mathcal{P}'$ is a partition of the set $V' := E^{-1}(V)$, of size at most m. The coarsest stable refinement of $\mathcal{P}'$ together with the blocks B'' is the coarsest stable refinement of $\mathcal{P}$. The preprocessing and postprocessing take $\mathcal{O}(m+n)$ time if we have available the preimage set $E^{-1}(v)$ of each element $v \in V$. Henceforth, we shall assume $|E(\{v\})| \geq 1$ for all $v \in V$. This implies $m \geq n$.

Running time of the basic algorithm. We can implement the refinement algorithm to run in time $\mathcal{O}(mn)$ by storing for each element $v \in V$ its preimage set $E^{-1}(v)$. Finding a block of $\mathcal{Q}$ that is a splitter of $\mathcal{Q}$ and performing the appropriate splitting takes $\mathcal{O}(m)$ time. (Obtaining this bound is an easy exercise in list processing.) An $\mathcal{O}(mn)$ time bound for the entire algorithm follows.

Improved algorithm. To obtain a faster version of the algorithm, we need a good way to find splitters. In addition to the current partition $\mathcal{Q}$, we maintain another partition $\mathcal{X}$ such that $\mathcal{Q}$ is a refinement of $\mathcal{X}$ and $\mathcal{Q}$ is stable with respect to every block of $\mathcal{X}$ (in Section 9.3.4, $\mathcal{Q}$ will be called a *relative regular equivalence* w. r. t. $\mathcal{X}$). Initially $\mathcal{Q} = \mathcal{P}$ and $\mathcal{X}$ is the complete partition (containing V as its single block). The improved algorithm consists of repeating the following step until $\mathcal{Q} = \mathcal{X}$:

> Refine: Find a block $S \in \mathcal{X}$ that is not a block of $\mathcal{Q}$. Find a block $B \in \mathcal{Q}$ such that $B \subseteq S$ and $|B| \leq |S|/2$. Replace S within $\mathcal{X}$ by the two sets B and $S \setminus B$; replace $\mathcal{Q}$ by $\textsc{Split}(S \setminus B, \textsc{Split}(B, \mathcal{Q}))$.

The correctness of this improved algorithm follows from the correctness of the original algorithm and from the two ways given previously in which a partition can inherit stability with respect to a set.

Special case: If E is a function. Before discussing this algorithm in general, let us consider the special case in which E is a function, i.e., $|E(\{v\})| = 1$ for all $v \in V$. In this case, assume that $\mathcal{Q}$ is a partition stable with respect to a set S that is a union of some of the blocks of $\mathcal{Q}$, and $B \subseteq S$ is a block of $\mathcal{Q}$. Then $\textsc{Split}(B, \mathcal{Q})$ is stable with respect to $S \setminus B$ as well. This holds, since if B_1 is a block of $\textsc{Split}(B, \mathcal{Q})$, $B_1 \subseteq E^{-1}(B)$ implies $B_1 \cap E^{-1}(S \setminus B) = \emptyset$, and $B_1 \subseteq E^{-1}(S) \setminus E^{-1}(B)$ implies $B_1 \subseteq E^{-1}(S \setminus B)$. It follows that in each refinement step it suffices to replace $\mathcal{Q}$ by $\textsc{Split}(B, \mathcal{Q})$, since $\textsc{Split}(B, \mathcal{Q}) = \textsc{Split}(S \setminus B, \textsc{Split}(B, \mathcal{Q}))$. This is the idea underlying Hopcroft's 'process the smaller half' algorithm for the functional coarsest partition problem. The refining set B is at most half the size of the stable set S containing it.

Back to the general case. In the more general relational coarsest partition problem, stability with respect to both S and B does *not* imply stability with respect to $S \setminus B$, and Hopcroft's algorithm cannot be used. This is a serious problem since we cannot afford (in terms of running time) to scan the set $S \setminus B$ in order to perform one refinement step. Nevertheless, we are still able to exploit this idea by refining with respect to both B and $S \setminus B$ using a method that explicitly scans only B.

A preliminary lemma. Consider a general step in the improved refinement algorithm.

Lemma 9.2.15. *Suppose that partition $\mathcal{Q}$ is stable with respect to a set S that is a union of some of the blocks of $\mathcal{Q}$. Suppose also that partition $\mathcal{Q}$ is refined first with respect to a block $B \subseteq S$ and then with respect to $S \setminus B$. Then the following conditions hold:*

1. *Refining $\mathcal{Q}$ with respect to B splits a block $D \in \mathcal{Q}$ into two blocks $D_1 = D \cap E^{-1}(B)$ and $D_2 = D - D_1$ iff $D \cap E^{-1}(B) \neq \emptyset$ and $D \setminus E^{-1}(B) \neq \emptyset$.*
2. *Refining $\textsc{Split}(B, \mathcal{Q})$ with respect to $S \setminus B$ splits D_1 into two blocks $D_{11} = D_1 \cap E^{-1}(S \setminus B)$ and $D_{12} = D_1 - D_{11}$ iff $D_1 \cap E^{-1}(S \setminus B) \neq \emptyset$ and $D_1 \setminus E^{-1}(S \setminus B) \neq \emptyset$.*
3. *Refining $\textsc{Split}(B, \mathcal{Q})$ with respect to $S \setminus B$ does not split D_2.*
4. *$D_{12} = D_1 \cap (E^{-1}(B) \setminus E^{-1}(S \setminus B))$.*

Proof. Conditions 1 and 2 follow from the definition of Split.

Condition 3: Form Condition 1 it follows that if D is split, it is $D \cap E^{-1}(B) \neq \emptyset$. Since D is stable with respect to S, and since $B \subseteq S$, then $D_2 \subseteq D \subseteq E^{-1}(S)$. Since by Cond. 1 $D_2 \cap E^{-1}(B) = \emptyset$, it follows that $D_2 \subseteq E^{-1}(S \setminus B)$.

Condition 4: This follows from the fact that $D_1 \subseteq E^{-1}(B)$ and $D_{12} = D_1 \setminus E^{-1}(S \setminus B)$. □

Performing the three-way splitting of a block D into D_{11}, D_{12}, and D_2 as described in Lemma 9.2.15 is the hard part of the algorithm. Identity 4 of Lemma 9.2.15 is the crucial observation that we shall use in our implementation. Remember that scanning the set $S \setminus B$ takes (possibly) too long to obtain the claimed running time. We shall need an additional datastructure to determine $D_1 \setminus E^{-1}(S \setminus B) = (D \cap E^{-1}(B)) \setminus E^{-1}(S \setminus B)$ by scanning only B.

Running time of the improved algorithm. A given element of V is in at most $\log_2 n+1$ different blocks B used as refining sets, since each successive such set is at most half the size of the previous one. We shall describe an implementation of the algorithm in which a refinement step with respect to block B takes $\mathcal{O}(|B| + \sum_{u\in B} |E^{-1}(\{u\})|)$ time. From this an $\mathcal{O}(m \log n)$ overall time bound for the algorithm follows by summing over all blocks B used for refinement and over all elements in such blocks.

Datastructures. (See Section 9.1.3 for an example of a much simpler algorithm which already uses some of the ideas of this algorithm.)

Graph $G = (V, E)$ is represented by the sets V and E. Partitions $\mathcal{Q}$ and $\mathcal{X}$ are represented by doubly linked lists of their blocks.

A block S of $\mathcal{X}$ is called *simple* if it contains only a single block of $\mathcal{Q}$ (equal to S but indicated by its own record) and *compound* if it contains two or more blocks of $\mathcal{Q}$.

The various records are linked together in the following ways. Each edge uEv points its source u. Each vertex v points to a list of incoming edges uEv. This allows scanning the set $E^{-1}(\{v\})$ in time proportional to its size. Each block of $\mathcal{Q}$ has an associated integer giving its size and points to a doubly linked list of the vertices in it (allowing deletion in $\mathcal{O}(1)$ time). Each vertex points to the block of $\mathcal{Q}$ containing it. Each block of $\mathcal{X}$ points to a doubly linked list of the blocks of $\mathcal{Q}$ contained in it. Each block of $\mathcal{Q}$ points to the block of $\mathcal{X}$ containing it. We also maintain a set C of compound blocks of $\mathcal{X}$. Initially C contains the single block V, which is the union of the blocks of $\mathcal{P}$. If $\mathcal{P}$ contains only one block (after the preprocessing), $\mathcal{P}$ itself is the coarsest stable refinement and we terminate the algorithm here.

To make three-way splitting (see Lemma 9.2.15) fast we need one more collection of records. For each block S of $\mathcal{X}$ and each element $v \in E^{-1}(S)$ we maintain an integer $\textsc{Count}(v, S) := |S \cap E(\{v\})|$. Each edge uEv with $v \in S$ contains a pointer to $\textsc{Count}(u, S)$. Initially there is one count per vertex (i. e., $\textsc{Count}(v, V) = |E(\{v\})|$) and each edge uEv points to $\textsc{Count}(u, V)$.

This $\textsc{Count}$ function will help to determine the set $E^{-1}(B) \setminus E^{-1}(S \setminus B)$ in time proportional to $|\{uEv\,;\ v \in B\}|$ (see step 5 below).

Both the space needed for all the data structures and the initialization time is $\mathcal{O}(m)$.

The refinement algorithm consists of repeating refinement steps until C is empty.

Performing one refinement step. For clarity we divide one refinement step into 7 substeps.

1. (select a refining block). Remove some block S from C. (Block S is a compound block of $\mathcal{X}$.) Examine the first two blocks in the list of blocks of $\mathcal{Q}$ contained in S. Let B be the smaller one. (Break a tie arbitrarily.)

2. (update $\mathcal{X}$). Remove B from S and create a new (simple) block S' of $\mathcal{X}$ containing B as its only block of $\mathcal{Q}$. If S is still compound, put S back into C.

3. (compute $E^{-1}(B)$). Copy the vertices of B into a temporary set B'. (This facilitates splitting B with respect to itself during the refinement.) Compute $E^{-1}(B)$ by scanning the edges uEv such that $v \in B$ and adding each vertex u in such an edge to $E^{-1}(B)$ if it has not already been added. Duplicates are suppressed by marking vertices as they are encountered and linking them together for later unmarking. During the same scan compute $\textsc{Count}(u, B) = |\{v \in B\,;\ uEv\}|$, store this count in a new integer and make u point to it. These counts will be used in step 5.

4. (refine $\mathcal{Q}$ with respect to B). For each block D of $\mathcal{Q}$ containing some element (vertex) of $E^{-1}(B)$, split D into $D_1 = D \cap E^{-1}(B)$ and $D_2 = D \setminus D_1$. Do this by scanning the elements of $E^{-1}(B)$. To process an element $u \in E^{-1}(B)$, determine the block D of $\mathcal{Q}$ containing it and create an associated block D' if one does not already exist. Move u from D to D'.

During the scanning, construct a list of those blocks D that are split. After the scanning, process the list of split blocks. For each such block D with associated block D', mark D' as no longer being associated with D (so that it will be correctly processed in subsequent iterations of Step 4). Eliminate the record for D if D is now empty and, if D is nonempty and the block of $\mathcal{X}$ containing D and D' has been made compound by the split, add this block to C.

5. (compute $E^{-1}(B)\setminus E^{-1}(S\setminus B)$). Scan the edges uEv with $v \in B'$. To process an edge uEv, determine $\textsc{Count}(u, B)$ (to which u points) and $\textsc{Count}(u, S)$ (to which uEv points). If $\textsc{Count}(u, B) = \textsc{Count}(u, S)$, add u to $E^{-1}(B) \setminus E^{-1}(S \setminus B)$ if it has not been added already.

6. (refine $\mathcal{Q}$ with respect to $S \setminus B$). Proceed exactly as in Step 4 but scan $E^{-1}(B) \setminus E^{-1}(S \setminus B)$ (computed in Step 5) instead of $E^{-1}(B)$.

7. (update counts). Scan the edges uEv such that $v \in B'$. To process and edge uEv, decrement $\textsc{Count}(u, S)$ (to which uEv points). If this count becomes zero, delete the $\textsc{Count}$ record, and make uEv point to $\textsc{Count}(u, B)$ (to which u points). After scanning all the appropriate edges, discard B'.

Note that in step 5 only edges terminating in B' are scanned. Step 5 is correct (computes $E^{-1}(B) \setminus E^{-1}(S \setminus B)$) since for each vertex u in $E^{-1}(B)$, it holds that u is in $E^{-1}(B) \setminus E^{-1}(S \setminus B)$ iff u is not in $E^{-1}(S \setminus B)$ iff all edges starting at u and terminating in S terminate in B iff $\textsc{Count}(u, B) = \textsc{Count}(u, S)$.

The correctness of this implementation follows in a straightforward way from our discussion above of three-way splitting. The time spent in a refinement step is $\mathcal{O}(1)$ per edge terminating in B plus $\mathcal{O}(1)$ per vertex of B, for a total of $\mathcal{O}(|B| + \sum_{v \in B} |E^{-1}(\{v\})|)$ time. An $\mathcal{O}(m \log n)$ time bound for the entire algorithm follows as discussed above. It is possible to improve the efficiency of the algorithm by a constant factor by combining various steps, which have been kept separate for clarity.

Adaptation to Related Problems. The above algorithm turns out to be the key to efficiently solve several partition refinement problems that arise in this chapter. We will briefly sketch this generality.

Computing the maximal strong structural equivalence (as described in Section 9.1.3) or the relative regular equivalence (see Section 9.3.4) is much simpler than computing the regular interior. Nevertheless we can use the idea of iteratively splitting blocks according to intersection with certain neighborhoods. (See algorithm 21 and the comments in Section 9.3.4.) These problems can be solved by algorithms that run in $\mathcal{O}(m + n)$.

Computing the coarsest *equitable* (see Section 9.3.1) has been solved earlier than the problem of computing the regular interior (see [110] for an $\mathcal{O}(m \log n)$ algorithm and the comments in [459]).

Refining a partition w.r.t. *multiple relations* (see Definition 9.4.1) is also possible in $\mathcal{O}(m \log n)$ (if the number of relations is bounded by a constant). This extension of the algorithm can be used to compute the regular interior w.r.t. incoming and out-going edges. Shortly, a partition can be refined w.r.t. multiple relations by performing steps 3–7 (see above) for fixed B and S successively for all relations, one at a time. (See e.g., [207].)

9.2.4 The Role Assignment Problem

In this section we investigate the computational complexity of the decision problem whether a given graph admits a regular role assignment with prespecified role graph, or with prespecified number of equivalence classes. In this section we consider only undirected graphs.

The most complete characterization is from Fiala and Paulusma [209]. Let $k \in \mathbb{N}$ and R be an undirected graph, possibly with loops.

Problem 9.2.16 (k-Role Assignment (k-RA)). Given a graph G.
Question: Is there a regular equivalence for G with exactly k equivalence classes?

Problem 9.2.17 (R-Role Assignment (R-RA)). Given a graph G.
Question: Is there a regular role assignment $r : V(G) \to V(R)$ with role graph R?

Note that we require role assignments to be surjective mappings.

Theorem 9.2.18 ([209]). *k-RA is polynomially solvable for $k = 1$ and it is $\mathcal{NP}$-complete for all $k \geq 2$.*

Theorem 9.2.19 ([209]). *R-RA is polynomially solvable if each component of R consists of a single vertex (with or without a loop), or consists of two vertices without loops and it is $\mathcal{NP}$-complete otherwise.*

We give the proof of one special case of the R-Role Assignment Problem.

Theorem 9.2.20 ([493]). *Let R_0 be the graph in Figure 9.5. Then R_0-RA is $\mathcal{NP}$-complete.*

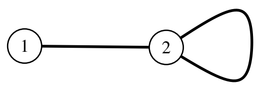

Fig. 9.5. Role graph R_0

Proof. It is easy to see that R-RA is in $\mathcal{NP}$ since one can easily check in polynomial time whether a given function $r : V \to \{1, 2\}$ is a 2-role assignment with role graph R_5.

We will show that the 3-satisfiability problem (3SAT) is polynomially transformable to R_0-RA. So let $U = \{u_1, \ldots, u_n\}$ be a set of variables and $C = \{c_1, \ldots, c_m\}$ be a set of clauses (each consisting of exactly three literals). We will construct a graph $G = (V, E)$ such that G is 2-role assignable with role graph R_0 if and only if C is satisfiable.

The construction will be made up of two components, truth-setting components and satisfaction testing components (see Figure 9.6).

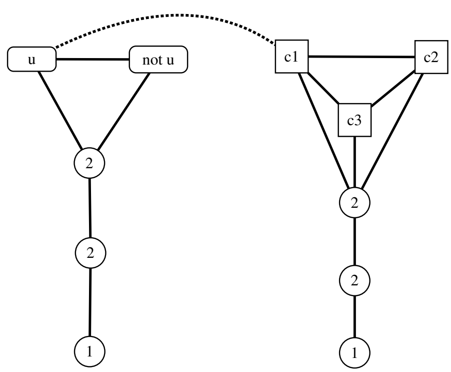

Fig. 9.6. Truth-setting component for variable u (left); satisfaction testing component for clause $\{c_1, c_2, c_3\}$ (right) and communication edge if literal c_1 equals u (dashed). The roles of the vertices in the pending paths are uniquely determined (as indicated by the labels 1 resp. 2) if the role assignment should be regular with role graph R_0

For each variable $u_i \in U$, there is a truth-setting component $T_i = (V_i, E_i)$ with

$$V_i := \{u_i, \overline{u_i}, a_{i1}, a_{i2}, a_{i3}\} \ ,$$
$$E_i := \{u_i\overline{u_i}, u_i a_{i3}, \overline{u_i} a_{i3}, a_{i1}a_{i2}, a_{i2}a_{i3}\} \ .$$

Note that, although we write $u_i\overline{u_i}$ for the edge $\{u_i, \overline{u_i}\}$, the graph is undirected.

The intuition behind the construction of T_i is the following: If a graph containing T_i as a subgraph (such that the a_{ij} are adjacent only to the vertices in V_i as specified above) admits a regular role assignment r with role graph R_0, then necessarily $r(a_{i1}) = 1$, since a_{i1} has degree one and a vertex which is assigned 2 must have degree ≥ 2. Then $r(a_{i2}) = 2$, since a 1-vertex is adjacent to a 2-vertex and $r(a_{i2}) = 2$, since a 2-vertex is adjacent to a 2-vertex. Finally exactly one of u_i or $\overline{u_i}$ is assigned 2, meaning that variable u_i is set to *true* or *false*, respectively. Thus component T_i ensures that a variable gets either *true* or *false*.

For each clause $c_j \in C$, let vertices c_{j1}, c_{j2}, and c_{j3} be three vertices corresponding to the three literals in the clause c_j. Then there is a satisfaction testing component $S_j = (V_j', E_j')$ with

$$V_j' := \{c_{j1}, c_{j2}, c_{j3}, b_{j1}, b_{j2}, b_{j3}\} \ ,$$
$$E_j' := \{c_{j1}c_{j2}, c_{j1}c_{j3}, c_{j2}c_{j3}, c_{j1}b_{j3}, c_{j2}b_{j3}, c_{j3}b_{j3}, b_{j1}b_{j2}, b_{j2}b_{j3}\} \ .$$

The intuition behind the construction of S_j is the following: If a graph containing S_j as a subgraph (such that the b_{jl} are adjacent only to the vertices in V_j as specified above) admits a regular role assignment r with role graph R_0, then necessarily $r(b_{j1}) = 1$, $r(b_{j2}) = r(b_{j3}) = 2$, which ensures that one of the vertices c_{j1}, c_{j2}, c_{j3} is assigned 1, thus ensuring that every adjacent vertex of this 1-vertex must be assigned 2. This will be crucial later.

The construction so far is only dependent on the number of variables and clauses. The only part of the construction that depends on which literals occur in which clauses is the collection of communication edges. For each clause $c_j = \{x_{j1}, x_{j2}, x_{j3}\} \in C$ the communication edges emanating from S_j are given by

$$E_j'' := \{c_{j1}x_{j1}, c_{j2}x_{j2}, c_{j3}x_{j3}\} \ .$$

(The x_{jl} are either variables in U or their negations.) Notice that for each c_{jk}, there is exactly one vertex that is adjacent to c_{jk} in E_j'', which is the corresponding literal vertex for c_{jk} in the clause c_j.

To complete the construction of our instance of R_0-RA, let $G = (V, E)$ with V being the union of all V_is and all V_j's and E the union of all E_is, all E_j's and all E_j''s.

As mentioned above, given a regular role assignment for G with role graph R_0, for each $j = 1, \ldots, m$ there is a vertex c_{jk} such that $r(c_{jk}) = 1$ implying that the corresponding adjacent literal is assigned 2. Setting this literal to *true* will satisfy clause c_j.

Thus we have shown that the formula is satisfiable if G is regularly R_0 assignable.

Conversely, suppose that C has a satisfying truth assignment. We obtain an assignment $r\colon V \to \{1, 2\}$ as follows. For each $i = 1, \ldots, n$ set $r(u_i)$ to 2 (and

$r(\overline{u_i})$ to 1) if and only if variable u_i is *true* and set the role of the vertices a_{ik} and b_{jk} as implied by the fact that r should be regular (see above). Moreover, for each $j = 1, \ldots, m$ let c_{jk}, $k \in \{1, 2, 3\}$, be some vertex whose corresponding literal in the clause c_j is *true* – such a k exists since the truth assignment is satisfying for C. Set $r(c_{jk}) := 1$ and $r(c_{jl}) := 2$ for $l \in \{1, 2, 3\}$, $l \neq k$.

The proof is complicated a bit by the fact that more than one literal in a clause might be true, but setting $r(c_{jk}) = 1$ is allowed for only one $k \in \{1, 2, 3\}$. Since a 2-vertex may be adjacent to another 2-vertex, this does not destroy the regularity of r. □

9.2.5 Existence of *k*-Role Assignments

We have seen in the previous section that the decision whether a graph admits a regular equivalence with exactly k equivalence classes is $\mathcal{NP}$-complete for general graphs. Nevertheless, there are easy-to-verify sufficient, if not necessary, conditions that guarantee the existence of regular k-role assignments. Briefly, the condition is that the graph differs not too much from a regular graph.

Theorem 9.2.21 ([474]). *For all $k \in \mathbb{N}$ there is a constant $c_k \in \mathbb{R}$ such that for all graphs G with minimal degree $\delta = \delta(G)$ and maximal degree $\Delta = \Delta(G)$ satisfying*

$$\delta \geq c_k \log(\Delta) \ ,$$

there is a regular equivalence for G with exactly k equivalence classes.

To exclude trivial counterexamples we assume in the following that all graphs in question have at least k vertices.

For the proof we need a uniform version of the LOVASZ LOCAL LEMMA.

Theorem 9.2.22 ([25, Chapter 5 Corollary1.2]). *Let $A_i, i \in I$, be events in a discrete probability space. If there exists M such that for every $i \in I$*

$$|\{A_j\,;\ A_j \text{ is not independent of } A_i\}| \leq M \ ,$$

and if there exists $p > 0$ such that $\Pr(A_i) \leq p$ for every $i \in I$, then

$$ep(M+1) \leq 1 \Longrightarrow \Pr\left(\bigcap_{i\in I} \overline{A_i}\right) > 0 \ ,$$

where e is the EULER *number $e = \sum_{i=0}^{\infty} 1/i\,!$.* □

Proof (of Theorem 9.2.21). Define $r : V \to \{1, \ldots, k\}$ as follows: For every $v \in V$ choose $r(v)$ uniformly at random from $\{1, \ldots, k\}$.

For $v \in V$, let A_v be the event that $r(N(v)) \neq \{1, \ldots, k\}$. It is

$$\Pr(A_v) \leq k\left(\frac{k-1}{k}\right)^{d(v)} \leq k\left(\frac{k-1}{k}\right)^{\delta(G)} \ .$$

Because all $r(w)$ are chosen independently and for a fixed value i, the probability that i is not used for any of the vertices adjacent to v is $\left(\frac{k-1}{k}\right)^{d(v)}$, and there are k choices for i.

Also note that A_v and A_w are not independent if and only if $N(v) \cap N(w) \neq \emptyset$. Hence, A_v with $M := \Delta(G)^2$ and $p := k\left(\frac{k-1}{k}\right)^{\delta(G)}$ satisfies the conditions of the Lovasz Local Lemma. Therefore,

$$ek\left(\frac{k-1}{k}\right)^{\delta(G)}(\Delta(G)^2+1) \leq 1 \Rightarrow \Pr\left(\bigcap_{v \in V} \overline{A_v}\right) > 0 \ . \tag{9.1}$$

If the righthand side of (9.1) holds, there exists at least one r such that $r(N(v)) = \{1, \ldots, k\}$ for every $v \in V$, that is, there exists at least one regular k-role assignment. In order to finish the proof we note that the lefthand side of (9.1) is equivalent to

$$\delta(G) \geq \frac{\log(ek(\Delta(G)^2+1))}{\log\left(\frac{k}{k-1}\right)} \ .$$

Clearly, there exists a constant c_k such that $c_k \log(\Delta(G))$ is greater than the righthand side of the above inequality. □

Conclusion. Regular equivalences are well investigated in computer science. Results indicate that many regular equivalences exist even in irregular graphs, but it is unclear how to define and/or compute the best, or at least a good one. Fast algorithms exist for the computation of the maximal regular equivalence or for the regular interior of an a priori partition. The maximal regular equivalence could be meaningful for directed graphs (for undirected it is simply the division into isolates and non-isolates). Also, the regular interior could be a good role assignment if one has an idea for the partition to be refined. Specifying the number of equivalence classes or the role graph yields $\mathcal{NP}$-hard problems, in the general case. Optimization approaches for these problems are presented in Section 10.1.7 in the next chapter.

9.3 Other Equivalences

In this section we briefly mention other (than structural or regular) types of role equivalences.

9.3.1 Exact Role Assignments

In this section we define a class of equivalence relations that is a subset of regular equivalences. These equivalences will be called *exact*. The associated partitions are also known as *equitable partitions* in graph theory, they have first been defined as *divisors* of graphs.

While for regular equivalences only the occurrence or non-occurrence of a role in the neighborhood of a vertex matters, for exact equivalences, the number of occurrence matters.

The graph model of this section are undirected multigraphs.

Definition 9.3.1. *A role assignment r is called* exact *if for all $u, v \in V$*

$$r(u) = r(v) \quad \Longrightarrow \quad r(N(u)) = r(N(v)) \ ,$$

where the last equation is an equation of multi-sets*, i. e., vertices, that have the same role, must have the same number of each of the other roles in their neighborhoods.*

The coloring in Figure 9.7 defines an exact role assignment for the shown graph.

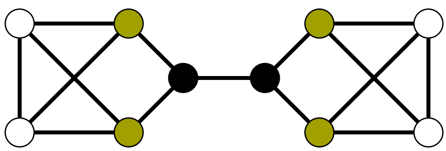

Fig. 9.7. An exact role assignment

While an equivalence is regular for a multigraph if and only if it is regular for the induced simple graph (each edge at most once), for exact equivalences the multiplicity of an edge matters.

It is straightforward to see that exact role assignments are regular, the converse is not true.

An equivalent definition is the following.

Definition 9.3.2 ([247]). *A partition $\mathcal{P} = \{C_1, \ldots, C_k\}$ of the vertex set V of an undirected (multi-)graph $G = (V, E)$ is called* equitable *if there are integers b_{ij}, $i, j = 1, \ldots, k$, such that each vertex in class C_i has exactly b_{ij} neighbors in class C_j. The matrix $B = (b_{ij})_{i,j=1,\ldots,k}$ defines a (directed) multi-graph, which is called the* quotient *of G modulo $\mathcal{P}$, denoted by $G/\mathcal{P}$.*

A partition is equitable if and only if the associated role assignment is exact. The above definition also extends the definition of the quotient or role graph (see Section 9.0.2) to multigraphs. Note that this is possible only for exact role assignments.

Note that even if the graph is undirected the quotient is possibly directed, meaning that the multiplicity of an edge may differ from the multiplicity of the reversed edge. This happens always if two 'adjacent' equivalence classes are of different size.

Exact role assignments are compatible with algebraic properties of a graph.

Theorem 9.3.3 ([247]). *Let G be a graph, $\mathcal{P}$ an equitable partition. Then, the characteristic polynomial of the quotient $G/\mathcal{P}$ divides the characteristic polynomial of G.* □

This theorem implies that the spectrum of the quotient $G/\mathcal{P}$ is a subset of the spectrum of G.

The set of all exact role assignments of a graph forms a lattice [191]. The maximal exact role assignment of a graph can be computed by an adaption of the algorithm in Section 9.2.3. (See [110] and the comments in [459].)

Many problems around exact role assignments are $\mathcal{NP}$-complete as well. For example the problem of deciding if a graph G admits an exact role assignment with quotient R is $\mathcal{NP}$-complete if both G and R are part of the input, or for some fixed R. This holds, since the $\mathcal{NP}$-complete problem of deciding whether a 3-regular graph has a perfect code [370], can be formulated as the problem of deciding whether G has an exact role assignment with quotient

$$R = \begin{bmatrix} 0 & 3 \\ 1 & 2 \end{bmatrix} .$$

The quotient over an equitable partition has much more in common with the original graph than, e. g., the role graph over a regular equivalence. Exact role assignments also ensure that equivalent vertices have the same degree, which is not true for regular role assignments.

Conclusion. Exact role assignments, also called equitable partitions are well investigated in algebraic graph theory. While some problems around equitable partitions are $\mathcal{NP}$-complete, there are efficient algorithms to compute the maximal equitable partition of a graph, or to compute the coarsest equitable refinement of an a priori partition. These algorithms could be used to compute role assignments, but, due to irregularities, the results contain in most cases too many classes and miss the underlying (possibly perturbed) structure. Brandes and Lerner [97] introduced a relaxation of equitable partitions that is tolerant against irregularities.

9.3.2 Automorphic and Orbit Equivalence

Automorphic equivalence expresses interchangeability of vertices.

Definition 9.3.4 ([191]). *Let $G = (V, E)$ be a graph, $u, v \in V$. Then u and v are said to be* automorphically equivalent *if there is an automorphism φ of G with $\varphi(u) = v$.*

Automorphically equivalent vertices cannot be distinguished only in terms of the graph structure. Therefore it could be argued that at least automorphically equivalent vertices should be considered to play the same role.

It is easy to see that structurally equivalent vertices are automorphically equivalent.

A partition of the vertex set which has the property that each pair of equivalent vertices is automorphically equivalent is not necessarily a regular equivalence. However we have the following result.

Proposition 9.3.5 ([190]). *Let $G = (V, E)$ be a graph with automorphism group $A(G)$, and $H < A(G)$ be a subgroup of $A(G)$. Then assigning roles according to the orbits of H defines an exact role assignment for G. Such a partition is called an* orbit partition.

Proof. Let r be a role assignment as in the formulation of the proposition. If $r(u) = r(v)$ then there exists $\varphi \in H$ such that $\varphi(u) = v$. If $x \in N^+(u)$, then $\varphi(x) \in N^+(\varphi(u)) = N^+(v)$. Furthermore $r(x) = r(\varphi(x))$ by definition. It follows that $r(N^+(u)) \subseteq r(N^+(v))$ (as multisets). The other inclusion and the corresponding assertion for the in-neighborhoods is shown similar. $\square$

In particular, orbit equivalences are regular.

For example, the coloring in Figure 9.7 defines the orbit partition of the automorphism group of the shown graph.

The set of orbit equivalences forms a proper subset of the set of all exact equivalences, which can be proved by any regular graph which is not vertex-transitive. For example, the complete partition for the graph in Figure 9.7 is exact but not an orbit partition.

The above proposition can also be used to prove that every undirected role primitive graph (see Section9.2.1) is a graph with trivial automorphism group [190]. This is not true for directed graphs as can be seen by directed cycles of prime length.

Orbit equivalence has the nice feature that its condition is invariant w. r. t. a shift to the complement graph. This does not hold neither for regular nor for exact equivalence.

The computation of orbit equivalences is related to the problem of computing the automorphism group which has open complexity status.

Conclusion. Automorphically equivalent vertices cannot be distinguished in terms of graph structure, but only by additional labels or attributes. It could therefore be argued that at least automorphically equivalent vertices play the same role. Computation of automorphic equivalence seems to be hard, but, in irregular networks, there won't be any significant automorphisms anyway.

9.3.3 Perfect Equivalence

Perfect equivalence is a restriction of regular equivalence. It expresses the idea that there must be a reason for two vertices for being *not* equivalent.

Definition 9.3.6 ([191]). *A role assignment r defines a* perfect equivalence *if for all $u, v \in V$*

$$r(u) = r(v) \quad \Longleftrightarrow \quad r(N^+(u)) = r(N^+(v)) \text{ and } r(N^-(u)) = r(N^-(v)).$$

A regular equivalence is perfect if and only if the induced role graph has no strong structural equivalent vertices (see Section 9.1).

The set of perfect equivalence relations of a graph is a lattice [191], which is neither a sublattice of all equivalence relations (Section 9.1.1) nor of the lattice of regular equivalence relations (Section 9.2.2). A *perfect interior* of an equivalence relation $\sim$ would be a coarsest perfect refinement of $\sim$ (compare Definition 9.2.10). In contrast to the regular interior, the perfect interior does not exist in general.

Theorem 9.3.7. *In general, the transitive closure (see Section 9.1.1) of the union of two perfect equivalence relations is not perfect. In particular, for some equivalences there is no perfect interior.*

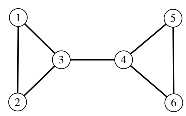

Fig. 9.8. Graph for the proof of Theorem 9.3.7. Supremum of two perfect equivalences is not perfect

Proof. Consider the graph in Figure 9.8 and the two perfect partitions $\mathcal{P}_1 = \{\{1,5\},\{2,6\}\{3,4\}\}$ and $\mathcal{P}_2 = \{\{1,2\},\{5,6\}\{3\},\{4\}\}$. The transitive closure of $\mathcal{P}_1$ and $\mathcal{P}_2$ is $\mathcal{P} = \{\{1,2,5,6\},\{3,4\}\}$, which is not perfect.

For the second statement, note that $\mathcal{P}_1$ and $\mathcal{P}_2$ are both perfect refinements of $\mathcal{P}$ and are both maximal w. r. t. this property. □

The second statement has a more trivial proof: For a graph with two strong structurally equivalent vertices, the identity partition has no perfect refinement.

Some decision problems concerning perfect equivalence are $\mathcal{NP}$-complete as well. This can be seen by Theorems 9.2.18 and 9.2.19, restricted to role graphs without strong structurally equivalent vertices.

Although perfect equivalences rule out some trivial regular equivalences, there is no evidence why roles shouldn't be strong structurally equivalent.

Conclusion. Perfect equivalence is a restriction of regular equivalence, but it doesn't seem to yield better role assignments. Some mathematical properties of regular equivalences get lost and there are examples where the condition on perfect equivalence rules out good regular role assignments.

9.3.4 Relative Regular Equivalence

Relative regular equivalence expresses the idea that equivalent vertices have equivalent neighborhoods in a coarser, predefined measure.

Definition 9.3.8 ([90]). *Let* $G = (V, E)$ *be a graph and* $r\colon V \to W$ *and* $r_0\colon V \to W_0$ *be two role assignments. Then,* r *is called* regular relative *to* r_0 *if* $r \leq r_0$ *(see Section 9.1.1 for the partial order on the set of role assignments) and for all* $u, v \in V$

$$r(u) = r(v) \Rightarrow r_0(N^+(u)) = r_0(N^+(v)) \text{ and } r_0(N^-(u)) = r_0(N^-(v)) \ .$$

A typical application [90] of relative regular equivalence is given by a network of symmetric friendship ties which a priori is divided into two disjoint friendship cliques A and B. Assume that within each clique every member has at least one tie to some other member of the same clique. The partition into these two cliques would be regular if either there is no tie between the two cliques or each actor would have, in addition to the intra-group ties, at least one tie to a member of the other group. But lets assume that some, but not all, actors have friendship ties to members of the other group. The partition into A and B is no longer regular. Now we can split each group into those actors having ties to some member of the other group and those who don't. Say we obtain the partition into A_1, A_2, B_1, and B_2. Neither is this partition (in general) regular: There might be some actors in, say, A_1 having intra-group ties only with members of A_1, some only with members of A_2, some with both; they don't have equivalent neighborhoods. But they have equivalent neighborhoods with respect to the coarse partition into A and B. Thus, the partition into A_1, A_2, B_1 and B_2 is regular relative to the partition into A and B.

Relative regularity below a fixed equivalence is preserved under refinement. (Compare Prop. 9.1.7 for a similar proposition for structural equivalence.)

Proposition 9.3.9. *Let* $\sim$, $\sim_1$, *and* $\sim_2$ *be equivalence relations on* V *such that* $\sim_1 \leq \sim_2$ *and* $\sim_2$ *is regular relative to* $\sim$. *Then so is* $\sim_1$.

Similar to Prop. 9.1.7, this proposition implies that the set of equivalences that are regular relative to a fixed equivalence $\sim$ is a sublattice of all equivalences and is completely described by the maximum of this set, denoted here by MRRE($\sim$).

Computing the MRRE($\sim$) is possible in linear time by an adaptation of the algorithm 21 for computing the maximal structural equivalence: Instead of splitting equivalence classes from the point of view of single vertices, classes are split from the point of view of the classes of $\sim$ (compare the algorithm in Section 9.2.3). Note that the classes of $\sim$ are fixed and the MRRE($\sim$) has been found after all classes of $\sim$ have been processed once.

Each refinement step in the CATREGE algorithm (see Section 9.2.3) computes an equivalence that is regular relative to the previous one, but the running time of one step is in $\mathcal{O}(n^2)$, which is worse than the above described algorithm on sparse graphs.

Conclusion. Relative regular equivalence is computationally simple but it needs an a priori partition of the vertices and, since its compatibility requirement is only local, is not expected to represent global network structure. It has most been applied in connection with multiple and composite relations (see, e. g., Winship-Pattison Role Equivalence in Section 9.5.1).

9.4 Graphs with Multiple Relations

Actors in a social network are often connected by more than one relation. For example, on the set of employees of a company there might be two relations GivesOrdersTo and IsFriendOf. It is often insufficient to treat these relations separately one at a time since their interdependence matters.

In this section we generalize the graph model to graphs with multiple relations, that is, collections of graphs with common vertex set.

Definition 9.4.1. *A* graph with multiple relations $\mathcal{G} = (V, \mathcal{E})$ *consists of a finite vertex set* V*, and a finite set of relations (finite set of edge sets)* $\mathcal{E} = \{E_i\}_{i=1,\dots,p}$*, where* $p \in \mathbb{N}$ *and* $E_i \subseteq V \times V$*.*

For the remainder of this section we often write 'graph' meaning 'graph with multiple relations'. A graph is identified with the one resulting from deleting duplicate relations, where we say that two relations are equal if they consist of the same pairs of vertices. That is relations don't have 'labels' but are distinguished by the pairs of vertices they contain.

The role graph of a graph with multiple relations is again a graph with (possibly) multiple relations. (Compare Definition 9.0.3 of the role graph of a graph with one relation.)

Definition 9.4.2. *Let* $\mathcal{G} = (V, \mathcal{E})$ *be a graph with multiple relations, and* $r \colon V \to W$ *be a role assignment. The* role graph *of* $\mathcal{G}$ *over* r *is the graph* $\mathcal{R} = (W, \mathcal{F})$*, where* $\mathcal{F} = \{F_i\,;\ i = 1, \dots, p\}$*, where* $F_i = \{(r(u), r(v))\,;\ (u, v) \in E_i\}$*.*

Note that F_i may be equal to F_j even if $E_i \neq E_j$ and that duplicate edge relations are eliminated ($\mathcal{F}$ is a set).

From the above definition we can see that role assignments are actually mappings of vertices and relations. That is $r \colon V \to W$ defines uniquely a mapping of relations $r_{\mathrm{rel}} \colon \mathcal{E} \to \mathcal{F}$. Note that r_{rel} does not map edges of $\mathcal{G}$ onto edges of $\mathcal{R}$ but relations, i. e. edge sets, onto relations.

Having more then one relation, the possibilities for defining different types of role assignments explode. See [579, 471] for a large number of possibilities. We will sketch some of them.

The easiest way to translate definitions for different types of vertex partitions (see Sections 9.1, 9.2, and 9.3) to graphs with multiple relations is by the following generic definition.

Definition 9.4.3. *A role assignment* $r \colon V \to W$ *is said to be of a specific type* t *for a graph* $\mathcal{G} = (V, \mathcal{E})$ *with multiple relations, if for each* $E \in \mathcal{E}$*,* r *is of type* t *for the graph* (V, E)*.*

We illustrate this for the definition of regular equivalence relations.

Definition 9.4.4 ([579]). *Let* $\mathcal{G} = (V, \mathcal{E})$ *be a graph. A role assignment* $r \colon V \to W$ *is called* regular *for* $\mathcal{G}$ *if for each* $E \in \mathcal{E}$*,* r *is regular for graph* (V, E)*.*

Besides this natural translation of role assignments from graphs to graphs with multiple relations there is a weaker form (e.g. *weak regular network homomorphism* [579]), which makes use of the mapping of relations r_{rel}.

Theorems for certain types of vertex partitions (see Sections 9.1, 9.2, and 9.3) mostly translate to the case of multiple relations if we apply Definition 9.4.3.

Next we introduce a stronger form of compatibility with multiply relations. Regular role assignments as defined in Definition 9.4.4 make sure that equivalent vertices have, in each of the graphs relations identical ties to equivalent counterparts. Sometimes it is considered as desirable that they have the same combinations of relations to equivalent counterparts. That is, if we consider the example at the beginning of this section, it matters whether an individual gives orders to someone and is the friend of another individual or whether he gives orders to a friend.

Definition 9.4.7 formalizes this. First we need some preliminary definitions:

Definition 9.4.5 ([579]). *Given a graph $\mathcal{G} = (V, \mathcal{E})$ and $u, v \in V$, we define the* bundle (of relations) *from u to v as*

$$B_{uv} = \{E \in \mathcal{E}\,;\ (u, v) \in E\}\ .$$

These bundles define a new graph with multiple relations.

Definition 9.4.6 ([191, 579]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph and $\mathcal{B}$ be the set of all non-empty bundles. For each bundle $B \in \mathcal{B}$ defines a graph with vertex set V and edge set M_B where $(u, v) \in M_B$ if and only if $B_{uv} = B$. M_B is called a* multiplex *relation induced by the graph $\mathcal{G} = (V, \mathcal{E})$. Let $\mathcal{M} = \{M_B\}_{B \in \mathcal{B}}$, then $MPX(\mathcal{G}) := (V, \mathcal{M})$ is called the* multiplex graph *of $\mathcal{G}$.*

For each pair of vertices (u, v) there is a unique bundle associated with it. This bundle may be either empty or a member of $\mathcal{B}$ (the set of all non-empty bundles). This implies that either (u, v) is a member of no M_B or has only one such multiplex relation. Thus, the multiplex graph of a graph can be viewed as a graph with a single relation, but with edge-labels. We call such a graph a *multiplex graph* [579]. That is, a multiplex graph is a graph $\mathcal{G} = (V, \mathcal{M})$ such that for each pair of relations $M_1, M_2 \in \mathcal{M}$ either $M_1 \cap M_2 = \emptyset$ or $M_1 = M_2$ holds.

For example, the multiplex graph $MPX(\mathcal{G})$ of a graph $\mathcal{G}$, is a multiplex graph.

Now we can define the type of equivalence relation which ensures that equivalent vertices have the same bundles of relations to equivalent counterparts.

Definition 9.4.7 ([191]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph with multiple relations. A role assignment $r\colon V \to W$ that is regular for $MPX(\mathcal{G})$ is called* multiplex regular *for $\mathcal{G}$.*

As in the above definition one might define *multiplex strong structural* role assignments, but one can easily verify that a strong structural role assignment on a graph (with multiple relations) is necessarily strong structural on the corresponding multiplex graph.

Remark 9.4.8. An equivalent definition of multiplex regular role assignments is given in [83]: Let $\mathcal{G} = (V, \mathcal{E})$ be a graph, where $\mathcal{E} = \{E_1, \ldots, E_p\}$. Let

$$\mathcal{M} := \left\{ \bigcap_{i \in I} E_i \,;\; I \subseteq \{1, \ldots, p\},\, I \neq \emptyset \right\} .$$

Then the regular role assignments of $(V, \mathcal{M})$ are exactly the multiplex regular role assignments of $\mathcal{G}$.

Regular role assignments of a graph are in general not multiplex regular. Regularity however is preserved in the opposite direction.

Proposition 9.4.9 ([579]). *If $\mathcal{G} = (V, \mathcal{E})$ is a graph, $C := MPX(\mathcal{G})$, and $r\colon V \to W$ a role assignment then the following holds.*

1. *If r is regular for C then it is regular for $\mathcal{G}$.*
2. *If r is strong structural for C then it is strong structural for $\mathcal{G}$.*

Proof. For the proof of 1 and 2 let $E \in \mathcal{E}$ be a relation of $\mathcal{G}$ and let $u, v, u' \in V$ with $(u, v) \in E$ and $r(u) = r(u')$. Let B_{uv} be the bundle of relations of u and v (in particular $E \in B_{uv}$) and let $M := \{(w, w')\,;\; B_{ww'} = B_{uv}\}$ be the corresponding multiplex relation (in particular $(u, v) \in M$).

1. If we assume that r is regular for C, there exist $v' \in V$ such that $r(v') = r(v)$ and $(u', v') \in M$, in particular it is $(u', v') \in E$ which shows the out-part of regularity for $\mathcal{G}$.
2. If we assume that r is strong structural for C, then $(u', v) \in M$, in particular it is $(u', v) \in E$ which shows the out-part of the condition for r being strong structural for $\mathcal{G}$.

The in-parts are treated analogously. □

9.5 The Semigroup of a Graph

Social relations also have an indirect influence: If A and B are friends and B and C are enemies then this (probably) has some influence on the relation between A and C.

In this section we want to formalize such higher-order relations and highlight the relationship with role assignments.

The following definitions and theorems can be found, essentially, in [579], but have been generalized here to graphs with multiple relations (see Section 9.4).

Labeled paths of relations (like EnemyOfAFriend) are formalized by composition of relations; beware of the order.

Definition 9.5.1. *If Q and R are two binary relations on V then the* (Boolean) product *of Q with R is denoted by QR and defined as*

$$QR := \{(u, v)\,;\; \exists w \in V \text{ such that } (u, w) \in Q \text{ and } (w, v) \in R\} .$$

Boolean multiplication of relations corresponds to Boolean multiplication of the associated adjacency matrices, where for two $\{0,1\}$ matrices A and B the Boolean product AB is defined as

$$(AB)_{ij} = \bigvee_{k=1}^{n} A_{ik} \wedge B_{kj} \ .$$

It is also possible to define *real* multiplication of weighted relations or multi-edge sets by real matrix multiplication (this has been advocated e. g., in [89]).

Definition 9.5.2. *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph (with multiple relations). Then, the* semigroup *induced by $\mathcal{G}$ is defined to be*

$$S(\mathcal{G}) := \{E_1 \dots E_k \, ; \ k \in \mathbb{N}, \, E_1, \dots, E_k \in \mathcal{E}\} \ .$$

We also write $S(\mathcal{E})$ for $S(\mathcal{G})$.

Note that two elements in $S(\mathcal{G})$ are equal if and only if they contain the same set of ordered pairs in $V \times V$.

Furthermore, note that $S(\mathcal{G})$ is indeed a semigroup since the multiplication of relations is associative, i. e., $(AB)C = A(BC)$ holds for all relations A, B, and C.

In general, $S(\mathcal{G})$ has no neutral element, relations have no inverse and the multiplication is not commutative.

Although the length of strings in the definition of $S(\mathcal{G})$ is unbounded, $S(\mathcal{G})$ is finite since the number of its elements is bounded by $2^{(|V|^2)}$, the number of all binary relations over V.

The interesting thing about composite relations is the identities satisfied by them. For example we could imagine that on a network of individuals with two relations Friend and Enemy, the identities FriendFriend=Friend and FriendEnemy=EnemyFriend=Enemy hold. At least the fact whether these identities hold or not gives us valuable information about the network. In all cases identities exist necessarily since $S(\mathcal{G})$ is finite but the set of all strings $\{E_1 \dots E_k \, ; \ k \in \mathbb{N}, \, E_i \in \mathcal{E}\}$ is not.

Role assignments identify individuals. Thus they introduce more identities on the semigroup of the graph. The remainder of this section investigates the relationship between role assignments and the identification of relations.

A role assignment on a graph induce a mapping on the induced semigroup.

Definition 9.5.3 ([579]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph with multiple relations and $r \colon V \to W$ a role assignment. For $Q \in S(\mathcal{G})$, $r_{\mathrm{rel}}(Q)$ (compare Section 9.4) is the relation on W defined by $r_{\mathrm{rel}}(Q) := \{(r(u), r(v)) \, ; \ (u,v) \in Q\}$ called the* relation induced *by Q and r. Thus r induces a mapping r_{rel} on the semigroup $S(\mathcal{G})$.*

Note that in general $r_{\mathrm{rel}}(S(\mathcal{G}))$ is not the semigroup of the role graph of $\mathcal{G}$ over r, however, this is true if r is regular. Role assignments do not necessarily preserve composition, i. e., r_{rel} is not a semigroup homomorphism. One of the main results (see Theorem 9.5.6) of this section is that regular role assignments have this property.

Lemma 9.5.4 ([579]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph and $r \colon V \to W$ a role assignment which is regular with respect to Q and $R \in S(\mathcal{G})$. Then, $r_{\mathrm{rel}}(QR) = r_{\mathrm{rel}}(Q) r_{\mathrm{rel}}(R)$.*

Proof. Let $w, w' \in W$ with $(w, w') \in r_{\mathrm{rel}}(QR)$. By the definition of $r_{\mathrm{rel}}(QR)$ there exist $v, v' \in V$ such that $f(v) = w$, $f(v') = w'$, and $(v, v') \in QR$. Therefore there is a vertex $u \in V$ with $(v, u) \in Q$ and $(u, v') \in R$ implying $(w, r(u)) \in r_{\mathrm{rel}}(Q)$ and $(r(c), w') \in r_{\mathrm{rel}}(R)$, whence $(w, w') \in r_{\mathrm{rel}}(Q) r_{\mathrm{rel}}(R)$. We conclude $r_{\mathrm{rel}}(QR) \subseteq r_{\mathrm{rel}}(Q) r_{\mathrm{rel}}(R)$. Note that this holds without the assumption of r being regular.

Conversely, let $w, w' \in W$ with $(w, w') \in r_{\mathrm{rel}}(Q) r_{\mathrm{rel}}(R)$. Then there is a $z \in W$ such that $(w, z) \in r_{\mathrm{rel}}(Q)$ and $(z, w') \in r_{\mathrm{rel}}(R)$. By the definition of r_{rel} there are $v, v', u_1, u_2 \in V$ with $r(v) = w$, $r(v') = w'$, $r(u_1) = r(u_2) = z$, $(v, u_1) \in Q$, and $(u_2, v') \in R$. Since r is regular and $r(u_1) = r(u_2)$ there is a vertex $v'' \in V$ with $r(v'') = f(v')$ and $(u_1, v'') \in R$. It follows that $(v, v'') \in QR$ whence $(w, w') = (r(v), r(v'')) \in r_{\mathrm{rel}}(QR)$, implying $r_{\mathrm{rel}}(Q) r_{\mathrm{rel}}(R) \subseteq r_{\mathrm{rel}}(QR)$. □

The next theorem shows that regular or strong structural on the set of generator relations $\mathcal{E}$ implies regular resp. strong structural on the semigroup $S(\mathcal{E})$. This is the second step in proving Theorem 9.5.6.

Theorem 9.5.5 ([579]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph. If $r \colon V \to W$ is regular (strong structural) with respect to $\mathcal{E}$ then r is regular (strong structural) for any relation in $S(\mathcal{G})$.*

Proof. By induction on the string length of a relation in $S(\mathcal{G})$ written as a product of generating relations (see definition 9.5.2), it suffices to show that if r is regular (strong structural) with respect to two relations $Q, R \in S(\mathcal{G})$, then it is regular (strong structural) for the product QR. So let $Q, R \in S(\mathcal{G})$ be two relations and $u, v \in V$ such that $(r(u), r(v)) \in r_{\mathrm{rel}}(QR)$. By Lemma 9.5.4, this implies $(r(u), r(v)) \in r_{\mathrm{rel}}(Q) r_{\mathrm{rel}}(R)$, whence there is a $w \in W$ such that $(r(u), w) \in r_{\mathrm{rel}}(Q)$ and $(w, r(v)) \in r_{\mathrm{rel}}(R)$. Since r is surjective, there exists $u_0 \in V$ with $r(u_0) = w$, and it is $(r(u), r(u_0)) \in r_{\mathrm{rel}}(Q)$ and $(r(u_0), r(v)) \in r_{\mathrm{rel}}(R)$.

Now, suppose that r is *regular* with respect to Q and R. We have to show the existence of $c, d \in V$ such that $(c, v) \in QR$, $(u, d) \in QR$, $r(c) = r(u)$ and $r(d) = r(v)$. Since r is regular with respect to Q and $(r(u), r(u_0)) \in r_{\mathrm{rel}}(Q)$ there exists $u_1 \in V$ such that $r(u_1) = r(u_0)$ and $(u, u_1) \in Q$. Similarly, since r is regular with respect to R and $(r(u_0), r(v)) \in r_{\mathrm{rel}}(R)$, there exists $d \in V$ such that $r(d) = r(v)$, and $(u_1, d) \in R$. Since $(u, u_1) \in Q$ and $(u_1, d) \in R$ it follows $(u, d) \in QR$, which is the first half of what we have to show. The proof of the second half can be done along the same lines.

Now, suppose that f is *strong structural* with respect to Q and R. Then $(r(u), r(u_0)) \in r_{\mathrm{rel}}(Q)$ and $(r(u_0), r(v)) \in r_{\mathrm{rel}}(R)$ immediately implies $(u, u_0) \in Q$ and $(u_0, v) \in R$, whence $(u, v) \in QR$. □

The next theorem might be seen as the main result of this section. It states that regular role assignments induce homomorphisms on the induced semigroups.

Theorem 9.5.6 ([579]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph with multiple relations. If $r \colon V \to W$ is a regular role assignment with role graph $\mathcal{R}$, then $r_{\mathrm{rel}} \colon S(\mathcal{G}) \to S(\mathcal{R})$ is a surjective semigroup homomorphism.*

Proof. We know from Lemma 9.5.4 that the identity $r_{\mathrm{rel}}(QR) = r_{\mathrm{rel}}(Q)r_{\mathrm{rel}}(R)$ holds whenever r is regular with respect to Q and R. Theorem 9.5.5 states that r is regular with respect to all relations in $S(\mathcal{G})$. Thus the image of $S(\mathcal{G})$ under r_{rel} is equal to $S(\mathcal{R})$ (the images of the generator relations $\mathcal{E}$ are the generator relations of the semigroup of the role graph $S(\mathcal{R})$) and r_{rel} is a semigroup homomorphism. □

The condition that r be regular, is not necessary for r_{rel} being a semigroup homomorphism. Kim and Roush [355] gave a more general sufficient condition. Also compare [471].

The next theorem shows that the role graph of a strong structural role assignment has the same semigroup as the original graph.

Theorem 9.5.7 ([579]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph with multiple relations. If $r \colon V \to W$ is a strong structural role assignment with role graph $\mathcal{R}$, then $r_{\mathrm{rel}} \colon S(\mathcal{G}) \to S(\mathcal{R})$ is a semigroup isomorphism.*

Proof. By Theorem 9.5.6 r_{rel} is a surjective semigroup homomorphism. It remains to show that r_{rel} is injective. So let $Q, R \in S(\mathcal{G})$ with $r_{\mathrm{rel}}(Q) = r_{\mathrm{rel}}(R)$. Then, for all $u, v \in V$ if holds $(u, v) \in Q$ iff $(r(u), r(v)) \in r_{\mathrm{rel}}(Q)$ (since r is strong) iff $(r(u), r(v)) \in r_{\mathrm{rel}}(R)$ iff $(u, v) \in R$ (since r is strong). □

Do Semigroup-Homomorphisms Reduce Networks? The above theorems give the idea to an alternative approach to find role assignments: In Theorem 9.5.6 it has been shown that role assignments introduce new identities on the semigroup of (generator and compound) relations of a network. Conversely, one could impose identities on relations that are almost satisfied, or that are considered to be reasonable. Now the interesting question is: *Does identification of relations imply identification of vertices of the graph which generated the semigroup?* (See [73].)

That is, given a graph $\mathcal{G}$ with semigroup $S(\mathcal{G})$ and a surjective semigroup homomorphism $S(\mathcal{G}) \to S'$ onto some semigroup S', *is there a graph $\mathcal{G}'$ and a graph homomorphism $\mathcal{G} \to \mathcal{G}'$ such that S' is the semigroup generated by $\mathcal{G}'$?*

This would be the counterpart of Theorem 9.5.6, which states that role assignments on graphs induce, under the condition of regularity, reductions of the induced semigroups, (i. e., surjective semigroup homomorphisms).

The answer is in general *no*, simply for the reason that not every semigroup is a semigroup of relations. But *under what conditions on S' and on the semigroup* *homomorphism would we get a meaningful role graph and a meaningful role assignment?*

Although the question is open for the general case some examples can be found in [89] and [471].

9.5.1 Winship-Pattison Role Equivalence

The condition for regular equivalent vertices is: *equivalent vertices have the same ties to equivalent counterparts*. In this section the phrase *to equivalent counterparts* is replaced by the weaker requirement *to some vertices*. As mentioned in Remark 9.5.9 the four equivalences defined in this section, are special cases of relative regular equivalence (see Section 9.3.4).

Definition 9.5.8. *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph and $\sim$ an equivalence on V. Then $\sim$ is said to be a* weak role equivalence *for $\mathcal{G}$ if for all $u, v, w \in V$ and $E \in \mathcal{E}$, $u \sim v$ implies both*

- *uRw implies there exists x such that vRx,*
- *wRu implies there exists x such that xRv.*

Note that in contrast to the definition of regular equivalence one does not consider the role of x. So weak role-equivalent vertices don't share the same relations *to equivalent counterparts*, but they only share the same relations. If the graph has one single relation, the maximal weak role equivalence is simply the partition into isolates, sinks, sources, and vertices with positive in- and out-degree.

The indifference in regard to the role of adjacent vertices makes weak role equivalence a much weaker requirement than e. g., regular or strong structural equivalences.

Weak role equivalence could have been defined using relative regular equivalence (see Section 9.3.4).

Remark 9.5.9. Weak role equivalences are exactly the equivalences which are regular relative to the complete partition. This remark immediately generalizes to the next three definitions.

Weak role equivalence can be tightened in two directions: to include multiplexity, which leads to Definition 9.5.11, or to include composition of relations, which leads to Definition 9.5.10.

Definition 9.5.10. *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph, $S := S(\mathcal{G})$ its semigroup, and $\sim$ an equivalence on V. Then $\sim$ is called a* compositional equivalence *of $\mathcal{G}$ if it is a weak role equivalence of (V, S) (see Definition 9.5.8).*

Note that in contrast to regular equivalences, where an equivalence is regular with respect to $\mathcal{E}$ if and only if it is regular with respect to $S(\mathcal{E})$, it makes a difference whether we require $\sim$ to be a weak role equivalence of $\mathcal{G}$ or of (V, S). Compositional equivalences are weak role equivalences.

Definition 9.5.11 ([579]). *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph, $C = (V, \mathcal{M}) := MPX(\mathcal{G})$ its multiplex graph (see Definition 9.4.6) and $\sim$ an equivalence on V. Then, $\sim$ is called a* bundle equivalence *of $\mathcal{G}$ if it is a weak role equivalence (see Definition 9.5.8) of C.*

Bundle equivalences are weak role equivalences.

Winship-Pattison role equivalence is most often defined in terms of the *role-set* of an actor (see [471, p. 79ff]): Two actors are equivalent if they have the same role-sets (also compare [82, p. 81]). We restate the definitions given there in our terminology.

Definition 9.5.12. *Let $\mathcal{G} = (V, \mathcal{E})$ be a graph. An equivalence relation $\sim$ on V is called a* local role equivalence *or* Winship-Pattison role equivalence *if $\sim$ is a bundle equivalence (see Definition 9.5.11) of the graph $(V, S(\mathcal{G}))$.*

Local role equivalences are both bundle and compositional equivalences. Local role equivalences are, in general, not regular, which immediately implies the same for the three other (weaker) equivalences defined in this section: Let vertices u and v be connected by a bidirected edge and v have an out-going edge to a third vertex w. Then u and v are locally role equivalent but not regularly equivalent.

Conclusion. The semigroup of a graph is a possibility to describe the interaction of multiple and compound relations. An idea to use identification of relations in order to get role assignments has been sketched. This approach seems to be rather hard, both theoretically and computationally.

9.6 Chapter Notes

Vertex partitions that yield role assignments have first been introduced by Lorrain and White [394], who defined structural equivalence.

Sailer [501] pointed out that structural equivalence is to restrictive to meet the intuitive notion of social role. He proposed that actors play the same role if they are connected to *role-equivalent* actors (in contrast to *identical* actors, as structural equivalence demands). His idea of structural relatedness has been formalized as regular equivalence by White and Reitz in the seminal paper [579]. In this work, they gave a unified treatment of structural, regular, and other equivalences for graphs with single or multiple relations. Furthermore, they developed conditions for graph homomorphisms to induce (structural or regular) vertex partitions and to be compatible with the composition of relations.

Borgatti and Everett [82, 83, 190, 191] established many properties of the set of regular equivalences, including lattice structure, and developed the algorithm CATREGE to compute the maximal regular equivalence of a graph. Furthermore they introduced other types of vertex partitions to define roles in graphs. Boyd and Everett [90] further clarified the lattice structure and defined relative regular equivalence.

Marx and Masuch [408] commented that regular equivalence is already known, under the name of bisimulation in computer science. Their report has been the reason that we found the algorithm of Paige and Tarjan [459], which can compute the maximal regular equivalence and is much faster than CATREGE.

Roberts and Sheng [493] first showed that there are $\mathcal{NP}$-complete problems stemming from regular role assignments. A more complete treatment is from Fiala and Paulusma [209].

Role assignments for graphs with multiple and composite relations are already treated in [394, 579]. The possibilities to define role assignments in graphs with multiple relations are abundant. We could sketch only few of them in this chapter. Additional reading is, e. g., Kim and Roush [355] and Pattison [471] who found many conditions for vertex partitions to be compatible with the composition of relations. In the latter book, the algebraic structure of semigroups of relations is presented in detail. Boyd [89] advocated the use of real matrix multiplication to define semigroups stemming from graphs. These semigroups often admit sophisticated decompositions, which in turn, induce decompositions or reductions of the graphs that generated these semigroups.

In order to be able to deal with the irregularities of empirical networks, a formalization of role assignment must – in addition to choosing the right compatibility criterion – provide some kind of relaxation. (See Wasserman and Faust [569] for a more detailed explanation.) Relaxation has not been treated in this chapter, which has been focused on the 'ideal' case of vertex partitions that satisfy strictly the different compatibility constraints. Possibilities to relax structural equivalence, optimizational approaches for regular equivalence, and stochastic methods for role assignments are presented in Chapter 10 about blockmodels. Brandes and Lerner [97] introduced a relaxation of equitable partitions to provide a framework for role assignments that are tolerant towards irregularities.

Bibliography

1. Serge Abiteboul, Mihai Preda, and Gregory Cobena. Adaptive on-line page importance computation. In *Proceedings of the 12th International World Wide Web Conference (WWW12)*, pages 280–290, Budapest, Hungary, 2003.
2. Forman S. Acton. *Numerical Methods that Work*. Mathematical Association of America, 1990.
3. Alan Agresti. *Categorical Data Analysis*. Wiley, 2nd edition, 2002.
4. Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
5. Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983.
6. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
7. Ravindra K. Ahuja and James B. Orlin. A fast and simple algorithm for the maximum flow problem. *Operations Research*, 37(5):748–759, September/October 1989.
8. Ravindra K. Ahuja and James B. Orlin. Distance-based augmenting path algorithms for the maximum flow and parametric maximum flow problems. *Naval Research Logistics Quarterly*, 38:413–430, 1991.
9. William Aiello, Fan R. K. Chung, and Linyuan Lu. A random graph model for massive graphs. In *Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing (STOC'00)*, pages 171–180, May 2000.
10. Martin Aigner. *Combinatorial Theory*. Springer-Verlag, 1999.
11. Martin Aigner and Eberhard Triesch. Realizability and uniqueness in graphs. *Discrete Mathematics*, 136:3–20, 1994.
12. Donald Aingworth, Chandra Chekuri, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). In *Proceedings of the 7th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'96)*, 1996.
13. Jin Akiyama, Francis T. Boesch, Hiroshi Era, Frank Harary, and Ralph Tindell. The cohesiveness of a point of a graph. *Networks*, 11(1):65–68, 1981.
14. Richard D. Alba. A graph theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:113–126, 1973.
15. Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.
16. Réka Albert, Hawoong Jeong, and Albert-László Barabási. Diameter of the world wide web. *Nature*, 401:130–131, September 1999.
17. Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, July 2000.
18. Mark S. Aldenderfer and Roger K. Blashfield. *Cluster Analysis*. Sage, 1984.
19. Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.
20. Noga Alon. Generating pseudo-random permutations and maximum flow algorithms. *Information Processing Letters*, 35(4):201–204, 1990.

21. Noga Alon, Fan R. K. Chung, and Ronald L. Graham. Routing permutations on graphs via matchings. *SIAM Journal on Discrete Mathematics*, 7:513–530, 1994.
22. Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Randoms Structures and Algorithms*, 13(3–4):457–466, 1998.
23. Noga Alon and Vitali D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory Series B*, 38:73–88, 1985.
24. Noga Alon and Joel Spencer. *The Probabilistic Method*. Wiley, 1992.
25. Noga Alon, Joel Spencer, and Paul Erdős. *The Probabilistic Method*. Wiley, 1992.
26. Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
27. Charles J. Alpert and Andrew B. Kahng. Recent directions in netlist partitioning: A survey. *Integration: The VLSI Journal*, 19(1-2):1–81, 1995.
28. Ashok T. Amin and S. Louis Hakimi. Graphs with given connectivity and independence number or networks with given measures of vulnerability and survivability. *IEEE Transactions on Circuit Theory*, 20(1):2–10, 1973.
29. Carolyn J. Anderson, Stanley Wasserman, and Bradley Crouch. A p^* primer: Logit models for social networks. *Social Networks*, 21(1):37–66, January 1999.
30. Carolyn J. Anderson, Stanley Wasserman, and Katherine Faust. Building stochastic blockmodels. *Social Networks*, 14:137–161, 1992.
31. James G. Anderson and Stephen J. Jay. The diffusion of medical technology: Social network analysis and policy research. *The Sociological Quarterly*, 26:49–64, 1985.
32. Jacob M. Anthonisse. The rush in a directed graph. Technical Report BN 9/71, Stichting Mathematisch Centrum, 2e Boerhaavestraat 49 Amsterdam, October 1971.
33. Arvind Arasu, Jasmine Novak, Andrew S. Tomkins, and John Tomlin. PageRank computation and the structure of the web: experiments and algorithms. short version appeared in Proceedings of the 11th International World Wide Web Conference, Poster Track, November 2001.
34. Ludwig Arnold. On the asymptotic distribution of the eigenvalues of random matrices. *Journal of Mathematical Analysis and Applications*, 20:262–268, 1967.
35. Sanjeev Arora, David R. Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of $\mathcal{NP}$-hard problems. *Journal of Computer and System Sciences*, 58(1):193–210, 1999.
36. Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander fbws, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing (STOC'04)*, pages 222–231. ACM Press, 2004.
37. Yuichi Asahiro, Refael Hassin, and Kazuo Iwama. Complexity of finding dense subgraphs. *Discrete Applied Mathematics*, 121(1–3):15–26, 2002.
38. Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.
39. Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, and Alberto Marchetti-Spaccamela. *Complexity and Approximation - Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag, 2nd edition, 2002.
40. Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
41. Alain Barrat and Martin Weigt. On the properties of small-world network models. *The European Physical Journal B*, 13:547–560, 2000.
42. Vladimir Batagelj. Notes on blockmodeling. *Social Networks*, 19(2):143–155, April 1997.

43. Vladimir Batagelj and Ulrik Brandes. Efficient generation of large random networks. *Physical Review E*, 2005. To appear.
44. Vladimir Batagelj and Anuška Ferligoj. Clustering relational data. In Wolfgang Gaul, Otto Opitz, and Martin Schader, editors, *Data Analysis*, pages 3–15. Springer-Verlag, 2000.
45. Vladimir Batagelj, Anuška Ferligoj, and Patrick Doreian. Generalized blockmodeling. *Informatica: An International Journal of Computing and Informatics*, 23:501–506, 1999.
46. Vladimir Batagelj and Andrej Mrvar. Pajek – A program for large network analysis. *Connections*, 21(2):47–57, 1998.
47. Vladimir Batagelj and Matjaž Zaveršnik. An $\mathcal{O}(m)$ algorithm for cores decomposition of networks. Technical Report 798, IMFM Ljublana, Ljubljana, 2002.
48. Douglas Bauer, S. Louis Hakimi, and Edward F. Schmeichel. Recognizing tough graphs is NP-hard. *Discrete Applied Mathematics*, 28:191–195, 1990.
49. Michel Bauer and Olivier Golinelli. Random incidence matrices: moments of the spectral density. *Journal of Statistical Physics*, 103:301–307, 2001. arXiv cond-mat/0007127.
50. Alex Bavelas. A mathematical model for group structure. *Human Organizations*, 7:16–30, 1948.
51. Alex Bavelas. Communication patterns in task oriented groups. *Journal of the Acoustical Society of America*, 22:271–282, 1950.
52. Murray A. Beauchamp. An improved index of centrality. *Behavioral Science*, 10:161–163, 1965.
53. M. Becker, W. Degenhardt, Jürgen Doenhardt, Stefan Hertel, G. Kaninke, W. Keber, Kurt Mehlhorn, Stefan Näher, Hans Rohnert, and Thomas Winter. A probabilistic algorithm for vertex connectivity of graphs. *Information Processing Letters*, 15(3):135–136, October 1982.
54. Richard Beigel. Finding maximum independent sets in sparse and general graphs. In *Proceedings of the 10th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'99)*, pages 856–857. IEEE Computer Society Press, 1999.
55. Lowell W. Beineke and Frank Harary. The connectivity function of a graph. *Mathematika*, 14:197–202, 1967.
56. Lowell W. Beineke, Ortrud R. Oellermann, and Raymond E. Pippert. The average connectivity of a graph. *Discrete Mathematics*, 252(1):31–45, May 2002.
57. Claude Berge. *Graphs*. North-Holland, 3rd edition, 1991.
58. Noam Berger, Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver M. Riordan. Degree distribution of the FKP network model. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming (ICALP'03)*, pages 725–738, 2003.
59. Julian E. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society, Series B*, 36:196–236, 1974.
60. Sergej Bezrukov, Robert Elsässer, Burkhard Monien, Robert Preis, and Jean-Pierre Tillich. New spectral lower bounds on the bisection width of graphs. *Theoretical Computer Science*, 320:155–174, 2004.
61. Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. *ACM Transactions on Internet Technology*, 2004. in press.
62. Robert E. Bixby. The minimum number of edges and vertices in a graph with edge connectivity n and m n-bonds. *Bulletin of the American Mathematical Society*, 80(4):700–704, 1974.
63. Robert E. Bixby. The minimum number of edges and vertices in a graph with edge connectivity n and m n-bonds. *Networks*, 5:253–298, 1981.

64. Francis T. Boesch, Frank Harary, and Jerald A. Kabell. Graphs as models of communication network vulnerability: Connectivity and persistence. *Networks*, 11:57–63, 1981.
65. Francis T. Boesch and R. Emerson Thomas. On graphs of invulnerable communication nets. *IEEE Transactions on Circuit Theory*, CT-17, 1970.
66. Béla Bollobás. *Extremal graph theory*. Academic Press, 1978.
67. Béla Bollobás. *Modern Graph Theory*, volume 184 of *Graduate Texts in Mathematics*. Springer-Verlag, 1998.
68. Béla Bollobás and Oliver M. Riordan. Mathematical results on scale-free random graphs. In Stefan Bornholdt and Heinz Georg Schuster, editors, *Handbook of Graphs and Networks: From the Genome to the Internet*, pages 1–34. Wiley-VCH, 2002.
69. Béla Bollobás, Oliver M. Riordan, Joel Spencer, and Gábor Tusnády. The degree sequence of a scale-free random graph process. *Randoms Structures and Algorithms*, 18:279–290, 2001.
70. Immanuel M. Bomze, Marco Budinich, Panos M. Pardalos, and Marcello Pelillo. The maximum clique problem. In Ding-Zhu Du and Panos M. Pardalos, editors, *Handbook of Combinatorial Optimization (Supplement Volume A)*, volume 4, pages 1–74. Kluwer Academic Publishers Group, 1999.
71. Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2:113–120, 1972.
72. Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.
73. Phillip Bonacich. What is a homomorphism? In Linton Clarke Freeman, Douglas R. White, and A. Kimbal Romney, editors, *Research Methods in Social Network Analysis*, chapter 8, pages 255–293. George Mason University Press, 1989.
74. J. Bond and Claudine Peyrat. Diameter vulnerability in networks. In Yousef Alavi, Gary Chartrand, Linda Lesniak, Don R. Lick, and Curtiss E. Wall, editors, *Graph Theory with Applications to Algorithms and Computer Science*, pages 123–149. Wiley, 1985.
75. Robert R. Boorstyn and Howard Frank. Large scale network topological optimization. *IEEE Transaction on Communications*, Com-25:29–37, 1977.
76. Kellogg S. Booth. Problems polynomially equivalent to graph isomorphism. Technical report, CS-77-04, University of Ljublana, 1979.
77. Kellogg S. Booth and George S. Lueker. Linear algorithms to recognize interval graphs and test for consecutive ones property. *Proceedings of the 7th Annual ACM Symposium on the Theory of Computing (STOC'75)*, pages 255–265, 1975.
78. Ravi B. Boppana. Eigenvalues and graph bisection: an average case analysis. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science (FOCS'87)*, pages 280–285, October 1987.
79. Ravi B. Boppana and Magnús M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, 1992.
80. Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-$\mathcal{NP}$ have short interactive proofs? *Information Processing Letters*, 25:127–132, 1987.
81. Stephen P. Borgatti. Centrality and AIDS. *Connections*, 18(1):112–115, 1995.
82. Stephen P. Borgatti and Martin G. Everett. The class of all regular equivalences: Algebraic structure and computation. *Social Networks*, 11(1):65–88, 1989.
83. Stephen P. Borgatti and Martin G. Everett. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361–376, 1993.
84. Stephen P. Borgatti and Martin G. Everett. Models of core/periphery structures. *Social Networks*, 21(4):375–395, 1999.
85. Stephen P. Borgatti and Martin G. Everett. A graph-theoretic perspective on centrality. Unpublished manuscript, 2004.

86. Stephen P. Borgatti, Martin G. Everett, and Paul R. Shirey. LS sets, lambda sets and other cohesive subsets. *Social Networks*, 12(4):337–357, 1990.
87. Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *Proceedings of the 10th International World Wide Web Conference (WWW10)*, pages 415–429, Hong Kong, 2001.
88. Rodrigo A. Botagfogo, Ehud Rivlin, and Ben Shneiderman. Structural analysis of hypertexts: Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems*, 10(2):142–180, 1992.
89. John P. Boyd. *Social Semigroups*. George Mason University Press, 1991.
90. John P. Boyd and Martin G. Everett. Relations, residuals, regular interiors, and relative regular equivalence. *Social Networks*, 21(2):147–165, April 1999.
91. Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
92. Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.
93. Ulrik Brandes and Sabine Cornelsen. Visual ranking of link structures. *Journal of Graph Algorithms and Applications*, 7(2):181–201, 2003.
94. Ulrik Brandes and Daniel Fleischer. Centrality measures based on current fbw. In *Proceedings of the 22nd International Symposium on Theoretical Aspects of Computer Science (STACS'05)*, volume 3404 of *Lecture Notes in Computer Science*, 2005. To appear.
95. Ulrik Brandes, Marco Gaertler, and Dorothea Wagner. Experiments on graph clustering algorithms. In *Proceedings of the 11th Annual European Symposium on Algorithms (ESA'03)*, volume 2832 of *Lecture Notes in Computer Science*, pages 568–579, September 2003.
96. Ulrik Brandes, Patrick Kenis, and Dorothea Wagner. Communicating centrality in policy network drawings. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):241–253, 2003.
97. Ulrik Brandes and Jürgen Lerner. Structural similarity in graphs. In *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC'04)*, volume 3341 of *Lecture Notes in Computer Science*, pages 184–195, 2004.
98. Ronald L. Breiger. Toward an operational theory of community elite structures. *Quality and Quantity*, 13:21–57, 1979.
99. Ronald L. Breiger, Scott A. Boorman, and Phipps Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of Mathematical Psychology*, 12:328–383, 1975.
100. Ronald L. Breiger and James G. Ennis. Personae and social roles: The network structure of personality types in small groups. *The Sociological Quarterly*, 42:262–270, 1979.
101. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
102. Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew S. Tomkins, and Janet Wiener. Graph structure in the Web. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33(1–6):309–320, 2000.
103. Coen Bron and Joep A. G. M. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.
104. Tian Bu and Don Towsley. On distinguishing between Internet power law topology generators. In *Proceedings of Infocom'02*, 2002.

105. Mihai Bădoiu. Approximation algorithm for embedding metrics into a two-dimensional space. In *Proceedings of the 14th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'03)*, pages 434–443, 2003.
106. Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19:255–259, 1998.
107. Ronald S. Burt. Positions in networks. *Social Forces*, 55:93–122, 1976.
108. Duncan S. Callaway, Mark E. J. Newman, Steven H. Strogatz, and Duncan J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 25(85):5468–5471, December 2000.
109. Kenneth L. Calvert, Matthew B. Doar, and Ellen W. Zegura. Modeling Internet topology. *IEEE Communications Magazine*, 35:160–163, June 1997.
110. A. Cardon and Maxime Crochemore. Partitioning a graph in $\mathcal{O}(|a| \log_2 |v|)$. *Theoretical Computer Science*, 19:85–98, 1982.
111. Tami Carpenter, George Karakostas, and David Shallcross. Pracical Issues and Algorithms for Analyzing Terrorist Networks. invited talk at WMC 2002, 2002.
112. Peter J. Carrington, Greg H. Heil, and Stephen D. Berkowitz. A goodness-of-fit index for blockmodels. *Social Networks*, 2:219–234, 1980.
113. Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of the 3rd International Workshop on Approximatin Algorithms for Combinatorial Optimization (APPROX'00)*, volume 1931 of *Lecture Notes in Computer Science*, pages 84–95. Springer-Verlag, 2000.
114. Gary Chartrand. A graph-theoretic approach to a communications problem. *SIAM Journal on Applied Mathematics*, 14(5):778–781, July 1966.
115. Gary Chartrand, Gary L. Johns, Songlin Tian, and Steven J. Winters. Directed distance on digraphs: Centers and medians. *Journal of Graph Theory*, 17(4):509–521, 1993.
116. Gary Chartrand, Grzegorz Kubicki, and Michelle Schultz. Graph similarity and distance in graphs. *Aequationes Mathematicae*, 55(1-2):129–145, 1998.
117. Qian Chen, Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. The origin of power laws in internet topologies revisited. In *Proceedings of Infocom'02*, 2002.
118. Joseph Cheriyan and Torben Hagerup. A randomized maximum-fbw algorithm. *SIAM Journal on Computing*, 24(2):203–226, 1995.
119. Joseph Cheriyan, Torben Hagerup, and Kurt Mehlhorn. An $o(n^3)$-time maximum-fbw algorithm. *SIAM Journal on Computing*, 25(6):144–1170, December 1996.
120. Joseph Cheriyan and John H. Reif. Directed s-t numberings, rubber bands, and testing digraph k-vertex connectivity. In *Proceedings of the 3rd Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'92)*, pages 335–344, January 1992.
121. Joseph Cheriyan and Ramakrishna Thurimella. Fast algorithms for k-shredders and k-node connectivity augmentation. *Journal of Algorithms*, 33:15–50, 1999.
122. Boris V. Cherkassky. An algorithm for constructing a maximal fbw through a network requiring $\mathcal{O}(n^2\sqrt{p})$ operations. *Mathematical Methods for Solving Economic Problems*, 7:117–126, 1977. (In Russian).
123. Boris V. Cherkassky. A fast algorithm for constructing a maximum fbw through a network. In *Selected Topics in Discrete Mathematics: Proceedings of the Moscow Discrete Mathematics Seminar, 1972-1990*, volume 158 of *American Mathematical Society Translations – Series 2*, pages 23–30. AMS, 1994.
124. Steve Chien, Cynthia Dwork, Ravi Kumar, and D. Sivakumar. Towards exploiting link evolution. In *Workshop on Algorithms and Models for the Web Graph*, November 2002.
125. Fan R. K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.

126. Fan R. K. Chung, Vance Faber, and Thomas A. Manteuffel. An upper bound on the diameter of a graph from eigenvalues associated with its laplacian. *SIAM Journal on Discrete Mathematics*, 7(3):443–457, 1994.
127. Fan R. K. Chung, Linyuan Lu, and Van Vu. Eigenvalues of random power law graphs. *Annals of Combinatorics*, 7:21–33, 2003.
128. Fan R. K. Chung, Linyuan Lu, and Van Vu. The spectra of random graphs with given expected degree. *Proceedings of the National Academy of Science of the United States of America*, 100(11):6313–6318, May 2003.
129. Vašek Chvátal. Tough graphs and hamiltionian circuits. *Discrete Mathematics*, 5, 1973.
130. Reuven Cohen, Keren Erez, Daniel ben Avraham, and Shlomo Havlin. Resilience of the Internet to random breakdown. *Physical Review Letters*, 21(85):4626–4628, November 2000.
131. Colin Cooper and Alan M. Frieze. A general model of web graphs. *Randoms Structures and Algorithms*, 22:311–335, 2003.
132. Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.
133. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press, 2nd edition, 2001.
134. Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling.* Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2nd edition, 2001.
135. Dragoš M. Cvetković, Michael Doob, and Horst Sachs. *Spectra of Graphs.* Johann Ambrosius Barth Verlag, 1995.
136. Dragoš M. Cvetković, Peter Rowlinson, and Slobodan Simic. *Eigenspaces of Graphs.* Cambridge University Press, 1997.
137. Andrzej Czygrinow. Maximum dispersion problem in dense graphs. *Operations Research Letter*, 27(5):223–227, 2000.
138. Peter Dankelmann and Ortrud R. Oellermann. Bounds on the average connectivity of a graph. *Discrete Applied Mathematics*, 129:305–318, August 2003.
139. George B. Dantzig. Application of the simplex method to a transportation problem. In Tjalling C. Koopmans, editor, *Activity Analysis of Production and Allocation*, volume 13 of *Cowles Commission for Research in Economics*, pages 359–373. Wiley, 1951.
140. George B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In Tjalling C. Koopmans, editor, *Activity Analysis of Production and Allocation*, volume 13 of *Cowles Commission for Research in Economics*, pages 339–347. Wiley, 1951.
141. George B. Dantzig and Delbert R. Fulkerson. On the max-fbw min-cut theorem of networks. In *Linear Inequalities and Related Systems*, volume 38 of *Annals of Mathematics Studies*, pages 215–221. Princeton University Press, 1956.
142. Camil Demetrescu and Giuseppe F. Italiano. A new approach to dynamic all pairs shortest paths. In *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing (STOC'03)*, pages 159–166, June 2003.
143. Guiseppe Di Battista and Roberto Tamassia. Incremental planarity testing. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS'89)*, pages 436–441, October/November 1989.
144. Guiseppe Di Battista and Roberto Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15:302–318, 1996.
145. Reinhard Diestel. *Graph Theory.* Graduate Texts in Mathematics. Springer-Verlag, 2nd edition, 2000.
146. Edsger W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

147. Stephen Dill, Ravi Kumar, Kevin S. McCurley, Sridhar Rajagopalan, D. Sivakumar, and Andrew S. Tomkins. Self-similarity in the web. *ACM Transactions on Internet Technology*, 2(3):205–223, August 2002.
148. Chris H. Q. Ding, Xiaofeng He, Parry Husbands, Hongyuan Zha, and Horst D. Simon. PageRank, HITS and a unified framework for link analysis. LBNL Tech Report 49372, NERSC Division, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, USA, November 2001. updated Sept. 2002 (LBNL-50007), presented in the poster session of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, 2003.
149. Chris H. Q. Ding, Hongyuan Zha, Xiaofeng He, Parry Husbands, and Horst D. Simon. Link analysis: Hubs and authorities on the world wide web. *SIAM Review*, 46(2), 2004. to appear, published electronically May, 3, 2004.
150. Yefim Dinitz. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Mathematics-Doklady*, 11(5):1277–1280, 1970.
151. Yefim Dinitz. Bitwise residual decreasing method and transportation type problems. In A. A. Fridman, editor, *Studies in Discrete Mathematics*, pages 46–57. Nauka, 1973. (In Russian).
152. Yefim Dinitz. Finding shortest paths in a network. In Y. Popkov and B. Shmulyian, editors, *Transportation Modeling Systems*, pages 36–44. Institute for System Studies, Moscow, 1978.
153. Yefim Dinitz, Alexander V. Karzanov, and M. V. Lomonosov. On the structure of the system of minimum edge cuts in a graph. In A. A. Fridman, editor, *In Studies in Discrete Optimization*, pages 290–306. Nauka, 1976.
154. Yefim Dinitz and Ronit Nossenson. Incremental maintenance of the 5-edge-connectivity classes of a graph. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory (SWAT'00)*, volume 1851 of *Lecture Notes in Mathematics*, pages 272–285. Springer-Verlag, July 2000.
155. Yefim Dinitz and Jeffery Westbrook. Maintaining the classes of 4-edge-connectivity in a graph on-line. *Algorithmica*, 20(3):242–276, March 1998.
156. Gabriel A. Dirac. Extensions of Turán's theorem on graphs. *Acta Mathematica Academiae Scientiarum Hungaricae*, 14:417–422, 1963.
157. Matthew B. Doar. A better model for generating test networks. In *IEEE GLOBECOM'96*, 1996.
158. Wolfgang Domschke and Andreas Drexl. *Location and Layout Planning: An International Bibliography*. Springer-Verlag, Berlin, 1985.
159. William E. Donath and Alan J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.
160. Patrick Doreian. Using multiple network analytic tools for a single social network. *Social Networks*, 10:287–312, 1988.
161. Patrick Doreian and Louis H. Albert. Partitioning political actor networks: Some quantitative tools for analyzing qualitative networks. *Journal of Quantitative Anthropology*, 1:279–291, 1989.
162. Patrick Doreian, Vladimir Batagelj, and Anuška Ferligoj. Symmetric-acyclic decompositions of networks. *Journal of Classification*, 17(1):3–28, 2000.
163. Patrick Doreian, Vladimir Batagelj, and Anuška Ferligoj. Generalized blockmodeling of two-mode network data. *Social Networks*, 26(1):29–53, 2004.
164. Sergey N. Dorogovtsev and Jose Ferreira F. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, June 2002.
165. Sergey N. Dorogovtsev and Jose Ferreira F. Mendes. *Evolution of Networks*. Oxford University Press, 2003.
166. Sergey N. Dorogovtsev, Jose Ferreira F. Mendes, and Alexander N. Samukhin. Structure of growing networks: Exact solution of the Barabási-Albert's model. http://xxx.sissa.it/ps/cond-mat/0004434, April 2000.

167. Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II. On completeness for W[1]. *Theoretical Computer Science*, 141(1–2):109–131, 1995.
168. Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer-Verlag, 1999.
169. Zvi Drezner and Horst W. Hamacher, editors. *Facility Location: Application and Theory*. Springer-Verlag, 2002.
170. Petros Drineas, Alan M. Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56:9–33, 2004.
171. Jack Edmonds. Edge-disjoint branchings. In Randall Rustin, editor, *Courant Computer Science Symposium 9: Combinatorial Algorithms (1972)*, pages 91–96. Algorithmics Press, 1973.
172. Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, April 1972.
173. Eugene Egerváry. On combinatorial properties of matrices. *Mat. Lapok*, 38:16–28, 1931.
174. Friedrich Eisenbrand and Fabrizio Grandoni. On the complexity of fixed parameter clique and dominating set. *Theoretical Computer Science*, 326(1–3):57–67, 2004.
175. Peter Elias, Amiel Feinstein, and Claude E. Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2(4):117–119, December 1956.
176. Robert Elsässer and Burkhard Monien. Load balancing of unit size tokens and expansion properties of graphs. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'03)*, pages 266–273, 2003.
177. Lars Engebretsen and Jonas Holmerin. Towards optimal lower bounds for clique and chromatic number. *Theoretical Computer Science*, 299(1-3):537–584, 2003.
178. David Eppstein. Fast hierarchical clustering and other applications of dynamic closest pairs. *j-ea*, 5:1–23, 2000.
179. David Eppstein and Joseph Wang. Fast approximation of centrality. In *Proceedings of the 12th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'01)*, 2001.
180. Paul Erdős and Tibor Gallai. Graphs with prescribed degrees of vertices (in hungarian). *Matematikai Lapok*, 11:264–274, 1960.
181. Paul Erdős and Alfred Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.
182. Abdol-Hossein Esfahanian. Lower-bounds on the connectivities of a graph. *Journal of Graph Theory*, 9(4):503–511, 1985.
183. Abdol-Hossein Esfahanian and S. Louis Hakimi. On computing the connectivities of graphs and digraphs. *Networks*, 14(2):355–366, 1984.
184. Stephen Eubank, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, and Nan Wang. Structural and algorithmic aspects of massive social networks. In *Proceedings of the 14th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'04)*, pages 718–727, 2004.
185. Shimon Even. *Algorithmic Combinatorics*. Macmillan, 1973.
186. Shimon Even. An algorithm for determining whether the connectivity of a graph is at least k. *SIAM Journal on Computing*, 4(3):393–396, September 1975.
187. Shimon Even. *Graph Algorithms*. Computer Science Press, 1979.
188. Shimon Even and Robert E. Tarjan. Network flow and testing graph connectivity. *SIAM Journal on Computing*, 4(4):507–518, December 1975.
189. Martin G. Everett. Graph theoretic blockings k-plexes and k-cutpoints. *Journal of Mathematical Sociology*, 9:75–84, 1982.

190. Martin G. Everett and Stephen P. Borgatti. Role colouring a graph. *Mathematical Social Sciences*, 21:183–188, 1991.
191. Martin G. Everett and Stephen P. Borgatti. Regular equivalence: General theory. *Journal of Mathematical Sociology*, 18(1):29–52, 1994.
192. Martin G. Everett and Stephen P. Borgatti. Analyzing clique overlap. *Connections*, 21(1):49–61, 1998.
193. Martin G. Everett and Stephen P. Borgatti. Peripheries of cohesive subsets. *Social Networks*, 21(4):397–407, 1999.
194. Martin G. Everett and Stephen P. Borgatti. Extending centrality. In Peter J. Carrington, John Scott, and Stanley Wasserman, editors, *Models and Methods in Social Network Analysis*. Cambridge University Press, 2005. To appear.
195. Martin G. Everett, Philip Sinclair, and Peter Dankelmann. Some centrality results new and old. Submitted, 2004.
196. Alex Fabrikant, Elias Koutsoupias, and Christos H. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the Internet. In *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP'02)*, volume 2380 of *Lecture Notes in Computer Science*, 2002.
197. Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *Proceedings of SIGCOMM'99*, 1999.
198. Illés Farkas, Imre Derényi, Albert-László Barabási, and Tamás Vicsek. Spectra of "real-world" graphs: Beyond the semicircle law. *Physical Review E*, 64, August 2001.
199. Katherine Faust. Comparison of methods for positional analysis: Structural and general equivalences. *Social Networks*, 10:313–341, 1988.
200. Katherine Faust and John Skvoretz. Logit models for affiliation networks. *Sociological Methodology*, 29(1):253–280, 1999.
201. Uriel Feige, Guy Kortsarz, and David Peleg. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.
202. Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Randoms Structures and Algorithms*, 16(2):195–208, 2000.
203. Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM Journal on Computing*, 31(4):1090–1118, 2002.
204. Uriel Feige and Michael A. Seltser. On the densest k-subgraph problem. Technical Report CS97-16, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, Israel, 1997.
205. Trevor Fenner, Mark Levene, and George Loizou. A stochastic evolutionary model exhibiting power-law behaviour with an exponential cutoff. http://xxx.sissa.it/ps/cond-mat/0209463, June 2004.
206. Anuška Ferligoj, Patrick Doreian, and Vladimir Batagelj. Optimizational approach to blockmodeling. *Journal of Computing and Information Technology*, 4:63–90, 1996.
207. Jean-Claude Fernandez. An implementation of an efficient algorithm for bisimulation equivalence. *Science of Computer Programming*, 13(1):219–236, 1989.
208. William L. Ferrar. *Finite Matrices*. Oxford University Press, London, 1951.
209. Jiří Fiala and Daniël Paulusma. The computational complexity of the role assignment problem. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming (ICALP'03)*, pages 817–828. Springer-Verlag, 2003.
210. Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):289–305, 1973.
211. Miroslav Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 1:619–633, 1975.

212. Stephen E. Fienberg and Stanley Wasserman. Categorical data analysis of a single sociometric relation. In Samuel Leinhardt, editor, *Sociological Methodology*, pages 156–192. Jossey Bass, 1981.
213. Stephen E. Fienberg and Stanley Wasserman. Comment on an exponential family of probability distributions. *Journal of the American Statistical Association*, 76(373):54–57, March 1981.
214. Philippe Flajolet, Kostas P. Hatzis, Sotiris Nikoletseas, and Paul Spirakis. On the robustness of interconnections in random graphs: A symbolic approach. *Theoretical Computer Science*, 287(2):515–534, September 2002.
215. Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.
216. Lisa Fleischer. Building chain and cactus representations of all minimum cuts from Hao-Orlin in the same asymptotic run time. *Journal of Algorithms*, 33(1):51–72, October 1999.
217. Robert W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.
218. Lester R. Ford, Jr. and Delbert R. Fulkerson. Maximal fbw through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.
219. Lester R. Ford, Jr. and Delbert R. Fulkerson. A simple algorithm for finding maximal network fbws and an application to the Hitchcock problem. *Canadian Journal of Mathematics*, 9:210–218, 1957.
220. Lester R. Ford, Jr. and Delbert R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
221. Scott Fortin. The graph isomorphism problem. Technical Report 96-20, University of Alberta, Edmonton, Canada, 1996.
222. Ove Frank and David Strauss. Markov graphs. *Journal of the American Statistical Association*, 81:832–842, 1986.
223. Greg N. Frederickson. Ambivalent data structures for dynamic 2-edge-connectivity and k smallest spanning trees. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'91)*, pages 632–641, October 1991.
224. Michael L. Fredman. New bounds on the complexity of the shortest path problem. *SIAM Journal on Computing*, 5:49–60, 1975.
225. Michael L. Fredman and Dan E. Willard. Trans-dichotomous algorithms for minimum spanning trees and shortest paths. *Journal of Computer and System Sciences*, 48(3):533–551, 1994.
226. Linton Clarke Freeman. A set of measures of centrality based upon betweeness. *Sociometry*, 40:35–41, 1977.
227. Linton Clarke Freeman. Centrality in social networks: Conceptual clarification I. *Social Networks*, 1:215–239, 1979.
228. Linton Clarke Freeman. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Booksurge Publishing, 2004.
229. Linton Clarke Freeman, Stephen P. Borgatti, and Douglas R. White. Centrality in valued graphs: A measure of betweenness based on network fbw. *Social Networks*, 13(2):141–154, 1991.
230. Noah E. Friedkin. Structural cohesion and equivalence explanations of social homogeneity. *Sociological Methods and Research*, 12:235–261, 1984.
231. Delbert R. Fulkerson and George B. Dantzig. Computation of maximal fbws in networks. *Naval Research Logistics Quarterly*, 2:277–283, 1955.
232. Delbert R. Fulkerson and G. C. Harding. On edge-disjoint branchings. *Networks*, 6(2):97–104, 1976.
233. Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

234. Harold N. Gabow. Scaling algorithms for network problems. *Journal of Computer and System Sciences*, 31(2):148–168, 1985.
235. Harold N. Gabow. Path-based depth-first search for strong and biconnected components. *Information Processing Letters*, 74:107–114, 2000.
236. Zvi Galil. An $\mathcal{O}(V^{5/3}E^{2/3})$ algorithm for the maximal flow problem. *Acta Informatica*, 14:221–242, 1980.
237. Zvi Galil and Giuseppe F. Italiano. Fully dynamic algorithms for edge connectivity problems. In *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing (STOC'91)*, pages 317–327, May 1991.
238. Zvi Galil and Amnon Naamad. An $\mathcal{O}(EV \log^2 V)$ algorithm for the maximal flow problem. *Journal of Computer and System Sciences*, 21(2):203–217, October 1980.
239. Giorgio Gallo, Michail D. Grigoriadis, and Robert E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.
240. Michael R. Garey and David S. Johnson. *Computers and Intractability. A Guide to the Theory of $\mathcal{NP}$-Completeness.* W. H. Freeman and Company, 1979.
241. Michael R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified $\mathcal{NP}$-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.
242. Christian Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(3):393–408, 1998.
243. Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis.* Chapman & Hall Texts in Statistical Science. Chapman & Hall/CRC, 2nd edition, June 1995.
244. Horst Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.
245. Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter. *Markov Chain Monte Carlo in Practice.* Interdisciplinary Statistics. Chapman & Hall/CRC, 1996.
246. Chris Godsil. Tools from linear algebra. Research report, University of Waterloo, 1989.
247. Chris Godsil and Gordon Royle. *Algebraic Graph Theory.* Graduate Texts in Mathematics. Springer-Verlag, 2001.
248. Andrew V. Goldberg. Finding a maximum density subgraph. Technical Report UCB/CSB/ 84/171, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1984.
249. Andrew V. Goldberg. A new max-flow algorithm. Technical Memo MIT/LCS/TM-291, MIT Laboratory for Computer Science, November 1985.
250. Andrew V. Goldberg and Satish Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45(5):783–797, 1998.
251. Andrew V. Goldberg and Satish Rao. Flows in undirected unit capacity networks. *SIAM Journal on Discrete Mathematics*, 12(1):1–5, 1999.
252. Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, 1988.
253. Andrew V. Goldberg and Kostas Tsioutsiouliklis. Cut tree algorithms: An experimental study. *Journal of Algorithms*, 38(1):51–83, 2001.
254. Donald L. Goldsmith. On the second order edge connectivity of a graph. *Congressus Numerantium*, 29:479–484, 1980.
255. Donald L. Goldsmith. On the n-th order edge-connectivity of a graph. *Congressus Numerantium*, 32:375–381, 1981.
256. Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* John Hopkins University Press, 3rd edition, 1996.

257. Ralph E. Gomory and T.C. Hu. Multi-terminal network flows. *Journal of SIAM*, 9(4):551–570, December 1961.
258. Ralph E. Gomory and T.C. Hu. Synthesis of a communication network. *Journal of SIAM*, 12(2):348–369, 1964.
259. Ramesh Govindan and Anoop Reddy. An analysis of Internet inter-domain topology and route stability. In *Proceedings of Infocom'97*, 1997.
260. Fabrizio Grandoni and Giuseppe F. Italiano. Decremental clique problem. In *Proceedings of the 30th International Workshop on Graph-Theoretical Conecpts in Computer Science (WG'04)*, Lecture Notes in Computer Science. Springer-Verlag, 2004. To appear.
261. George Grätzer. *General Lattice Theory*. Birkhäuser Verlag, 1998.
262. Jerrold R. Griggs, Miklós Simonovits, and George Rubin Thomas. Extremal graphs with bounded densities of small subgraphs. *Journal of Graph Theory*, 29(3):185–207, 1998.
263. Geoffrey Grimmett and Colin J. H. McDiarmid. On colouring random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 77:313–324, 1975.
264. Dan Gusfield. Connectivity and edge-disjoint spanning trees. *Information Processing Letters*, 16(2):87–89, 1983.
265. Dan Gusfield. Very simple methods for all pairs network flow analysis. *SIAM Journal on Computing*, 19(1):143–155, 1990.
266. Ronald J. Gutman. Reach-based routing: A new approach to shortest path algorithms optimized for road networks. In *Proceedings of the 6th Workshop on Algorithm Engineering and Experiments (ALENEX'04)*, Lecture Notes in Computer Science, pages 100–111. SIAM, 2004.
267. Carsten Gutwenger and Petra Mutzel. A linear time implementation of SPQR-trees. In *Proceedings of the 8th International Symposium on Graph Drawing (GD'00)*, volume 1984 of *Lecture Notes in Computer Science*, pages 70–90, January 2001.
268. Willem H. Haemers. Eigenvalue methods. In Alexander Schrijver, editor, *Packing and Covering in Combinatorics*, pages 15–38. Mathematisch Centrum, 1979.
269. Per Hage and Frank Harary. *Structural models in anthropology*. Cambridge University Press, 1st edition, 1983.
270. S. Louis Hakimi. On the realizability of a set of integers as degrees of the vertices of a linear graph. *SIAM Journal on Applied Mathematics*, 10:496–506, 1962.
271. S. Louis Hakimi. Optimum location of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459, 1964.
272. Jianxiu Hao and James B. Orlin. A faster algorithm for finding the minimum cut in a graph. In *Proceedings of the 3rd Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'92)*, pages 165–174, January 1992.
273. Frank Harary. Status and contrastatus. *Sociometry*, 22:23–43, 1959.
274. Frank Harary. The maximum connectivity of a graph. *Proceedings of the National Academy of Science of the United States of America*, 48(7):1142–1146, July 1962.
275. Frank Harary. A characterization of block-graphs. *Canadian Mathematical Bulletin*, 6(1):1–6, January 1963.
276. Frank Harary. Conditional connectivity. *Networks*, 13:347–357, 1983.
277. Frank Harary. General connectivity. In Khee Meng Koh and Hian-Poh Yap, editors, *Proceedings of the 1st Southeast Asian Graph Theory Colloquium*, volume 1073 of *Lecture Notes in Mathematics*, pages 83–92. Springer-Verlag, 1984.
278. Frank Harary and Per Hage. Eccentricity and centrality in networks. *Social Networks*, 17:57–63, 1995.
279. Frank Harary and Yukihiro Kodama. On the genus of an n-connected graph. *Fundamenta Mathematicae*, 54:7–13, 1964.

280. Frank Harary and Helene J. Kommel. Matrix measures for transitivity and balance. *Journal of Mathematical Sociology*, 6:199–210, 1979.
281. Frank Harary and Robert Z. Norman. The dissimilarity characteristic of Husimi trees. *Annals of Mathematics*, 58(2):134–141, 1953.
282. Frank Harary and Herbert H. Paper. Toward a general calculus of phonemic distribution. *Language: Journal of the Linguistic Society of America*, 33:143–169, 1957.
283. Frank Harary and Geert Prins. The block-cutpoint-tree of a graph. *Publicationes Mathematicae Debrecen*, 13:103–107, 1966.
284. Frank Harary and Ian C. Ross. A procedure for clique detection using the group matrix. *Sociometry*, 20:205–215, 1957.
285. David Harel and Yehuda Koren. On clustering using random walks. In *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'01)*, volume 2245 of *Lecture Notes in Computer Science*, pages 18–41. Springer-Verlag, 2001.
286. Erez Hartuv and Ron Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4-6):175–181, 2000.
287. Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.
288. Vaclav Havel. A remark on the existence of finite graphs (in czech). *Casopis Pest. Math.*, 80:477–480, 1955.
289. Taher H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.
290. Taher H. Haveliwala and Sepandar D. Kamvar. The second eigenvalue of the Google matrix. Technical report, Stanford University, March 2003.
291. Taher H. Haveliwala, Sepandar D. Kamvar, and Glen Jeh. An analytical comparison of approaches to personalized PageRank. Technical report, Stanford University, June 2003.
292. Taher H. Haveliwala, Sepandar D. Kamvar, Dan Klein, Christopher D. Manning, and Gene H. Golub. Computing PageRank using power extrapolation. Technical report, Stanford University, July 2003.
293. George R. T. Hendry. On graphs with a prescribed median. I. *Journal of Graph Theory*, 9:477–487, 1985.
294. Michael A. Henning and Ortrud R. Oellermann. The average connectivity of a digraph. *Discrete Applied Mathematics*, 140:143–153, May 2004.
295. Monika R. Henzinger and Michael L. Fredman. Lower bounds for fully dynamic connectivity problems in graphs. *Algorithmica*, 22(3):351–362, 1998.
296. Monika R. Henzinger and Valerie King. Fully dynamic 2-edge connectivity algorithm in polylogarithmic time per operation. SRC Technical Note 1997-004a, Digital Equipment Corporation, Systems Research Center, Palo Alto, California, June 1997.
297. Monika R. Henzinger and Johannes A. La Poutré. Certificates and fast algorithms for biconnectivity in fully-dynamic graphs. SRC Technical Note 1997-021, Digital Equipment Corporation, Systems Research Center, Palo Alto, California, September 1997.
298. Monika R. Henzinger, Satish Rao, and Harold N. Gabow. Computing vertex connectivity: New bounds from old techniques. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS'96)*, pages 462–471, October 1996.
299. Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):713–721, 1963.

300. Karen S. Holbert. A note on graphs with distant center and median. In V. R. Kulli, editor, *Recent Sudies in Graph Theory*, pages 155–158, Gulbarza, India, 1989. Vishwa International Publications.
301. Paul W. Holland, Kathryn B. Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5:109–137, 1983.
302. Paul W. Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50, March 1981.
303. Jacob Holm, Kristian de Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM*, 48(4):723–760, 2001.
304. Petter Holme. Congestion and centrality in traffic fbw on complex networks. *Advances in Complex Systems*, 6(2):163–176, 2003.
305. Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulnerability of complex networks. *Physical Review E*, 65(056109), 2002.
306. Klaus Holzapfel. *Density-based clustering in large-scale networks*. PhD thesis, Technische Universität München, 2004.
307. Klaus Holzapfel, Sven Kosub, Moritz G. Maaß, Alexander Offtermatt-Souza, and Hanjo Täubig. A zero-one law for densities of higher order. Manuscript, 2004.
308. Klaus Holzapfel, Sven Kosub, Moritz G. Maaß, and Hanjo Täubig. The complexity of detecting fixed-density clusters. In *Proceedings of the 5th Italian Conference on Algorithms and Complexity (CIAC'03)*, volume 2653 of *Lecture Notes in Computer Science*, pages 201–212. Springer-Verlag, 2003.
309. John E. Hopcroft and Robert E. Tarjan. Finding the triconnected components of a graph. Technical Report TR 72-140, CS Dept., Cornell University, Ithaca, N.Y., August 1972.
310. John E. Hopcroft and Robert E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158, September 1973.
311. John E. Hopcroft and Robert E. Tarjan. Efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, June 1973.
312. John E. Hopcroft and Robert E. Tarjan. Dividing a graph into triconnected components. Technical Report TR 74-197, CS Dept., Cornell University, Ithaca, N.Y., February 1974.
313. John E. Hopcroft and Robert E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, October 1974.
314. John E. Hopcroft and J.K. Wong. A linear time algorithm for isomorphism of planar graphs. In *Proceedings of the 6th Annual ACM Symposium on the Theory of Computing (STOC'74)*, pages 172–184, 1974.
315. Radu Horaud and Thomas Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180, 1989.
316. Wen-Lian Hsu. $\mathcal{O}$(MN) algorithms for the recognition and isomorphism problems on circular-arc graphs. *SIAM Journal on Computing*, 24:411–439, 1995.
317. T.C. Hu. Optimum communication spanning trees. *SIAM Journal on Computing*, 3:188–195, 1974.
318. Xiaohan Huang and Victor Y. Pan. Fast rectangular matrix multiplication and applications. *Journal of Complexity*, 14(2):257–299, 1998.
319. Charles H. Hubbell. In input-output approach to clique identification. *Sociometry*, 28:377–399, 1965.
320. Piotr Indyk and Jiří Matoušek. Low-distortion embeddings of finite metric spaces. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, April 2004.
321. Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7(4):413–423, 1978.

322. Kenneth E. Iverson. *A Programming Language.* Wiley, 1962.
323. Matthew O. Jackson and Asher Wolinsky. A strategic model of social and economic networks. *Journal of Economic Theory*, 71:474–486, 1996.
324. Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data.* Prentice Hall, 1988.
325. Anil K. Jain, M. N. Murty, and Patrick J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
326. Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th International World Wide Web Conference (WWW12)*, pages 271–279, Budapest, Hungary, 2003.
327. Hawoong Jeong, Sean P. Mason, Albert-László Barabási, and Zoltan N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411, 2001. Brief communications.
328. Mark Jerrum. Large cliques elude the Metropolis process. *Randoms Structures and Algorithms*, 3(4):347–359, 1992.
329. Mark Jerrum and Alistair Sinclair. Fast uniform generation of regular graphs. *Theoretical Computer Science*, 73:91–100, 1990.
330. Tang Jian. An $O(2^{0.304n})$ algorithm for solving maximum independent set problem. *IEEE Transactions on Computers*, C-35(9):847–851, 1986.
331. Bin Jiang. I/O- and CPU-optimal recognition of strongly connected components. *Information Processing Letters*, 45(3):111–115, March 1993.
332. Cheng Jin, Qian Chen, and Sugih Jamin. Inet topology generator. Technical Report CSE-TR-433-00, EECS Department, University of Michigan, 2000.
333. David S. Johnson, Jan Karel Lenstra, and Alexander H. G. Rinnooy Kan. The complexity of the network design problem. *Networks*, 9:279–285, 1978.
334. David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.
335. Ian T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, 2002.
336. Camille Jordan. Sur les assemblages de lignes. *Journal für reine und angewandte Mathematik*, 70:185–190, 1869.
337. Ferenc Juhász. On the spectrum of a random graph. *Colloquia Mathematica Societatis János Bolyai*, 25:313–316, 1978.
338. Sepandar D. Kamvar, Taher H. Haveliwala, and Gene H. Golub. Adaptive methods for the computation of PageRank. Technical report, Stanford University, April 2003.
339. Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Exploiting the block structure of the web for computing PageRank. Technical report, Stanford University, March 2003.
340. Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Extrapolation methods for accelerating PageRank computations. In *Proceedings of the 12th International World Wide Web Conference (WWW12)*, pages 261–270, Budapest, Hungary, 2003.
341. Arkady Kanevsky, Roberto Tamassia, Guiseppe Di Battista, and Jianer Chen. On-line maintenance of the four-connected components of a graph. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'91)*, pages 793–801, October 1991.
342. Ravi Kannan and V. Vinay. Analyzing the structure of large graphs. Manuscript, 1999.
343. David R. Karger and Matthew S. Levine. Finding maximum flows in undirected graphs seems easier than bipartite matching. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC'98)*, pages 69–78, May 1998.

344. David R. Karger and Clifford Stein. An $\tilde{O}(n^2)$ algorithm for minimum cuts. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing (STOC'93)*, pages 757–765, May 1993.
345. Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
346. Richard M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.
347. Richard M. Karp. Probabilistic analysis of some combinatorial search problems. In Joseph F. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 1–19. Academic Press, 1976.
348. George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.
349. Alexander V. Karzanov. On finding maximum flows in networks with special structure and some applications. In *Matematicheskie Voprosy Upravleniya Proizvodstvom*, volume 5, pages 66–70. Moscow State University Press, 1973. (In Russian).
350. Alexander V. Karzanov. Determining the maximal flow in a network by the method of preflows. *Soviet Mathematics-Doklady*, 15(2):434–437, 1974.
351. Alexander V. Karzanov and Eugeniy A. Timofeev. Efficient algorithm for finding all minimal edge cuts of a nonoriented graph. *Cybernetics*, 22(2):156–162, 1986.
352. Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
353. Subhash Khot. Improved approximation algorithms for max clique, chromatic number and approximate graph coloring. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'01)*, pages 600–609. IEEE Computer Society Press, 2001.
354. Subhash Khot. Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 136–145. IEEE Computer Society Press, 2004.
355. K. H. Kim and F. W. Roush. Group relationships and homomorphisms of boolean matrix semigroups. *Journal of Mathematical Psychology*, 28:448–452, 1984.
356. Valerie King, Satish Rao, and Robert E. Tarjan. A faster deterministic maximum flow algorithm. In *Proceedings of the 3rd Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'92)*, pages 157–164, January 1992.
357. G. Kishi. On centrality functions of a graph. In N. Saito and T. Nishizeki, editors, *Proceedings of the 17th Symposium of Research Institute of Electrical Communication on Graph Theory and Algorithms*, volume 108 of *Lecture Notes in Computer Science*, pages 45–52, Sendai, Japan, October 1980. Springer.
358. G. Kishi and M. Takeuchi. On centrality functions of a non-directed graph. In *Proceedings of the 6th Colloquium on Microwave Communication*, Budapest, 1978.
359. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
360. Jon M. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing (STOC'00)*, May 2000.
361. Jon M. Kleinberg. An impossibility theorem for clustering. In *Proceedings of 15th Conference: Neiral Information Processing Systems, Advances in Neural Information Processing Systems*, 2002.
362. Daniel J. Kleitman. Methods for investigating connectivity of large graphs. *IEEE Transactions on Circuit Theory*, 16(2):232–233, May 1969.

363. Ton Kloks, Dieter Kratsch, and Haiko Müller. Finding and counting small induced subgraphs efficiently. *Information Processing Letters*, 74(3–4):115–121, 2000.
364. David Knoke and David L. Rogers. A blockmodel analysis of interorganizational networks. *Sociology and Social Research*, 64:28–52, 1979.
365. Donald E. Knuth. Two notes on notation. *American Mathematical Monthly*, 99:403–422, 1990.
366. Dénes Kőnig. Graphen und Matrizen. *Mat. Fiz. Lapok*, 38:116–119, 1931.
367. Avrachenkov Konstantin and Nelly Litvak. Decomposition of the Google PageRank and optimal linking strategy. Technical Report 5101, INRIA, Sophia Antipolis, France, January 2004.
368. Tamás Kővári, Vera T. Sós, and Pál Turán. On a problem of Zarankiewicz. *Colloquium Mathematicum*, 3:50–57, 1954.
369. Paul L. Krapivsky, Sidney Redner, and Francois Leyvraz. Connectivity of growing random networks. http://xxx.sissa.it/ps/cond-mat/0005139, September 2000.
370. Jan Kratochvíl. *Perfect Codes in General Graphs*. Academia Praha, 1991.
371. V. Krishnamoorthy, K. Thulasiraman, and M. N. S. Swamy. Incremental distance and diameter sequences of a graph: New measures of network performance. *IEEE Transactions on Computers*, 39(2):230–237, February 1990.
372. Joseph B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonparametric hypothesis. *Psychometrika*, 29:1–27, March 1964.
373. Joseph B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, June 1964.
374. Luděk Kučera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2–3):193–212, 1995.
375. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew S. Tomkins, and Eli Upfal. Stochastic models for the web graph. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS'00)*, 2000.
376. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 31(11–16):1481–1493, 1999.
377. Johannes A. La Poutré, Jan van Leeuwen, and Mark H. Overmars. Maintenance of 2- and 3-connected components of graphs, Part I: 2- and 3-edge-connected components. Technical Report RUU-CS-90-26, Dept. of Computer Science, Utrecht University, July 1990.
378. Amy N. Langville and Carl D. Meyer. Deeper inside PageRank. Technical report, Department of Mathematics, North Carolina State University, Raleigh, NC, USA, March 2004. accepted by *Internet Mathematics*.
379. Amy N. Langville and Carl D. Meyer. A survey of eigenvector methods of web information retrieval. Technical report, Department of Mathematics, North Carolina State University, Raleigh, NC, USA, January 2004. accepted by *The SIAM Review*.
380. Luigi Laura, Stefano Leonardi, Stefano Millozzi, and Ulrich Meyer. Algorithms and experiments for the webgraph. In *Proceedings of the 11th Annual European Symposium on Algorithms (ESA'03)*, volume 2832 of *Lecture Notes in Computer Science*, 2003.
381. Eugene L. Lawler. Cutsets and partitions of hypergraphs. *Networks*, 3:275–285, 1973.
382. Eugene L. Lawler, Jan Karel Lenstra, and Alexander H. G. Rinnooy Kan. Generating all maximal independent sets: $\mathcal{NP}$-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9(3):558–565, 1980.

383. Chris Pan-Chi Lee, Gene H. Golub, and Stefanos A. Zenios. A fast two-stage algorithm for computing PageRank. Technical Report SCCM-03-15, Stanford University, 2003.
384. Erich L. Lehmann. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer-Verlag, 2nd edition, 1997.
385. Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer Texts in Statistics. Springer-Verlag, 2nd edition, 1998.
386. L. Ya. Leifman. An efficient algorithm for partitioning an oriented graph into bicomponents. *Cybernetics*, 2(5):15–18, 1966.
387. Ronny Lempel and Shlomo Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33:387–401, 2000. volume coincides with the Proceedings of the 9th international World Wide Web conference on Computer networks.
388. Ronny Lempel and Shlomo Moran. Rank-stability and rank-similarity of link-based web ranking algorithms in authority-connected graphs. *Information Retrieval, special issue on Advances in Mathematics/Formal Methods in Information Retrieval*, 2004. in press.
389. Thomas Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. Wiley, 1990.
390. Linda Lesniak. Results on the edge-connectivity of graphs. *Discrete Mathematics*, 8:351–354, 1974.
391. Robert Levinson. Pattern associativity and the retrieval of semantic networks. *Computers & Mathematics with Applications*, 23(2):573–600, 1992.
392. Nathan Linial, László Lovász, and Avi Wigderson. A physical interpretation of graph connectivity and its algorithmic applications. In *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science (FOCS'86)*, pages 39–48, October 1986.
393. Nathan Linial, László Lovász, and Avi Wigderson. Rubber bands, convex embeddings and graph connectivity. *Combinatorica*, 8(1):91–102, 1988.
394. François Lorrain and Harrison C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1:49–80, 1971.
395. Emmanuel Loukakis and Konstantinos-Klaudius Tsouros. A depth first search algorithm to generate the family of maximal independent sets of a graph lexicographically. *Computing*, 27:249–266, 1981.
396. László Lovász. Connectivity in digraphs. *Journal of Combinatorial Theory Series B*, 15(2):174–177, August 1973.
397. László Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.
398. Anna Lubiw. Some $\mathcal{NP}$-complete problems similar to graph isomorphism. *SIAM Journal on Computing*, 10:11–24, 1981.
399. Fabrizio Luccio and Mariagiovanna Sami. On the decomposition of networks in minimally interconnected subnetworks. *IEEE Transactions on Circuit Theory*, CT-16:184–188, 1969.
400. R. Duncan Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15:169–190, 1950.
401. R. Duncan Luce and Albert Perry. A method of matrix analysis of group structure. *Psychometrika*, 14:95–116, 1949.
402. Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25:42–65, 1982.
403. Saunders Mac Lane. A structural characterization of planar combinatorial graphs. *Duke Mathematical Journal*, 3:460–472, 1937.
404. Wolfgang Mader. Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen. *Archiv der Mathematik*, 23:219–224, 1972.

405. Damien Magoni and Jean Jacques Pansiot. Analysis of the autonomous system network topology. *Computer Communication Review*, 31(3):26–37, July 2001.
406. Vishv M. Malhotra, M. Pramodh Kumar, and S. N. Maheshwari. An $\mathcal{O}(|V|^3)$ algorithm for finding maximum flows in networks. *Information Processing Letters*, 7(6):277–278, October 1978.
407. Yishay Mansour and Baruch Schieber. Finding the edge connectivity of directed graphs. *Journal of Algorithms*, 10(1):76–85, March 1989.
408. Maarten Marx and Michael Masuch. Regular equivalence and dynamic logic. *Social Networks*, 25:51–65, 2003.
409. Rudi Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–132, 1979.
410. David W. Matula. The cohesive strength of graphs. In *The Many Facets of Graph Theory, Proc.*, volume 110 of *Lecture Notes in Mathematics*, pages 215–221. Springer-Verlag, 1969.
411. David W. Matula. k-components, clusters, and slicings in graphs. *SIAM Journal on Applied Mathematics*, 22(3):459–480, May 1972.
412. David W. Matula. Graph theoretic techniques for cluster analysis algorithms. In J. Van Ryzin, editor, *Classification and clustering*, pages 95–129. Academic Press, 1977.
413. David W. Matula. Determining edge connectivity in $\mathcal{O}(nm)$. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science (FOCS'87)*, pages 249–251, October 1987.
414. James J. McGregor. Backtrack search algorithms and the maximal common subgraph problem. *Software - Practice and Experience*, 12(1):23–24, 1982.
415. Brendan D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.
416. Brendan D. McKay and Nicholas C. Wormald. Uniform generation of random regular graphs of moderate degree. *Journal of Algorithms*, 11:52–67, 1990.
417. Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. BRITE: An approach to universal topology generation. In *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'01)*, 2001.
418. Alberto Medina, Ibrahim Matta, and John Byers. On the origin of power laws in Internet topologies. *Computer Communication Review*, 30(2), April 2000.
419. Karl Menger. Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.
420. Milena Mihail, Christos Gkantsidis, Amin Saberi, and Ellen W. Zegura. On the semantics of internet topologies. Technical Report GIT-CC-02-07, Georgia Institute of Technology, 2002.
421. Stanley Milgram. The small world problem. *Psychology Today*, 1:61, 1967.
422. Gary L. Miller and Vijaya Ramachandran. A new graph triconnectivity algorithm and its parallelization. *Combinatorica*, 12(1):53–76, 1992.
423. Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, October 2002.
424. J. Clyde Mitchell. Algorithms and network analysis: A test of some analytical procedures on Kapferer's tailor shop material. In Linton Clarke Freeman, Douglas R. White, and A. Kimbal Romney, editors, *Research Methods in Social Network Analysis*, pages 365–391. George Mason University Press, 1989.
425. Bojan Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory Series B*, 47(3):274–291, 1989.
426. Bojan Mohar. Eigenvalues, diameter and mean distance in graphs. *Graphs and Combinatorics*, 7:53–64, 1991.

427. Bojan Mohar. The laplacian spectrum of graphs. In Yousef Alavi, Gary Chartrand, Ortrud R. Oellermann, and Allen J. Schwenk, editors, *Graph Theory, Combinatorics, and Applications*, pages 871–898. Wiley, 1991.
428. Bojan Mohar and Svatopluk Poljak. Eigenvalues in combinatorial optimization. In Richard A. Brualdi, Shmuel Friedland, and Victor Klee, editors, *Combinatorial and Graph-Theoretical Problems in Linear Algebra*, pages 107–151. Springer-Verlag, 1993.
429. Robert J. Mokken. Cliques, clubs, and clans. *Quality and Quantity*, 13:161–173, 1979.
430. Burkhard Möller. Zentralitäten in Graphen. Diplomarbeit, Fachbereich Informatik und Informationswissenschaft, Universität Konstanz, July 2002.
431. John W. Moon. On the diameter of a graph. *Michigan Mathematical Journal*, 12(3):349–351, 1965.
432. John W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965.
433. Robert L. Moxley and Nancy F. Moxley. Determining Point-Centrality in Uncontrived Social Networks. *Sociometry*, 37:122–130, 1974.
434. N. C. Mullins, L. L. Hargens, P. K. Hecht, and Edward L. Kick. The group structure of cocitation clusters: A comparative study. *American Sociological Review*, 42:552–562, 1977.
435. Ian Munro. Efficient determination of the transitive closure of a directed graph. *Information Processing Letters*, 1(2):56–58, 1971.
436. Siegfried F. Nadel. *The Theory of Social Structure*. Cohen & West LTD, 1957.
437. Kai Nagel. Traffic networks. In Stefan Bornholdt and Heinz Georg Schuster, editors, *Handbook of Graphs and Networks: From the Genome to the Internet*. Wiley-VCH, 2002.
438. Walid Najjar and Jean-Luc Gaudiot. Network resilience: A measure of network fault tolerance. *IEEE Transactions on Computers*, 39(2):174–181, February 1990.
439. Georg L. Nemhause and Laurence A. Wolesy. *Integer and Combinatorial Optimization*. Wiley, 1988.
440. Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 26(2):415–419, 1985.
441. Mark E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(208701), 2002.
442. Mark E. J. Newman. Fast algorithm for detecting community structure in networks. arXiv cond-mat/0309508, September 2003.
443. Mark E. J. Newman. A measure of betweenness centrality based on random walks. arXiv cond-mat/0309045, 2003.
444. Mark E. J. Newman and Michelle Girvan. Mixing patterns and community structure in networks. In Romualdo Pastor-Satorras, Miguel Rubi, and Albert Diaz-Guilera, editors, *Statistical Mechanics of Complex Networks*, volume 625 of *Lecture Notes in Physics*, pages 66–87. Springer-Verlag, 2003.
445. Mark E. J. Newman and Michelle Girvan. Findind and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.
446. Mark E. J. Newman and Juyong Park. Why social networks are different from other types of networks. *Physical Review E*, 68(036122), 2003.
447. Mark E. J. Newman, Steven H. Strogatz, and Duncan J. Watts. Random graph models of social networks. *Proceedings of the National Academy of Science of the United States of America*, 99:2566–2572, 2002.
448. Mark E. J. Newman, Duncan J. Watts, and Steven H. Strogatz. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:026118, 2001.

449. Andrew Y. Ng, Alice X. Zheng, and Micheal I. Jordan. Link analysis, eigenvectors and stability. In *Proceedings of the senventeenth international joint conference on artificial intelligence*, pages 903–910, Seattle, Washington, 2001.
450. Victor Nicholson, Chun Che Tsai, Marc A. Johnson, and Mary Naim. A subgraph isomorphism theorem for molecular graphs. In *Proceedings of The International Conference on Graph Theory and Topology in Chemistry*, pages 226–230, 1987.
451. U. J. Nieminen. On the Centrality in a Directed Graph. *Social Science Research*, 2:371–378, 1973.
452. National laboratory for applied network research routing data, 1999.
453. Krzysztof Nowicki and Tom A.B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96:1077–1087, 2001.
454. Esko Nuutila and Eljas Soisalon-Soininen. On finding the strongly connected components in a directed graph. *Information Processing Letters*, 49(1):9–14, January 1994.
455. Ortrud R. Oellermann. A note on the ℓ-connectivity function of a graph. *Congressus Numerantium*, 60:181–188, December 1987.
456. Ortrud R. Oellermann. On the l-connectivity of a graph. *Graphs and Combinatorics*, 3:285–291, 1987.
457. Maria G.R. Ortiz, Jose R.C. Hoyos, and Maria G.R. Lopez. The social networks of academic performance in a student context of poverty in mexico. *Social Networks*, 26(2):175–188, 2004.
458. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Manuscript, 1999.
459. Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–983, 1987.
460. Ignacio Palacios-Huerta and Oscar Volij. The measurement of intellectual influence. *Econometrica*, 2004. accepted for publication.
461. Christopher Palmer, Phillip Gibbons, and Christos Faloutsos. Fast approximation of the "neighbourhood" function for massive graphs. Technical Report CMUCS-01-122, Carnegie Mellon Uiversity, 2001.
462. Christopher Palmer, Georgos Siganos, Michalis Faloutsos, Christos Faloutsos, and Phillip Gibbons. The connectivity and fault-tolerance of the Internet topology. In *Workshop on Network-Related Data Management (NRDM 2001)*, 2001.
463. Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal. Using PageRank to characterize Web structure. In *Proceedings of the 8th Annual International Conference on Computing Combinatorics (COCOON'02)*, volume 2387 of *Lecture Notes in Computer Science*, pages 330–339, 2002.
464. Apostolos Papadopoulos and Yannis Manolopoulos. Structure-based similarity search with graph histograms. In *DEXA Workshop*, pages 174–178, 1999.
465. Britta Papendiek and Peter Recht. On maximal entries in the principal eigenvector of graphs. *Linear Algebra and its Applications*, 310:129–138, 2000.
466. Panos M. Pardalos and Jue Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.
467. Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, 1998.
468. Romualdo Pastor-Satorras, Alexei Vázquez, and Alessandro Vespignani. Dynamical and correlation properties of the internet. *Physical Review Letters*, 87(258701), 2001.
469. Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemics and immunization in scale-free networks. In Stefan Bornholdt and Heinz Georg Schuster, editors, *Handbook of Graphs and Networks: From the Genome to the Internet*. Wiley-VCH, 2002.
470. Keith Paton. An algorithm for the blocks and cutnodes of a graph. *Communications of the ACM*, 14(7):468–475, July 1971.

471. Philippa Pattison. *Algebraic Models for Social Networks.* Cambridge University Press, 1993.
472. Philippa Pattison and Stanley Wasserman. Logit models and logistic regressions for social networks: II. Multivariate relations. *British Journal of Mathematical and Statistical Psychology*, 52:169–193, 1999.
473. Marvin C. Paull and Stephen H. Unger. Minimizing the number of states in incompletely specified sequential switching functions. *IRE Transaction on Electronic Computers*, EC-8:356–367, 1959.
474. Aleksandar Pekeč and Fred S. Roberts. The role assignment model nearly fits most social networks. *Mathematical Social Sciences*, 41:275–293, 2001.
475. Claudine Peyrat. Diameter vulnerability of graphs. *Discrete Applied Mathematics*, 9, 1984.
476. Steven Phillips and Jeffery Westbrook. On-line load balancing and network flow. *Algorithmica*, 21(3):245–261, 1998.
477. Jean-Claude Picard and Maurice Queyranne. A network flow solution to some nonlinear 0-1 programming problems, with an application to graph theory. *Networks*, 12:141–159, 1982.
478. Jean-Claude Picard and H. D. Ratliff. Minimum cuts and related problems. *Networks*, 5(4):357–370, 1975.
479. Gabriel Pinski and Francis Narin. Citation influence for journal aggregates of scientific publications: theory, with application to the literature of physics. *Information Processing & Management*, 12:297–312, 1976.
480. André Pönitz and Peter Tittmann. Computing network reliability in graphs of restricted pathwidth. http://www.peter.htwm.de/publications/Reliability.ps, 2001.
481. R. Poulin, M.-C. Boily, and B.R. Mâsse. Dynamical systems to define centrality in social networks. *Social Networks*, 22:187–220, 2000.
482. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.
483. C.H. Proctor and C. P. Loomis. Analysis of sociometric data. In Marie Jahoda, Morton Deutsch, and Stuart W. Cook, editors, *Research Methods in Social Relations*, pages 561–586. Dryden Press, 1951.
484. Paul W. Purdom, Jr. A transitive closure algorithm. Computer Sciences Technical Report #33, University of Wisconsin, July 1968.
485. Paul W. Purdom, Jr. A transitive closure algorithm. *BIT*, 10:76–94, 1970.
486. Pavlin Radoslavov, Hongsuda Tangmunarunkit, Haobo Yu, Ramesh Govindan, Scott Shenker, and Deborah Estrin. On characterizing network topologies and analyzing their impact on protocol design. Technical Report 00-731, Computer Science Department, University of Southern California, February 2000.
487. Rajeev Raman. Recent results on the single-source shortest paths problem. *ACM SIGACT News*, 28(2):81–87, 1997.
488. John W. Raymond, Eleanor J. Gardiner, and Peter Willet. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6):631–644, 2002.
489. Ronald C. Read and Derek G. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1:339–363, 1977.
490. John H. Reif. A topological approach to dynamic graph connectivity. *Information Processing Letters*, 25(1):65–70, 1987.
491. Franz Rendl and Henry Wolkowicz. A projection technique for partitioning the nodes of a graph. *Annals of Operations Research*, 58:155–180, 1995.
492. John A. Rice. *Mathematical Statistics and Data Analysis.* Duxbury Press, 2nd edition, 1995.
493. Fred S. Roberts and Li Sheng. $\mathcal{NP}$-completeness for 2-role assignability. Technical Report 8, Rutgers Center for Operation Research, 1997.

494. Garry Robins, Philippa Pattison, and Stanley Wasserman. Logit models and logistic regressions for social networks: III. Valued relations. *Psychometrika*, 64:371–394, 1999.
495. John Michael Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.
496. Liam Roditty and Uri Zwick. On dynamic shortest paths problems. In *Proceedings of the 12th Annual European Symposium on Algorithms (ESA'04)*, volume 3221 of *Lecture Notes in Computer Science*, pages 580–591, 2004.
497. Arnon S. Rosenthal. *Computing Reliability of Complex Systems*. PhD thesis, University of California, 1974.
498. Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, 8th edition, 2003.
499. Britta Ruhnau. Eigenvector-centrality – a node-centrality? *Social Networks*, 22:357–365, 2000.
500. Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31:581–603, 1966.
501. Lee Douglas Sailer. Structural equivalence: Meaning and definition, computation and application. *Social Networks*, 1:73–90, 1978.
502. Thomas Schank and Dorothea Wagner. Approximating clustering-coefficient and transitivity. Technical Report 2004-9, Universität Karlsruhe, Fakultät für Informatik, 2004.
503. Claus P. Schnorr. Bottlenecks and edge connectivity in unsymmetrical networks. *SIAM Journal on Computing*, 8(2):265–274, May 1979.
504. Uwe Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37:312–323, 1988.
505. Alexander Schrijver. *Theory of linear and integer programming*. Wiley, 1986.
506. Alexander Schrijver. Paths and flows—a historical survey. *CWI Quarterly*, 6(3):169–183, September 1993.
507. Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, 2003.
508. Joseph E. Schwartz. An examination of CONCOR and related methods for blocking sociometric data. In D. R. Heise, editor, *Sociological Methodology 1977*, pages 255–282. Jossey Bass, 1977.
509. Jennifer A. Scott. An Arnoldi code for computing selected eigenvalues of sparse real unsymmetric matrices. *ACM Transactions on Mathematical Software*, 21:423–475, 1995.
510. John R. Seeley. The net of reciprocal influence. *Canadian Journal of Psychology*, III(4):234–240, 1949.
511. Stephen B. Seidman. Clique-like structures in directed networks. *Journal of Social and Biological Structures*, 3:43–54, 1980.
512. Stephen B. Seidman. Internal cohesion of LS sets in graphs. *Social Networks*, 5(2):97–107, 1983.
513. Stephen B. Seidman. Network structure and minimum degree. *Social Networks*, 5:269–287, 1983.
514. Stephen B. Seidman and Brian L. Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978.
515. Stephen B. Seidman and Brian L. Foster. A note on the potential for genuine cross-fertilization between anthropology and mathematics. *Social Networks*, 172:65–72, 1978.
516. Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. In *Graph-Theoretic Concepts in Computer Science, 28th International Workshop, WG 2002*, volume 2573 of *Lecture Notes in Computer Science*, pages 379–390. Springer-Verlag, 2002.

517. Micha Sharir. A strong-connectivity algorithm and its applications in data fbw analysis. *Computers & Mathematics with Applications*, 7(1):67–72, 1981.
518. Yossi Shiloach. An $\mathcal{O}(n \cdot I \log^2 I)$ maximum-fbw algorithm. Technical Report STAN-CS-78-702, Computer Science Department, Stanford University, December 1978.
519. Alfonso Shimbel. Structural parameters of communication networks. *Bulletin of Mathematical Biophysics*, 15:501–507, 1953.
520. F. M. Sim and M. R. Schwartz. Does CONCOR find positions? Unpublished manuscript, 1979.
521. Alistair Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Birkhäuser Verlag, 1993.
522. Brajendra K. Singh and Neelima Gupte. Congestion and Decongestion in a communication network. arXiv cond-mat/0404353, 2004.
523. Mohit Singh and Amitabha Tripathi. Order of a graph with given vertex and edge connectivity and minimum degree. *Electronic Notes in Discrete Mathematics*, 15, 2003.
524. Peter J. Slater. Maximin facility location. *Journal of National Bureau of Standards*, 79B:107–115, 1975.
525. Daniel D. Sleater and Robert E. Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, June 1983.
526. Giora Slutzki and Oscar Volij. Scoring of web pages and tournaments – axiomatizations. Technical report, Iowa State University, Ames, USA, February 2003.
527. Christian Smart and Peter J. Slater. Center, median and centroid subgraphs. *Networks*, 34:303–311, 1999.
528. Peter H. A. Sneath and Robert R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W. H. Freeman and Company, 1973.
529. Tom A.B. Snijders. Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2), April 2002.
530. Tom A.B. Snijders and Krzysztof Nowicki. Estimation and prediction of stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997.
531. Anand Srivastav and Katja Wolf. Finding dense subgraphs with semidefinite programming. In *Proceedings of the 1st International Workshop on Approximatin Algorithms for Combinatorial Optimization (APPROX'98)*, volume 1444 of *Lecture Notes in Computer Science*, pages 181–191. Springer-Verlag, 1998.
532. Angelika Steger and Nicholas C. Wormald. Generating random regular graphs quickly. *Combinatorics, Probability and Computing*, 8:377–396, 1999.
533. Karen A. Stephenson and Marvin Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11:1–37, 1989.
534. Volker Stix. Finding all maximal cliques in dynamic graphs. *Computational Optimization and Applications*, 27(2):173–186, 2004.
535. Josef Stoer and Roland Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, 1993.
536. Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.
537. Sun Microsystems. *Sun Performance Library User's Guide*.
538. Melvin Tainiter. Statistical theory of connectivity I: Basic definitions and properties. *Discrete Mathematics*, 13(4):391–398, 1975.
539. Melvin Tainiter. A new deterministic network reliability measure. *Networks*, 6(3):191–204, 1976.
540. Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Network topologies, power laws, and hierarchy. Technical Report 01-746, Computer Science Department, University of Southern California, 2001.

541. Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Network topologies, power laws, and hierarchy. *ACM SIGCOMM Computer Communication Review*, 32(1):76, 2002.
542. Robert E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, June 1972.
543. Robert E. Tarjan. Finding a maximum clique. Technical Report 72-123, Department of Computer Science, Cornell University, Ithaca, NY, 1972.
544. Robert E. Tarjan. A note on finding the bridges of a graph. *Information Processing Letters*, 2(6):160–161, 1974.
545. Robert E. Tarjan and Anthony E. Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6(3):537–546, 1977.
546. Mikkel Thorup. On RAM priority queues. In *Proceedings of the 7th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'96)*, pages 59–67, 1996.
547. Mikkel Thorup. Undirected single source shortest paths with positive integer weights in linear time. *Journal of the ACM*, 46(3):362–394, 1999.
548. Mikkel Thorup. On ram priority queues. *SIAM Journal on Computing*, 30(1):86–109, 2000.
549. Mikkel Thorup. Fully dynamic all-pairs shortest paths: Faster and allowing negative cycles. In *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT'04)*, volume 3111 of *Lecture Notes in Computer Science*, pages 384–396. Springer-Verlag, 2004.
550. Gottfried Tinhofer. On the generation of random graphs with given properties and known distribution. *Appl. Comput. Sci. Ber. Prakt. Inf.*, 13:265–296, 1979.
551. Po Tong and Eugene L. Lawler. A faster algorithm for finding edge-disjoint branchings. *Information Processing Letters*, 17(2):73–76, August 1983.
552. Miroslaw Truszczyński. Centers and centroids of unicyclic graphs. *Mathematica Slovaca*, 35:223–228, 1985.
553. Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.
554. Pál Turán. On an extremal problem in graph theory. *Matematikai és Fizikai Lapok*, 48:436–452, 1941.
555. William T. Tutte. A theory of 3-connected graphs. *Indagationes Mathematicae*, 23:441–455, 1961.
556. William T. Tutte. *Connectivity in graphs.* Number 15 in Mathematical Expositions. University of Toronto Press, 1966.
557. Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM Journal on Computing*, 31(2):398–427, 2001.
558. Thomas W. Valente and Robert K. Foreman. Integration and radiality: measuring the extent of an individual's connectedness and reachability in a network. *Social Networks*, 1:89–105, 1998.
559. Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.
560. Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.
561. Edwin R. van Dam and Willem H. Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 373:241–272, 2003.
562. René van den Brink and Robert P. Gilles. An axiomatic social power index for hierarchically structured populations of economic agents. In Robert P. Gilles and Picter H.M. Ruys, editors, *Imperfections and Behaviour in Economic Organizations*, pages 279–318. Kluwer Academic Publishers Group, 1994.
563. René van den Brink and Robert P. Gilles. Measuring domination in directed networks. *Social Networks*, 22(2):141–157, May 2000.

564. Stijn M. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.
565. Santosh Vempala, Ravi Kannan, and Adrian Vetta. On clusterings - good, bad and spectral. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS'00)*, pages 367–378, 2000.
566. The Stanford WebBase Project. http://www-diglib.stanford.edu/ testbed/doc2/-WebBase/.
567. Yuchung J. Wang and George Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.
568. Stephen Warshall. A theorem on boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962.
569. Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
570. Stanley Wasserman and Philippa Pattison. Logit models and logistic regressions for social networks: I. An introduction to Markov graphs and p^*. *Psychometrika*, 60:401–426, 1996.
571. David S. Watkins. QR-like algorithms for eigenvalue problems. *Journal of Computational and Applied Mathematics*, 123:67–83, 2000.
572. Alison Watts. A dynamic model of network formation. *Games and Economic Behavior*, 34:331–341, 2001.
573. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393:440–442, 1998.
574. Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.
575. Alfred Weber. *Über den Standort der Industrien*. J. C. B. Mohr, Tübingen, 1909.
576. Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, 2001.
577. Jeffery Westbrook and Robert E. Tarjan. Maintaining bridge-connected and biconnected components on-line. *Algorithmica*, 7:433–464, 1992.
578. Douglas R. White and Stephen P. Borgatti. Betweenness Centrality Measures for Directed Graphs. *Social Networks*, 16:335–346, 1994.
579. Douglas R. White and Karl P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5:193–234, 1983.
580. Scott White and Padhraic Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, 2003.
581. Hassler Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54:150–168, 1932.
582. R. W. Whitty. Vertex-disjoint paths and edge-disjoint branchings in directed graphs. *Journal of Graph Theory*, 11(3):349–358, 1987.
583. Harry Wiener. Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69:17–20, 1947.
584. Eugene P. Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *Annals of Mathematics*, 62:548–564, 1955.
585. Eugene P. Wigner. On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, 67:325–327, 1958.
586. Herbert S. Wilf. *generatingfunctionology*. pub-ap, 1994.
587. James H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, 1965.
588. Thomas Williams and Colin Kelley. *Gnuplot documentation*.
589. Gerhard Winkler. *Image Analysis, Random Fields, and Markov Chain Monte Carlo Methods*. Springer-Verlag, 2nd edition, 2003.
590. Gerhard J. Woeginger. Exact algorithms for $\mathcal{NP}$-hard problems: A survey. In *Proceedings of the 5th International Workshop on Combinatorial Optimization (Aussois'2001)*, volume 2570 of *Lecture Notes in Computer Science*, pages 185–207. Springer-Verlag, 2003.

591. Kesheng Wu and Hort Simon. Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 22(2):602–616, 2000.
592. Stefan Wuchty and Peter F. Stadler. Centers of complex networks. *Journal of Theoretical Biology*, 223:45–53, 2003.
593. Norman Zadeh. Theoretical efficiency of the Edmonds-Karp algorithm for computing maximal flows. *Journal of the ACM*, 19(1):184–192, 1972.
594. Bohdan Zelinka. Medians and peripherians tree. *Archivum Mathematicum (Brno)*, 4:87–95, 1968.
595. Uri Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication. *Electronic Colloquium on Computational Complexity (ECCC)*, 60(7), 2000.

This bibliography is available in BibTeX format from `www.inf.uni-konstanz.de/algo/gina/gina.bib`.

Index

acceptance region 281
adjacent **7**
Aitken extrapolation 75
algorithm
– ANF 299
– betweenness centrality 68
– BlockRank 78
– Burt 269
– Dijkstra's 63, 166
– dynamic PageRank approximation 80–82
– Floyd-Warshall 65, 298
– HITS 54
– Kleinberg 346
– Lanczos 387
– McKay's nauty 321, 329
– MDS
– – Bădoiu 262
– – Kruskal 259
– PageRank approximation 74
– power iteration 67
– Prim's 166
– SALSA 55
– shortcut value 70–72
– Stoer and Wagner 166
– triangle counting (AYZ) 304
all-pairs shortest paths problem *see* APSP
α-magnifier 396
APSP 10, 64
– dynamic 65
Arnoldi method 388
assortative mixing 112
attractiveness 279
authority 54, 107, 137
automorphism 13
– group 319

β-measure 99
BFS 9, 63, 68, 298
bisection width 399
block *see also* component, biconnected, 254
– cardinality 272
– density 272
– parameter 284
blockmodel 253
blockmodeling 253
– a posteriori 284
– a priori 284
– generalized 274
– stochastic 275
BlockRank 78, 91
bow tie structure 77, 429
breadth-first search *see* BFS

cactus 148, 157–158
Carrington-Heil-Berkowitz index 273
CATREGE 227
center 25
centrality
– bargaining 51
– betweenness 29–32
– – algorithm 68
– – approximation 74
– – current-flow 41
– – edge 31
– – max-flow 37, 66
– – random-walk 43
– – relative 88
– – shortest-path 29
– Botafogo et al. 57
– centroid value 23
– closeness 22
– – approximation 72
– – current-flow 42
– – random-walk 45
– cumulative number of nominations 57–59
– degree 20
– eccentricity 22
– edge centrality 34
– edge value 34

– eigenvector 49
– Hubbell 50
– Markov 45
– PageRank *see* PageRank
– radiality 23
– reach 32
– shortcut value 38
– status 47
– stress 28–29
– vertex 96, 98, 104
– vitality 36
– Wiener index 38
centralization 59
characteristic polynomial 374
chromatic number 397, 407
clique 114–115
– clan 115
– club 115
– maximal clique 114
– maximum clique 114, 339
cluster distance measure 269
clustering 179
– coefficient 344
– function 212
– fuzzy 209
– hierarchical 268
– nested 210
clustering coefficient 302, **303**, 317
– weighted 304
co-cited 111
coarsening 179
cohesiveness 418–419
colored relational structure 285
coloring 397
component
– biconnected 169–170
– connected **9**
– strongly connected **9**, 170–172
– triconnected 172–176
– weakly connected **9**
CONCOR 270
conditional independence 288
conductance 184
configuration 259, 290
– isomorphism 290
connectivity 11–12, 422
– algebraic 394
– conditional 421–422
– – edge 421
– edge 11
– mean 425–426
– vertex 11
core 129
core-periphery model 49
correlation coefficient 257
correlation matrix 257, 270
cospectral 379, 381
coverage 182
critical value 281
current 41–42
cut 10, 147
– s-t- 10
– minimum 11, 147
cut function 179
cut tree 149
cut-edge 143
cut-vertex 143
cutset *see* separator
cycle **9**

data mining 178
degree **8**
– average degree 133, 343, 393
– distribution 294
– – complementary cumulative 316
– in-degree **8**
– minimum m-degree 419–420
– out-degree **8**
degree sequence 355–364
– generating function 357–361
– graph construction 362
– realizable 361
degree vector 334
density 131
– degree of order 132
– error 273
depth-first search *see* DFS
destination
– of an edge 7
DFS 9, 123, 169, 172
diameter 296, 394, 422
– effective 297
– node 317
diameter sequence
– edge deleted 423–424
– vertex deleted 423–424
disconnection probability 434
discrete probability space 14
distance 10, 295
– average 295
– average connected distance 426
– average path distance 344
– characteristic 295
– Euclidean 257, 367
– MCES distance 338
– MCIS distance 338
– mean distance 394
– σ-distance 335
distortion 263, **301**, 317

– global 302
distribution
– degree 348
– exponential 278
– Markov field 289
– null 282
– p_1 279
– random field 289
dyad 278, 285

eccentricity 21, 295
– effective 297
edge **7**
– parallel 8
edge graph 34
edit distance 332
effective diameter 431
eigengap 76, 108
eigenvalue 14, 48, 58, 66, 348, 373
– histogram 408
– interlacing 388
eigenvector 14, 49, 58, 66, 373
equivalence 253
– automorphic 240
– bundle 251
– compositional 250
– local role 251
– multiplex regular 245
– orbit 241
– perfect 241
– regular 223, 244
– regular relative 243
– strong structural 218
– structural 220, 324
– weak role 250
– Winship-Pattison 251
event 14
expansiveness 279
expectation maximization 203

facility location 20
flow network **10**, 158–163
– type 1 160
– type 2 160
flow tree 148
fragmentation 426

Gauss-Seidel iteration 77
generating function 356
GI *see* graph isomorphism
Gibbs sampling 287
Gomory-Hu Tree *see* cut tree
goodness-of-fit 273
Google 2
graft 390
– symmetric 391
grap
– isomorphism 319–331
graph **7**
– block 78
– dependency 289
– directed **7**
– distance metric 332
– histogram 334
– incidence 35
– isomorphism **13**, 318
– loop-free 8
– matching 332
– multiplex 245
– product 392
– – modular 339
– quasi-stable 105
– random *see* random graph
– reduced 254
– regular 9, 364
– similarity 332
– simple 8
– stable 105
– sum 391
– undirected **7**
– unstable 105
– weighted **8**
– with multiple relations 244
graph model 342
– copying model 350
– Erdős-Rényi 342, 410
– initial attractiveness 350
– p^* 288–290
– p_1 276–281
– preferential attachment 348
– random 342
– small world 344
– Waxman 367
graph process 349

hierarchy 179
h-neighborhood 296
hop plot 296, 317
hub *see* authority
hypothesis
– alternative 281
– composite 281
– null 281
– simple 281

ILP **12**, 190
image matrix 253
– computing 271–272
incremental distance sequence
– edge-deleted 423–424

– vertex-deleted 423–424
independence number 398, 421
integer linear program *see* ILP
inter-cluster edges 179
interconnectedness 426
interlacing 388
Internet Movie Database 5
Internet topology 368–371
intra-cluster edges 179
inverse participation ratio 407
isomorphism 19
isoperimetric number 394

lambda set 142
λ-covering 163
laminar 157
lattice 220–221, 225
law of large numbers 286
likelihood
– equation 280
– function 280
– maximum likelihood estimation 280
line graph *see* edge graph
linear program 12
– integer *see* ILP
Linearized Chord Diagram 350
linkage process 198
local search 345
log-odds ratio 279
Luccio-Sami set 141
lumpable 79

margin 279
Markov chain **14**, 52, 288, 363
– aperiodic 15
– ergodic reversible 185
– homogeneous 14
– irreducible 15
Markov condition 14
matrix
– adjacency 13, 44, 48, 58
– correlation 257, 270
– fundamental 46
– generalized adjacency 285
– image 253
– incidence 13, 380
– Laplacian 13, 41, 379
– normalized Laplacian 13, 381
– $\mathcal{P}$-permuted adjacency 254
– stochastic 15
– transition 15, 52
max-flow problem 10, 36, 66, 135
maximum likelihood estimation 280
median 26
min-cost flow problem 11

monotonicity constraint 259
motif 306
motifs 340
multi-level approach 207
multidimensional scaling 258–268
multigraph **8**

neighborhood 274
– h- 296
network 7
network flow 10–11
network games 364
network resilience 435
network statistics 293
Neyman-Pearson lemma 282
non-complete extended p-sum 392
normal set 142
$\mathcal{NP}$-completeness 12

Oracle of Bacon 5
origin
– of an edge 7

p^*-model 288–290
p_1-model 276–281
PageRank 2, 53
– approximation 74
– dynamic approximation 80–82
– modular 91
– personalized 89
– topic sensitive 90
pairing 351
parameter estimation 202
particle hopping network 83
partition 100, 217
– equitable 239, 324
– orbit 241, 323
– vertex 323
path **9**
– shortest *see* shortest path
path algebra 297
path distance 335
performance 187
persistence 422–423
– edge 422
– vector 423
personalization vector 89
plex 126
polynomial
– characteristic 14, 374
– reliability 432–433
position 253
potential 41
power iteration 53
power law 294, 316, 348, 406

preferential attachment 348–350, 410
– continuum theory 352
– master equation 353
– rate equation 354
probabilistic resilience 435
product-moment coefficient 257
productivity *see* expansiveness

QR-decomposition 386

radius 25, 295
random graph 342, 408–415
random walk 15
rank-stable 110
ranking
– distance 110
– method 102
– problem 102
RASCAL 339
refinement 179
regular interior 227
rejection region 281
role 216
– primitive 224
role assignment 217
– exact 239
– multiplex regular 245
– problem 234
– regular 223, 244
– strong structural 218
– structural 220
role graph 218, 244
routing number 397

sample space 14
scatter diagram 260
search engine 2
semicircle law 409
semigroup 247
separator
– edge 11
– vertex 11
shifting process 201
shortest path **10**, 28–34, 297
– all-pairs 295, 298
– disjoint 300
– distinct 300
– – number of **300**, 317
– single-source 296, 298
single-source shortest paths problem *see* SSSP
spectral density 408
spectrum 14, 374–385
– adjacency 375
– Laplacian 380
splitting process 198
SSSP 10, 63
stress 260
structural index 19, 59
subgraph **9**, 336
– isomorphism 337

test statistic 281
– error types 281
– likelihood ratio 282
– log likelihood ratio 282
Theorem
– Courant-Fischer 382
– Edmonds' Branching 147
– Fiedler's 384
– Ford-Fulkerson 11, 145
– Hammersley-Clifford 289
– Kotzig's 147
– Matrix-Tree 380
– Max-Flow Min-Cut 10, 145
– Menger's 11, 145
– Turán's 114
– Whitney's 145
tightly-knit community 55
toughness 420–421
– edge- 421
trace 374
traffic-simulation 83
transition 274
transitivity **303**, 317
– index of 317
– ratio 317
transposition 274
traversal set 33
triangle 302
triple **303**, 317

vertex **7**
vertex expansion 396

walk **9**, 120, 132
– random 43, 52, 92, 142, 185, 204