How Hard is it to Determine if a Graph has a 2-Role Assignment?

Fred S. Roberts
Department of Mathematics
and
Center for Operations Research
Rutgers University
New Brunswick, NJ 08903 USA

Li Sheng[1]
Department of Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104 USA

Abstract

Role assignments, introduced by Everett and Borgatti [2], who called them role colorings, formalize the idea, arising in the theory of social networks, that individuals of the same social role will relate in the same way to the individuals playing counterpart roles. If G is a graph, a *k-role assignment* is a function r mapping the vertex set onto the set of integers $\{1, 2, \ldots, k\}$ so that if $r(x) = r(y)$, then the sets of roles assigned to the neighbors of x and y are the same. We ask how hard it is to determine if a graph has a 2-role assignment and show that recognizing if a graph has a 2-role assignment is NP-complete.

Acknowledgment

The authors gratefully acknowledge the support of the National Science Foundation under grants SES-9211492 and NSF-SBR97-09134 to Rutgers University.

Draft of August 17, 2000.

[1]Part of the work was done when the author was at the Center for Operations Research at Rutgers University.

1 Introduction

Role assignments, introduced by Everett and Borgatti [2], who called them role colorings, formalize the idea, arising in the theory of social networks, that individuals of the same social role will relate in the same way to the individuals playing counterpart roles. We define this idea precisely below. We investigate in this paper the computational complexity of the problem of determining if a graph has a role assignment with two different roles.

All graphs in the paper are simple graphs without loops or parallel edges and with more than one vertex, unless specified otherwise. Let $G = (V, E)$ be a graph with vertex set V and edge set E. An edge with end vertices u and v is denoted by uv. For $S \subseteq V$, $G(S)$ denotes the subgraph induced by S. For $v \in V$, the (*open*) *neighborhood* $N_G(v) = N(v)$ of vertex v in the graph G is the set of all vertices adjacent to v in G. For all graph-theoretical terminology that is not defined in this paper, see Golumbic [5] and Roberts [6].

Suppose $G = (V, E)$ is a graph. Let $r : V \to Z$, where Z is the set of positive integers, and call $r(x)$ the *role* assigned to x. For $S \subseteq V$, denote $r(S) = \{r(x) : x \in S\}$. We say that r defines a *role assignment* if

$$r(x) = r(y) \Rightarrow r(N(x)) = r(N(y)) \tag{1}$$

This is the definition introduced by Everett and Borgatti [2]. It was formalized earlier in the language of graph homomorphisms and vertex partitions by White and Reitz [10] and goes back to an earlier idea of Sailer [9]. The terms “regular equivalence” and “regular coloration” are also used for the same concept. For surveys of the literature of role assignments, see Everett and Borgatti [3] and Roberts and Sheng [8]. We say r is a k-*role assignment* if it is a role assignment and it uses exactly the integers $1, 2, \ldots, k$. For convenience, we identify role assignments which differ only by a permutation of roles. The graph of Figure 1 shows a role assignment. Note for example that $r(N(b)) = r(N(c)) = r(N(d)) = \{1, 2, 3\}$.

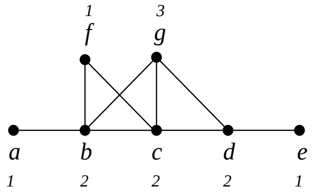


Figure 1: A role assignment

We say that G is *k-role assignable* if it has a k-role assignment. One of the issues in the study of role assignments has been to determine which graphs are k-role assignable for given k. Roberts [7] studies this problem for indifference graphs.

The assignment $r(x) \equiv 1$ is a role assignment if and only if the graph has no isolated vertices or all isolated vertices, so this describes the graphs which are 1-role assignable. The assignment where $r(x)$ is different for each x is always a role assignment, so every graph of n vertices is n-role assignable.

The 2-role assignable graphs are the first interesting class of graphs. Given a k-role assignment, we can build a corresponding *role graph* by letting the vertices be the integers 1, 2, ..., k and taking an edge between i and j if and only if some vertex of role i is adjacent to some vertex of role j. If $k = 2$, there are six possible unlabeled role graphs, R_1 through R_6 of Figure 2. In what follows, we will always assume by symmetry that in role graph R_i, the left vertex in Figure 2 is 1.

Figure 2: The possible role graphs arising from 2-role assignments

It is easy to check if a graph G (with more than one vertex) has a 2-role assignment with role graph R_i for $i = 1$, 2, 3, or 4. Let I be the set of isolated vertices in G.

(i). G has a 2-role assignment with role graph R_1 if and only if $I = V(G)$.

(ii). G has a 2-role assignment with role graph R_2 if and only if $I \neq V(G)$ and $I \neq \phi$.

(iii). G has a 2-role assignment with role graph R_3 if and only if $I = \phi$ and G is disconnected.

(iv). G has a 2-role assignment with role graph R_4 if and only if $I = \phi$ and G is bipartite.

We will show that the problems of determining if a graph G has a 2-role assignment with role graph R_5 or R_6 are NP-complete. We will also show that the problem of determining if a graph has a 2-role assignment is NP-complete, and in particular the problem remains NP-complete for connected graphs. (Note that a disconnected graph is always 2-role assignable with role graph R_1, R_2 or R_3.) The significance of these conclusions is that there probably are no good algorithms for determining whether or not a graph has a 2-role assignment, and therefore whether or not the 2-role model fits data.

2 The NP-Completeness for 2-Role Assignability

We shall use the terminology and notation of Garey and Johnson [4], and in particular we say one problem Π_1 is *polynomially transformable* or simply *reducible* to another problem Π_2, denoted $\Pi_1 \preceq \Pi_2$, if there exists a function f mapping the instances of Π_1 into the instances of Π_2 such that f is computable deterministically in polynomial time, and a solution to the instance $f(I)$ of Π_2 gives a solution to the instance I of Π_1, for all I.

We shall reduce the well-known 3-satisfiability problem (3SAT) to the problem of determining whether a graph is 2-role assignable with role graph R_5. Since 3SAT is NP-complete, we shall conclude that our problem is as well, once we show it is in the class NP.

We first recall the 3-satisfiability problem. Let $U = \{u_1, u_2, \ldots, u_m\}$ be a set of Boolean variables. *A truth assignment* for U is a function $t : U \to \{T, F\}$. If $t(u) = T$ we say u is "true" under t; if $t(u) = F$ we say that u is "false". If u is a variable in U, then u and $\overline{u}$ are *literals* over U. The literal u is true under t if and only if the variable u is true under t; the literal $\overline{u}$ is true if and only if the variable u is false. A *clause* over U is a set of literals over U, such as $\{u_1, \overline{u}_3, u_8\}$. It represents the disjunction of those literals and is *satisfied* by a truth assignment if and only if at least one of its members is true under that assignment. We denote by $|c|$ the number of literals in the clause c. A collection C of clauses over U is *satisfiable* if and only if there exists some truth assignment for U that simultaneously satisfies all the clauses in C. Such a truth assignment is called a *satisfying truth assignment* for C. The SATISFIABILITY problem is specified as follows:

SATISFIABILITY
INSTANCE: A set U of variables and a collection C of clauses over U.
QUESTION: Is there a truth assignment for U that satisfies all the clauses in C?

Cook [1] showed that SATISFIABILITY is NP-complete. We shall be interested in the 3-SATISFIABILITY problem stated as follows.

3-SATISFIABILITY (3SAT)
INSTANCE: Collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses on a finite set U of variables such that $|c_i| = 3$ for $1 \leq i \leq m$.
QUESTION: Is there a truth assignment for U that satisfies all the clauses in C?

Theorem 2.1 (Cook [1]) *3-SATISFIABILITY is NP-complete.*

We will now define the problems in which we are interested.

2-ROLE ASSIGNABILITY with role graph R_5 (2RA5)
INSTANCE: A graph $G = (V, E)$ and the role graph R_5.
QUESTION: Is G 2-role assignable with role graph R_5?

This question can be interpreted as follows: Given a graph G, is there a role assignment for G with roles 1 and 2 so that every vertex of role 1 is adjacent to a vertex of role 1 and to no vertices of role 2 and every vertex of role 2 is adjacent to vertices of both roles 1 and 2?

2-ROLE ASSIGNABILITY with role graph R_5 for a graph with at least one triangle in each of its components (2RA5-TRIANGLE).
INSTANCE: A graph $G = (V, E)$ with at least one triangle in each of its components, and the role graph R_5.
QUESTION: Is G 2-role assignable with role graph R_5?

2-ROLE ASSIGNABILITY with role graph R_6 (2RA6)
INSTANCE: A graph $G = (V, E)$ and the role graph R_6.
QUESTION: Is G 2-role assignable with role graph R_6?

2-ROLE ASSIGNABILITY (2RA)
INSTANCE: A graph $G = (V, E)$.
QUESTION: Is G 2-role assignable?

Given a graph $G = (V, E)$, let $r : V \to \{1, 2, \ldots, k\}$, let $W \subseteq V$ and recall that $G(W)$ denotes the subgraph induced by the set W. Let $S \subseteq \{1, 2, \ldots, k\}$. We sometimes refer to this set of roles S as a *role-set*. For convenience, we use $i\mathrm{F}j$ on W to mean that every vertex of W of role i is forced to be adjacent to some vertex of W of role j; and $i\mathrm{N}j$ on W to mean that a vertex of W of role i can not be adjacent to any vertex of W of role j. Furthermore, we use $i\mathrm{F}S$ and $i\mathrm{N}S$ to mean that $i\mathrm{F}j$ and $i\mathrm{N}j$ respectively, for any $j \in S$.

Lemma 2.2 *For any graph $G = (V, E)$, let W_1 and W_2 be subsets of V. Let $r : W_1 \cup W_2 \to \{1, 2\}$ be a 2-role assignment with role graph R_5 for both $G(W_1)$ and $G(W_2)$. If a vertex of W_1 of role 1 is not adjacent to any vertex of W_2 of role 1, then r is a 2-role assignment of $G(W_1 \cup W_2)$ with role graph R_5.*

Proof. Let x and y be two vertices of $W_1 \cup W_2$ with $r(x) = r(y) = 1$. If $x \in W_1$ and $y \in W_2$, $xy \notin E$ by the condition; if x and y are both in W_i for some $i = 1$ or 2, $xy \notin E$ since r is a 2-role assignment of $G(W_i)$ with role graph R_5. This ensures that 1N1 on $W_1 \cup W_2$. It is routine to check that 1F2 and 2F$\{1, 2\}$ on $W_1 \cup W_2$. It follows that r is a 2-role assignment of $G(W_1 \cup W_2)$ with role graph R_5. □

<u>**Theorem**</u> **2.3** *2RA5 is NP-complete, that is the problem of determining if G has a 2-role assignment with role graph R_5 is NP-complete.*

Proof. It is easy to see that 2RA5 $\in$ NP since a nondeterministic algorithm need only guess an assignment of 2 roles and check in polynomial time whether it is a 2-role assignment with role graph R_5. Now, 3SAT is NP-complete by Theorem 2.1, so it remains to show that 3SAT $\preceq$ 2RA5.

To transform 3SAT to 2RA5, let $U = \{u_1, u_2, \ldots, u_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ be any instance of 3SAT. We must construct a graph $G = (V, E)$ such that G is 2-role assignable with role graph R_5 if and only if C is satisfiable.

The construction will be made up of two components, truth-setting components and satisfaction testing components, augmented by some additional edges for communicating between various components. (The construction is similar to one given in Garey and Johnson [4] for reducing 3SAT problem to the VERTEX COVER problem).

For each variable $u_i \in U$, there is a truth-setting component $T_i = (V_i, E_i)$ as shown in Figure 3(a), with

$$V_i = \{u_i, \overline{u}_i, a_{i1}, a_{i2}, a_{i3}\}$$

$$E_i = \{u_i\overline{u}_i, u_i a_{i3}, \overline{u}_i a_{i3}, a_{i1}a_{i2}, a_{i2}a_{i3}\}$$

By abuse of notation, the vertices u_i and $\overline{u}_i$ will be called the literal vertices or simply literals u_i and $\overline{u}_i$.

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. There is a satisfaction testing component $S_j = (V_j', E_j')$ as shown in Figure 3(b), 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 only part of the construction that depends on which literals occur in which clauses is the collection of communication edges. For each clause $c_j \in C$, let the three literals in c_j, i.e., elements of the form u_i or $\overline{u}_i$, be denoted by x_{j1}, x_{j2} and x_{j3}. Then the communication edges emanating from S_j are given by:

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

Notice that for each c_{jk}, $k \in \{1, 2, 3\}$, there is one and only one vertex that is adjacent to c_{jk} in E_j'', and this unique vertex adjacent to c_{jk} must be the corresponding literal vertex for c_{jk} in the clause c_j.

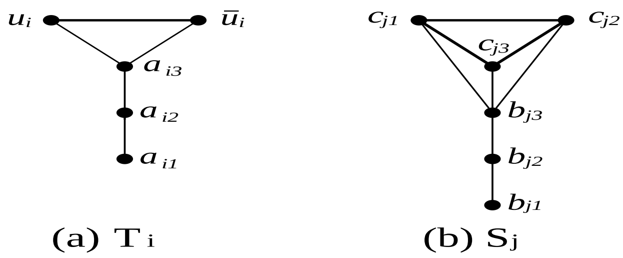


Figure 3: The truth setting component T_i for variable u_i and the satisfaction testing component S_j for clause c_j

To complete the construction of our instance of 2RA5, let $G = (V, E)$, where

$$V = (\bigcup_{i=1}^{n} V_i) \cup (\bigcup_{j=1}^{m} V_j')$$

and

$$E = (\bigcup_{i=1}^{n} E_i) \cup (\bigcup_{j=1}^{m} E_j') \cup (\bigcup_{j=1}^{m} E_j'')$$

Figure 4 shows an example of the graph obtained when $U = \{u_1, u_2, u_3, u_4\},\ C = \{\{u_1, u_2, \overline{u}_3\}, \{u_2, \overline{u}_3, u_4\}\}$.

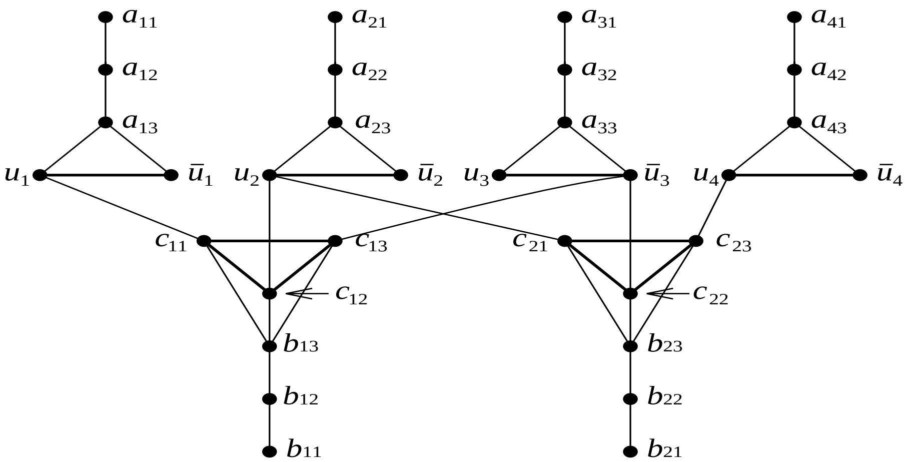


Figure 4: 2RA5 instance resulting from 3SAT instance in which $U = \{u_1, u_2, u_3, u_4\},\ C = \{\{u_1, u_2, \overline{u}_3\}, \{u_2, \overline{u}_3, u_4\}\}$.

It is easy to see how the construction can be accomplished in polynomial time. All that remains to be shown is that C is satisfiable if and only if G has a 2-role assignment with role graph R_5.

" $\Leftarrow$". First suppose that $r : V \rightarrow \{1, 2\}$ is a 2-role assignment with role graph R_5 for G. Then, for all $i \in [1, n]$ and $j \in [1, m]$,

$$r(a_{i1}) = r(b_{j1}) = 1,\ r(a_{i2}) = r(a_{i3}) = r(b_{j2}) = r(b_{j3}) = 2 \tag{2}$$

Therefore for each i, one of $\{u_i, \overline{u}_i\}$ must have role equal to 1 and the other have role equal to 2. We can use this 2-role assignment r to obtain a truth assignment $t : U \rightarrow \{T, F\}$. We merely set $t(u_i) = T$ if and only if $r(u_i) = 2$. To see that this truth assignment t satisfies each of the clauses c_j in C, consider the roles for the vertices in S_j. Since $r(b_{j2}) = r(b_{j3}) = 2$ and a vertex of role 2 must be adjacent to some vertex of role 1, we will have that $r(c_{jk}) = 1$ for some $k \in \{1, 2, 3\}$. Therefore, the literal vertex adjacent to c_{jk} in G must have role 2, that is the literal in the clause c_j corresponding to c_{jk} must be true under the truth assignment t. Hence, clause c_j is satisfied by t. Because this holds for every $c_j \in C$, it follows that t is a satisfying truth assignment for C. (To illustrate the proof with the example of Figure 4, note that a 2-role assignment is obtained by taking $r(a_{i1}) = r(b_{j1}) = 1$ for $i = 1, 2, 3, 4, j = 1, 2$, and $r(\overline{u}_1) = r(\overline{u}_2) = r(u_3) = r(\overline{u}_4) = r(c_{12}) = r(c_{22}) = 1$ and all other $r(x) = 2$. Thus, in particular, we have $t(u_2) = T$ and $t(\overline{u}_3) = T$ and this is enough to show that both clauses of C are satisfied.)

" $\Rightarrow$". Conversely, suppose C has a satisfying truth assignment $t : U \rightarrow \{T, F\}$. We can obtain an assignment $r : V \rightarrow \{1, 2\}$ from t as follows. For each T_i, if $t(u_i) = T$, let $r(u_i) = 2$, $r(\overline{u}_i) = 1$, if $t(\overline{u}_i) = T$, let $r(\overline{u}_i) = 2$, $r(u_i) = 1$, and let the $r(a_{ik})'s$ be the same as in equation (2). Now for each S_j, let c_{jk}, $k \in \{1, 2, 3\}$, be some vertex whose corresponding literal in the clause c_j is true under t. Note that such a vertex c_{jk} exists, because since t is a satisfying truth assignment for C, at least one literal in the clause c_j must be true under t. Let $r(c_{jk}) = 1, r(c_{jl}) = 2$ for $l \neq k$, and let the $r(b_{jk})'s$ be the same as in equation (2). Let $W_1 = \bigcup_{j=1}^{j=m} V'_j$ and $W_2 = \bigcup_{i=1}^{i=n} V_i$. Note that $W_1 \cup W_2 = V$, and r is a 2-role assignment with role graph R_5 for both $G(W_1)$ and $G(W_2)$. Notice also that if x is a vertex in W_1 that is adjacent to some vertex y in W_2, then $x = c_{jk}$ for some $j \in [1, m]$, $k \in \{1, 2, 3\}$. Since if $r(c_{jk}) = 1$ for some $j \in [1, m]$ and $k \in \{1, 2, 3\}$, the corresponding literal in the clause c_j must be true under t, that is the unique literal vertex adjacent to c_{jk} in G must have role 2.

This ensures that a vertex in W_1 of role 1 is not adjacent to any vertex in W_2 of role 1. By Lemma 2.2, r is a 2-role assignment of $G(W_1 \cup W_2) = G$ with role graph R_5. □

Corollary 2.4 *2RA5-TRIANGLE is NP-complete.*

Proof. The proof of Theorem 2.3 gives a graph with no isolated vertices and a triangle in each component. □

Theorem 2.5 *2RA is NP-complete, that is the problem of determining if a graph is 2-role assignable is NP-complete.*

Proof. It is easy to see that 2RA $\in$ NP since a nondeterministic algorithm need only guess an assignment of 2 roles and check in polynomial time whether it is a 2-role assignment. Now, 2RA5-TRIANGLE is NP-complete by Corollary 2.4, so it remains to show that 2RA5-TRIANGLE $\preceq$ 2RA.

To transform 2RA5-TRIANGLE to 2RA, let $G = (V, E)$ be an instance of 2RA5-TRIANGLE. We must construct a graph $G' = (V', E')$ such that G is 2-role assignable with role graph R_5 if and only if G' is is 2-role assignable.

The construction is as follows: Let $C_1, C_2, ...C_k$ be all the components of G. By abuse of notation, we use C_i to denote the vertex set of the component C_i as well as the subgraph of G induced by C_i. Let u_i be a vertex that is on a triangle of the component C_i. The construction of our instance of 2RA, $G' = (V'E')$, is completed by setting

$$V' = V \cup (\bigcup_{i=1}^{k} \{a_i, b_i, x_i, y_i\})$$

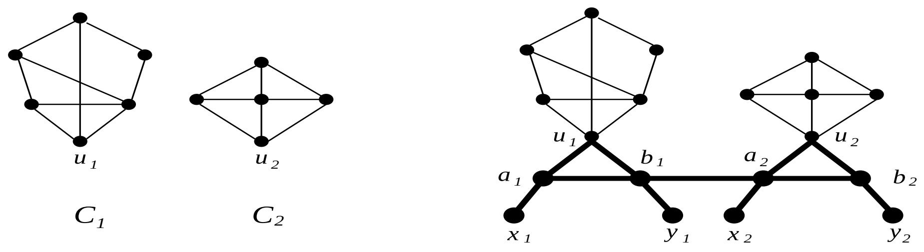
$$E' = E \cup (\bigcup_{i=1}^{k} \{u_i a_i, u_i b_i, a_i b_i, a_i x_i, b_i y_i\}) \cup E''$$

where

$$E'' = \bigcup_{i=1}^{k-1} \{b_i a_{i+1}\}$$

Notice that E'' ensures that G' is connected.

Figure 5(b) shows an example of the graph G' obtained when the 2RA5-TRIANGLE instance G is given in Figure 5(a).



(a) G (b) G′

Figure 5: 2RA instance G' resulting from 2RA5-TRIANGLE instance G with C_1 and C_2 being the two components of G, where the enlarged vertices and darkened edges are those added to G.

It is easy to see how the construction can be accomplished in polynomial time. All that remains to be shown is that G has a 2-role assignment with role graph R_5 if and only if G' has a 2-role assignment.

Suppose G has a 2-role assignment r with role graph R_5. Then for $i = 1, 2, \cdots, k$, letting $r(a_i) = r(b_i) = 2$ and $r(x_i) = r(y_i) = 1$ gives a 2-role assignment for G'. Conversely, suppose G' has a 2-role assignment r. Then the role graph is not R_1, R_2 or R_3 since G' is connected. It is not R_4 since G' has a triangle, and it is not R_6 since G' has a pendant vertex. Thus, it is R_5 and so 1N1, 1F2 and 2F$\{1, 2\}$ on V'. It follows that for $i = 1, 2, \cdots, k$, $r(x_i) = r(y_i) = 1, r(a_i) = r(b_i) = 2$. We show that r restricted to G gives a 2-role assignment with role graph R_5. We just need to show that 1N1, 1F2 and 2F$\{1, 2\}$ on V. The only complication is to investigate $r(N_G(u_i)), i = 1, 2, \cdots, k$. If $r(u_i) = 1$, then there is a neighbor of u_i in G since u_i is on a triangle of G, and any neighbor of u_i in G has role 2 since 1N1 on V' and $V \subseteq V'$. If $r(u_i) = 2$, since $r(a_i) = r(b_i) = 2$, u_i must be adjacent to some vertex of role 1 in G since 2F1 on V' and $V \subseteq V'$. However, u_i is on a triangle of G, and the other two vertices of that triangle cannot both have role 1 since 1N1 on V', so one of them has role 2. $\square$

Corollary 2.6 *The problem of determining if a connected graph has a 2-role assignment is NP-complete.*

Proof. The proof of Theorem 2.5 gives an instance G' of 2RA that is connected. □

__Theorem__ **2.7** *2RA6 is NP-complete, that is the problem of determining if G has a 2-role assignment with role graph R_6 is NP-complete.*

Proof. It is easy to see that 2RA6 $\in$ NP since a nondeterministic algorithm need only guess an assignment of 2 roles and check in polynomial time whether it is a 2-role assignment with role graph R_6. Now, 3SAT is NP-complete by Theorem 2.1, so it remains to show that 3SAT $\preceq$ 2RA6.

To transform 3SAT to 2RA6, let $U = \{u_1, u_2, \ldots, u_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ be any instance of 3SAT. We must construct a graph $G = (V, E)$ such that G is 2-role assignable with role graph R_6 if and only if C is satisfiable.

The construction is as follows:

For each variable $u_i \in U$, there is a truth-setting component $T_i = (V_i, E_i)$ as shown in Figure 6(a), with

$$V_i = \{u_i, \overline{u}_i, a_{i1}, a_{i2}\}$$

$$E_i = \{u_i a_{i1}, a_{i1}\overline{u}_i, u_i a_{i2}, a_{i2}\overline{u}_i\}$$

By abuse of notation, the vertices u_i and $\overline{u}_i$ will be called the literal vertices or simply literals u_i and $\overline{u}_i$.

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. There is a satisfaction testing component $S_j = (V_j, E_j)$ as shown in Figure 6(b), with

$$V_j' = \{c_{j1}, c_{j2}, c_{j3}, b_j, d_j\}$$

$$E_j' = \{c_{j1}b_j, c_{j2}b_j, c_{j3}b_j, b_j d_j\}$$

(Here, the b_j and d_j play the role of b_{j1}, b_{j2}, b_{j3} of Figure 3(b).)

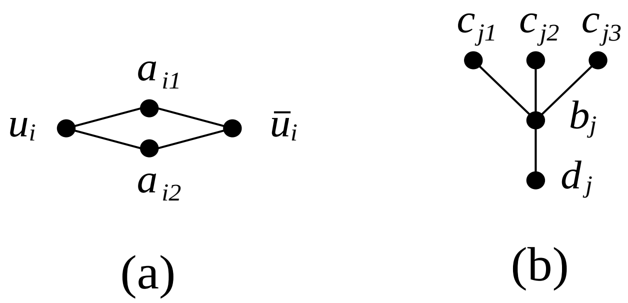


Figure 6: The substructures corresponding to variable u_i and clause c_j

Now, connect $d_1, d_2, \cdots, d_m$ by a big cycle using the vertices and edges shown in Figure 7 (a). That is, for any two clauses c_j, c_{j+1} (the addition is under $mod\ m$), let

$$V_j'' = \{f_j, e_{j1}, e_{j2}\}$$

$$E_j'' = \{d_j e_{j1}, e_{j1} f_j, f_j e_{j2}, e_{j2} d_{j+1}\}$$

The only part of the construction that depends on which literals occur in which clauses is a collection of communication edges between components. If literal x_i ($x_i = u_i$ or $\overline{u}_i$) is in position c_{jk} ($k = 1, 2, 3$) (we shall call vertex x_i the literal vertex corresponding to c_{jk} in G), then connect x_i and c_{jk} by the structure shown in Figure 7 (b), with

$$V_{jk}''' = \{g_{jk1}, g_{jk2}\}$$

$$E_{jk}''' = \{x_i g_{jk1}, g_{jk1} c_{jk}, x_i g_{jk2}, g_{jk2} c_{jk}\}$$

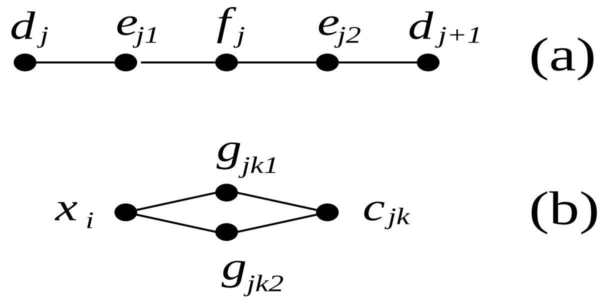


Figure 7: The substructure that connects clauses c_j and c_{j+1} and the substructure that connects c_{jk} with its corresponding literal vertex

To complete the construction of our instance of 2RA6, let $G = (V, E)$, where

$$V = (\bigcup_{i=1}^{n} V_i) \cup [\bigcup_{j=1}^{m} (V'_j \cup V''_j)] \cup (\bigcup_{j=1}^{m} \bigcup_{k=1}^{3} V'''_{jk})$$

and

$$E = (\bigcup_{i=1}^{n} E_i) \cup [\bigcup_{j=1}^{m} (E'_j \cup E''_j)] \cup (\bigcup_{j=1}^{m} \bigcup_{k=1}^{3} E'''_{jk})$$

Figure 8 shows an example of the graph obtained when $U = \{u_1, u_2, u_3, u_4\},\ C = \{\{u_1, \overline{u}_1, u_2\}, \{u_2, \overline{u}_2, u_3\}, \{\overline{u}_3, \overline{u}_3, \overline{u}_4\}\}$.

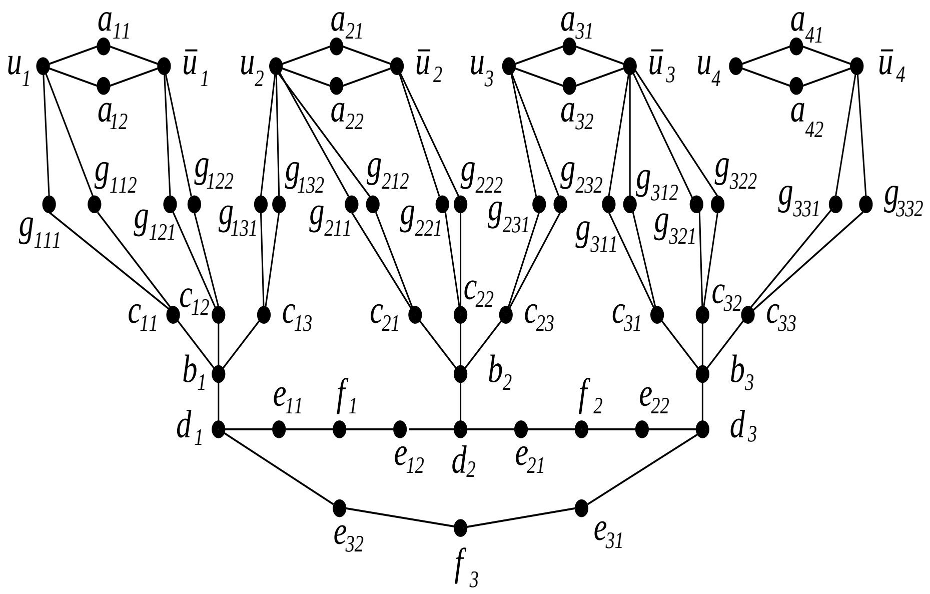


Figure 8: 2RA6 instance resulting from 3SAT instance in which $U = \{u_1, u_2, u_3, u_4\},\ C = \{\{u_1, \overline{u}_1, u_2\}, \{u_2, \overline{u}_2, u_3\}, \{\overline{u}_3, \overline{u}_3, \overline{u}_4\}\}$.

It is easy to see how the construction can be accomplished in polynomial time. All that remains to be shown is that C is satisfiable if and only if G has a 2-role assignment with role graph R_6.

"$\Leftarrow$". Suppose that $r : V \rightarrow \{1, 2\}$ is a 2-role assignment with role graph R_6 for G. We use the observation (which we shall call *observation O*) that the two neighbors of a degree two vertex must have different roles. By (repeated application of) observation O, all the d vertices have the same role, all the f vertices have the same role, and the role for d vertices is different from that for f vertices. So, we can assume WLOG that all d vertices have role 2 and all f vertices have role 1. By observation O, for each i, one of $u_i, \overline{u}_i$ must have role equal to 1 and the other have role equal to 2. A truth assignment $t : U \rightarrow \{T, F\}$ can be obtained by letting $t(u_i) = T$ if $r(u_i) = 2$, and letting $t(u_i) = F$ if $r(u_i) = 1$. To see that this truth assignment t satisfies each of the clauses c_j in C, consider the roles for the vertices corresponding to the clause c_j. It is easy to see that $r(c_{jk}) = 1$ for some $k \in \{1, 2, 3\}$, for otherwise all the neighbors of vertex b_j will have role 2. Then by observation O, the literal vertex corresponding to c_{jk} in G must have role 2, that is the literal in the clause c_j at the position c_{jk} is true under the truth assignment t. Hence, clause c_j is satisfied by t. Because this holds for every $c_j \in C$, it follows that t is a satisfying truth assignment for C.

" $\Rightarrow$". Conversely, suppose C has a satisfying truth assignment $t : U \rightarrow \{T, F\}$. We can obtain an assignment $r : V \rightarrow \{1, 2\}$ from t as follows. For each i, if $t(u_i) = T$, let $r(u_i) = 2$, $r(\overline{u}_i) = 1$; if $t(\overline{u}_i) = T$, let $r(\overline{u}_i) = 2$, $r(u_i) = 1$. If the literal at c_{jk} is true, let $r(c_{jk}) = 1$; otherwise, let $r(c_{jk}) = 2$. Notice that since t is a satisfying truth assignment for C, at least one literal in the clause c_j is true, which implies that at least one vertex in c_{j1}, c_{j2}, c_{j3} has role 1. Let all d vertices have role 2 and all f vertices have role 1. Let $r(a_{i1}) = r(e_{j1}) = r(g_{jk1}) = r(b_j) = 1$, $r(a_{i2}) = r(e_{j2}) = r(g_{jk2}) = 2$ for $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$, $k = 1, 2, 3$. It is routine to check that r is a 2-role assignment of G with role graph R_6. □

References

[1] S. A. Cook, The complexity of theorem-proving procedures, *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, Association for Computing Machinery, New York, 1971, 151–158.

[2] M. G. Everett and S. Borgatti, Role coloring a graph, *Math. Soc. Sci.* **21** (1991), 183–188.

[3] M. G. Everett and S. Borgatti, Regular equivalence: General theory, *J. of Math. Soc.*, **19** (1994), 29–52.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, 1979.

[5] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[6] F. S. Roberts, *Applied Combinatorics*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[7] F. S. Roberts, Role assignments and indifference graphs, *Recent Progress in Mathematical Psychology*, C.E. Dowling, F.S. Roberts, & P. Theuns (eds.), Lawrence Erlbaum Associates, Mahwah, NJ, 1998, 33-46.

[8] F. S. Roberts and L. Sheng, Role assignments, *Combinatorics, Graph Theory, and Algorithms*, Vol. II, Y. Alavi, D. Lick, & A. Schwenk (eds.), New Issues Press, Kalamazoo, MI, 1999, 729-745.

[9] L. D. Sailer, Structural equivalence: meaning and definition, computation and application, *Social Networks*, **1** (1978), 73–90.

[10] D. R. White, & K. P. Reitz,, Graph and semigroup homomorphisms on networks of relations, *Social Networks*, **5** (1983), 193-234.