Theoretical Computer Science 635 (2016) 74–84

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Computing role assignments of split graphs

Mitre Costa Dourado [1]

Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

ARTICLE INFO

Article history:
Received 18 October 2014
Received in revised form 2 May 2016
Accepted 4 May 2016
Available online 9 May 2016
Communicated by M. Latapy

Keywords:
Algorithms
NP-complete problems
Role assignments
Split graphs
Social networks

ABSTRACT

A k-role assignment of a simple graph G is a surjective function $r : V(G) \to \{1, \dots, k\}$ where $\{r(u') : u' \in N(u)\} = \{r(v') : v' \in N(v)\}$ for every pair $u, v \in V(G)$ with $r(u) = r(v)$. This concept appears in the context of social networks. It is known that the problem of finding a 2-role assignment of chordal graphs can be solved in linear time and that deciding whether a graph of this class admits a k-role assignment, for any fixed $k \geq 3$, is NP-complete. In this work, we investigate this problem on split graphs, a subclass of chordal graphs. We characterize the split graphs admitting a 3-role assignment. Such result leads to a linear time algorithm for this case. Furthermore, we prove that deciding whether a split graph has a k-role assignment is NP-complete for any fixed $k \geq 4$.

1. Introduction

We consider undirected finite graphs having no multiple edges. Given a graph G, we denote its vertex set by $V(G)$ and its edge set by $E(G)$. The cardinality of $V(G)$ is the *order of* G. For sets[2] $S, T \subseteq V(G)$, we denote by $G[S]$ the subgraph of G induced by S; and the *neighborhood of* S *in* G is $N_G(S) = \{u : u \in V(G)$ and there is $v \in S$ such that $uv \in E(G)\}$. We also denote $N_T(S) = T \cap N_G(S)$. If S is a singleton, for instance $S = \{v\}$, we can write $N_T(v)$ instead of $N_T(S)$. If every two vertices of $G[S]$ are adjacent (respectively, not adjacent), S is called a *clique of* G (respectively, an *independent set of* G).

Given two graphs G and R, the former having no loops and the latter possibly having loops, a *homomorphism* from G to R is a function $r : V(G) \to V(R)$ such that $r(u)r(v) \in E(R)$ whenever $uv \in E(G)$. We call G the *guest graph* and R the *host graph*. It is interesting to note that graph homomorphisms [18] generalize graph colorings, one of the most important research topics in graph theory [2]. If a homomorphism r is surjective and the restriction of r to the neighborhood of each $u \in V$, i.e., the function $r_u : N_G(u) \to N_R(r(u))$, is surjective, then r is said to be a *locally surjective homomorphism from* G *to* R or an R*-role assignment of* G. For $k = |V(R)|$, we say that r is a k*-role assignment of* G.

Role assignments were introduced in [10] under the name of *role colorings* and are also known as *color dominations* [19]. Applications to this problem can be found in social networks [10,23,24] and distributed computing [4,5]. The problem of deciding whether G admits an R-role assignment is called R-Role Assignment.[3] The complexity of R-Role Assignment was totally classified for general graphs in [11] where it was proved that this problem is NP-complete even for bipartite graphs. For trees, R-Role Assignment is solvable in polynomial time for any fixed R [12]. A polynomial time solution for Role

E-mail address: mitre@dcc.ufrj.br.

[1] Partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ).

[2] When we mean that S is a proper subset of T we write $S \subset T$.

[3] It is clear that a positive answer for this problem can exist only if the order of G is at least the order of R. Therefore, this requirement will be omitted in the sequel.

http://dx.doi.org/10.1016/j.tcs.2016.05.011
0304-3975/

Assignment, a stronger version of R-Role Assignment which receives as input both graphs G and R, was presented in [17] for unit interval graphs. Recently, it was shown [6] that this problem is NP-complete if G has pathwidth at most 4 while it can be solved in polynomial time if we bound the treewidth of G and at the same time bound the maximum degree of G or R.

The problem of deciding whether G admits a k-role assignment is known as k-Role Assignment. For general graphs, k-Role Assignment was shown to be NP-complete in [24] for $k = 2$, and in [11] for any fixed $k \geq 3$. For chordal graphs, such problem was shown to be solvable in linear time if $k = 2$ and to be NP-complete for every fixed $k \geq 3$ in [20]. A characterization of indifference graphs, a subclass of chordal graphs, admitting a 2-role assignment was given in [25]. In this work, we investigate the k-Role Assignment problem on split graphs, another subclass of chordal graphs.

Contribution and paper organization. A graph G is *split* if $V(G)$ can be partitioned into one clique C and one independent set I. A *split partition* (C, I) can be obtained in linear time [16]. This class also admits a characterization by forbidden subgraphs, namely, a graph is split if and only if it does not have an induced subgraph isomorphic to a cycle with 4 vertices, a cycle with 5 vertices, or a pair of disjoint edges [13]. Despite its simple definition, some important problems are NP-complete on split graphs, like Dominating set [3,8], Hamiltonian cycle [22], and Maximum Cut [1]. An entire chapter is dedicated to this class in the book [15]. It is interesting to observe that split graphs form a subclass of chordal graphs, a class for which k-Role Assignment is totally solved [20].

In this work, we show that finding a k-role assignment for a split graph can be done in linear time if and only if k is at most 3, unless P = NP. We begin showing that a 2-role assignment is always possible for a split graph. Let G be a split graph with split partition (C, I). If G is disconnected, consider a split partition in which C is maximum and assign role 1 to the isolated vertices and role 2 to the remaining ones; otherwise, consider a split partition in which I is maximum and assign role 1 to all vertices of C and role 2 to all vertices of I. Note that, for these 2-role assignments, it holds $E(R) = \varnothing$ if $E(G) = \varnothing$; $E(R) = \{22\}$ is disconnected and $E(G) \neq \varnothing$; $E(R) = \{12\}$ if G is connected and $|C| = 1$; and $E(R) = \{11, 12\}$ otherwise.

In Section 2, we study the behavior of some role assignments when restricted to subgraphs of the graph. In Section 3, we present a characterization of split graphs admitting a 3-role assignment. Such characterization leads to a linear time algorithm for finding a 3-role assignment of a split graph if one exists. In Section 4, we prove that k-Role Assignment is NP-complete for any fixed $k \geq 4$.

2. Some properties of role assignments on general graphs

In this section, we see some properties of role assignments that will be useful in the sequel. These properties hold for graphs in general and not only for split graphs. We begin recalling a basic property of homomorphisms.

Lemma 1. *[18] If f is a homomorphism from G to H, then the distance from $f(u)$ to $f(v)$ is at most the distance from u to v for any vertices $u, v \in V(G)$.*

The following result is direct from the previous one.

Corollary 1. *If G is a connected graph, then the host graph of every role assignment of G is also connected.*

The next result shows that, for every k-role assignment of a connected graph G and $v \in V(G)$, there is a connected subgraph of G of order k containing v such that all roles all represented.

Lemma 2. *If G is a connected graph and r a k-role assignment of G, then, for every vertex $v \in V(G)$, there is a set $S \subseteq V(G)$ such that $|S| = k$, $v \in S$, $r(u) \neq r(w)$ for any $u, w \in V(S)$, and $G[S]$ is connected.*

Proof. For any $v \in V(G)$, we show how to find a set $S \subseteq V(G)$ of size k containing v such that $r(u) \neq r(u')$ for any $u, u' \in S$ and $G[S]$ is connected. Initially, set $S = \{v\}$. Next, while there exist vertices not marked in S, choose and mark one of them, say u, and then, for each $y \in N_R(r(u))$ such that no vertex with role y was added to S yet, add to S one vertex of $N_G(u)$ with role y.

We show that S is the wanted set. Graph $G[S]$ is connected because a vertex is added to S only if it is adjacent to one previously added to S. Furthermore, since at most one vertex of every role was added to S, it holds $|S| \leq k$.

Now, suppose for contradiction that $|S| < k$. By Lemma 1, R is a connected graph. This means that there is $xx' \in E(R)$ such that there is $w \in S$ with $r(w) = x$ and none $w' \in S$ with $r(w') = x'$. Now, consider the iteration of the algorithm used to construct S in which w was chosen as a vertex of S not marked yet. Since $x' \in N_R(x) = N_R(r(w))$ and S has no vertices with role x', some vertex with role x' belonging to $N_G(u)$ must be added to S. Since w' is a vertex satisfying these properties, we have a contradiction. □

Now, we see that if r is an R-role assignment of G, $S \subseteq V(R)$, and V' is the set of vertices of G having role assignment in S, then the restriction of r to the vertices of V' is an $R[S]$-role assignment of $G[V']$.