Theoretical Computer Science 535 (2014) 46–53

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The algorithmic complexity of bondage and reinforcement problems in bipartite graphs

Fu-Tao Hu [a], Moo Young Sohn [b,*]

[a] School of Mathematical Sciences, Anhui University, Hefei, 230601, PR China
[b] Mathematics, Changwon National University, Changwon, 641-773, Republic of Korea

ARTICLE INFO

Article history:
Received 8 August 2013
Received in revised form 24 March 2014
Accepted 2 April 2014
Communicated by V.Th. Paschos

Keywords:
Complexity
NP-completeness
NP-hardness
Domination
Bondage
Total bondage
Reinforcement
Total reinforcement

ABSTRACT

Let $G=(V,E)$ be a graph. A subset $D \subseteq V$ is a dominating set if every vertex not in D is adjacent to a vertex in D. The domination number of G, denoted by $\gamma(G)$, is the smallest cardinality of a dominating set of G. The bondage number of a nonempty graph G is the smallest number of edges whose removal from G results in a graph with domination number larger than $\gamma(G)$. The reinforcement number of G is the smallest number of edges whose addition to G results in a graph with smaller domination number than $\gamma(G)$. In 2012, Hu and Xu proved that the decision problems for the bondage, the total bondage, the reinforcement and the total reinforcement numbers are all NP-hard in general graphs. In this paper, we improve these results to bipartite graphs.

1. Introduction

For terminology and notation on graph theory not given here, the reader is referred to Xu [19]. Let $G=(V,E)$ be a finite, undirected and simple graph, where $V=V(G)$ is the vertex set and $E=E(G)$ is the edge set of G. For a vertex $x \in V(G)$, let $N_G(x)=\{y : xy \in E(G)\}$ be the *open set of neighbors* of x and $N_G[x]=N_G(x) \cup \{x\}$ be the *closed set of neighbors* of x.

A subset $D \subseteq V$ is a *dominating set* of G if every vertex in $V-D$ has at least one neighbor in D. The *domination number* of G, denoted by $\gamma(G)$, is the minimum cardinality among all dominating sets of G. A dominating set D is called a *γ-set* of G if $|D|=\gamma(G)$. The domination is an important and classic notion that has become one of the most widely researched topics in graph theory and also is used to study property of networks frequently. A thorough study of domination appears in the books [7,8] by Haynes, Hedetniemi, and Slater. Among various problems related to the domination number, some focus on graph alterations and their effects on the domination number. Here, we are concerned with two particular graph modifications, the removal and addition of edges from a graph. The *bondage number* of G, denoted by $b(G)$, is the minimum number of edges whose removal from G results in a graph with a domination number larger than the one of G. The *reinforcement number* of G, denoted by $r(G)$, is the smallest number of edges whose addition to G results in a graph with a domination number smaller than the one of G. The bondage number and the reinforcement number were introduced by Fink et al. [3] and Kok, Mynhardt [13], respectively, in 1990. The reinforcement number for digraphs has been studied by Huang,

* Corresponding author.
E-mail addresses: hufu@mail.ustc.edu.cn (F.-T. Hu), mysohn@changwon.ac.kr (M.Y. Sohn).

http://dx.doi.org/10.1016/j.tcs.2014.04.005

Wang and Xu [12]. The bondage number and the reinforcement number are two important parameters for measuring the vulnerability and stability of the network domination under link failure and link addition. Recently, Xu [20] gave a review article on bondage numbers in 2013.

A dominating set D of a graph G without isolated vertices is called a *total dominating set* if every vertex in D is also adjacent to another vertex in D. The *total domination number* of G, denoted by $\gamma_t(G)$, is the minimum cardinality among all total dominating sets of G. In this paper, we use the symbol D_t to denote a total dominating set. A total dominating set D_t is called a γ_t*-set* of G if $|D_t| = \gamma_t(G)$. The total domination was introduced by Cockayne et al. [2]. Total domination in graphs has been extensively studied in the literature. In 2009, Henning [6] surveyed the recent results on total domination in graphs. The *total bondage number* of G without isolated vertices, denoted by $b_t(G)$, is the minimum number of edges whose removal from G results in a graph with a total domination number larger than the one of G. The *total reinforcement number* of G without isolated vertices, denoted by $r_t(G)$, is the smallest number of edges whose addition from G results in a graph with a total domination number smaller than the one of G. The total bondage number of a graph was first studied by Kulli and Patwari [14] and further studied by Sridharan, Elias, Subramanian [17], Huang and Xu [11]. The total reinforcement number of a graph was first studied by Sridharan, Elias, Subramanian [18] and further studied by Henning, Rad and Raczek [9].

For a graph parameter, knowing whether or not there exists a polynomial-time algorithm to compute its exact value is the essential problem. If the decision problem corresponding to the computation of this parameter is NP-hard or NP-complete, then polynomial-time algorithms for this parameter do not exist unless $\mathrm{NP} = P$. The problem of determining the domination number has been proved NP-complete for chordal bipartite graphs [15]. For the total domination number, the problem has been proved NP-complete for bipartite graphs [16]. There are many other complexity results for variations of domination, these results can be found in the two books [1,8] and the survey [6].

As regards the bondage problem, Hattingh et al. [5] showed that the restrained bondage problem is NP-complete even for bipartite graphs. Hu and Xu [10] have showed that the bondage, the total bondage, the reinforcement and the total reinforcement numbers are all NP-hard for general graphs. We know that even if a problem is known to be NP-hard or NP-complete, it may be possible to find a polynomial-time algorithm for a restricted set of instances from a particular application. The bondage number and reinforcement number in graphs are very interesting research problems in graph theory. There are many results about the bondage number and reinforcement number in bipartite graphs. Many famous networks are bipartite graphs, such as hypercube graphs, partial cube, grid graphs, median graphs and so on. If we proved these decision problems for the bondage and the reinforcement are all NP-hard, then the studies on the bondage number and reinforcement number in bipartite graphs are more meaningful and we can directly deduce the decision problems for the bondage and the reinforcement are both NP-hard in general graphs. So we should be concerned about the algorithmic complexity of the bondage and reinforcement problems in bipartite graphs.

In this paper, we will show that the decision problems for the bondage, the total bondage, the reinforcement and the total reinforcement numbers are all NP-hard even for bipartite graphs. In other words, there are not polynomial-time algorithms to compute these parameters unless $P = \mathrm{NP}$. The proofs are in Section 3, Section 4 and Section 5, respectively.

One can ask whether these four problems belong to NP or not. Since the problem of determining the domination number is NP-complete, and it is not clear that there is a polynomial algorithm to verify $\gamma(G - B) > \gamma(G)$ (or $\gamma(G + R) < \gamma(G)$) for any subset $B \subset E(G)$ (or $R \subset E(\bar{G})$), these four problems are not obviously seen to be in NP. We conjecture that they are not in NP. But we can not prove that determining the bondage and the reinforcement are not NP-problems. This will be our work to study further. In this paper, we only present the results that these four problems are all NP-hard in bipartite graphs.

2. 3-satisfiability problem

In *Computers and Intractability: A Guide to the Theory of NP-Completeness* [4], Garey and Johnson outline three steps to prove a decision problem to be NP-hard. We follow the three steps for proving our four decision problems to be NP-hard. We prove our results by describing a polynomial transformation from the known NP-complete problem: 3-satisfiability problem. To state the 3-satisfiability problem, in this section, we recall some terms.

Let U be a set of Boolean variables. A *truth assignment* for U is a mapping $t : U \to \{T, F\}$. If $t(u) = T$, then u is said to be "true" under t; if $t(u) = F$, then u is said to be "false" under t. If u is a variable in U, then u and $\bar{u}$ are *literals* over U. The literal u is true under t if and only if the variable u is true under t; the literal $\bar{u}$ is true if and only if the variable u is false.

A *clause* over U is a set of literals over U. It represents the disjunction of these literals and is *satisfied* by a truth assignment if and only if at least one of its members is true under that assignment. A collection $\mathscr{C}$ of clauses over U is *satisfiable* if and only if there exists some truth assignment for U that simultaneously satisfies all the clauses in $\mathscr{C}$. Such a truth assignment is called a *satisfying truth assignment* for $\mathscr{C}$. The 3-satisfiability problem is specified as follows.

3-**satisfiability problem (3SAT)**:

Instance: *A collection* $\mathscr{C} = \{C_1, C_2, \ldots, C_m\}$ *of clauses over a finite set* U *of variables such that* $|C_j| = 3$ *for* $j = 1, 2, \ldots, m$.

Question: *Is there a truth assignment for* U *that satisfies all the clauses in* $\mathscr{C}$*?*