Journal of Discrete Algorithms 9 (2011) 252–262

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

An exact algorithm for connected red–blue dominating set ☆

Faisal N. Abu-Khzam [a,*], Amer E. Mouawad [a], Mathieu Liedloff [b]

[a] *Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon*
[b] *Laboratoire d'Informatique Fondamentale d'Orléans, Université d'Orléans, 45067 Orléans Cedex 2, France*

ARTICLE INFO

Article history:
Available online 23 March 2011

Keywords:
Exact algorithms
Dominating set
Weighted Steiner tree

ABSTRACT

In the Connected Red–Blue Dominating Set problem we are given a graph G whose vertex set is partitioned into two parts R and B (red and blue vertices), and we are asked to find a connected subgraph induced by a subset S of B such that each red vertex of G is adjacent to some vertex in S. The problem can be solved in $\mathcal{O}^*(2^{n-|B|})$ time by reduction to the Weighted Steiner Tree problem. Combining exhaustive enumeration when $|B|$ is small with the Weighted Steiner Tree approach when $|B|$ is large, solves the problem in $\mathcal{O}^*(1.4143^n)$. In this paper we present a first non-trivial exact algorithm whose running time is in $\mathcal{O}^*(1.3645^n)$. We use our algorithm to solve the Connected Dominating Set problem in $\mathcal{O}^*(1.8619^n)$. This improves the current best known algorithm, which used sophisticated run-time analysis via the measure and conquer technique to solve the problem in $\mathcal{O}^*(1.8966^n)$.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Given a graph $G = (V, E)$, the Dominating Set problem asks for a minimum subset of V whose neighborhood is its complement in V. Several variations of this problem have recently become of significant theoretical interest because of their numerous applications. Problems like directed domination, multiple domination, distance domination, independent domination and connected domination are now common in the literature, and illustrate the growing interest and the increasing work on different variants of the domination problem.

The Connected Dominating Set problem (cds) requires that the dominating set induces a connected subgraph. cds received considerable attention lately, due to its applications in wireless ad hoc networks, in which a connected dominating set is used as an underlying structure for performing various functions. Examples include protocols for location-based routing, multicast/broadcast and energy conservation (see [3,11] for surveys on some of the algorithms and techniques for computing connected dominating sets in wireless ad hoc networks).

cds is $\mathcal{NP}$-complete [10], remains so even when restricted to planar graphs [4], and is not fixed parameter tractable in arbitrary graphs [5]. Most techniques discussed in [3] use approximation algorithms, which is not surprising when the best known exact algorithm has running time in $\mathcal{O}^*(1.8966^n)$[1] [7] on graphs of order n. In fact, the algorithm proposed in [7] solves the Maximum-Leaf Spanning Tree problem, which is equivalent to cds. In [9], Fomin et al. presented an $\mathcal{O}^*(1.94^n)$ algorithm for a direct solution to the problem, being the first exact algorithm breaking the 2^n barrier.

☆ This research has been supported in part by the research council of the Lebanese American University.

* Corresponding author.
E-mail addresses: faisal.abukhzam@lau.edu.lb (F.N. Abu-Khzam), amer.mouawad@lau.edu.lb (A.E. Mouawad), mathieu.liedloff@univ-orleans.fr (M. Liedloff).
URLs: http://www.csm.lau.edu.lb/fabukhzam (F.N. Abu-Khzam), http://www.univ-orleans.fr/lifo/Members/Mathieu.Liedlof (M. Liedloff).

[1] Throughout this paper we use the modified big-Oh notation that suppresses all polynomially bounded factors. For functions f and g we say $f(n) \in \mathcal{O}^*(g(n))$ if $f(n) \in \mathcal{O}(g(n)poly(n))$, where $poly(n)$ is a polynomial.

1570-8667/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.jda.2011.03.006

In this paper we consider the Connected Dominating Set problem applied to red–blue graphs. A graph $G = (V, E)$ is a red–blue graph whenever the vertex set V is partitioned into two sets R and B. The elements of R and B are called red and blue vertices respectively. A connected red–blue dominating set of G is a subset S of the blue vertices such that S dominates all red vertices of G and the subgraph of G induced by S is connected. The corresponding problem, denoted by CRBDS, asks to find a connected red–blue dominating set of smallest possible cardinality.

CRBDS can effectively replace the more general CDS in a large number of network protocols. Naturally, "powerful" computers in a network (i.e. servers) could be represented by blue nodes, while destination hosts (i.e. laptops) could be represented by red nodes thus making dominating set based routing more efficient. In power management, CRBDS would be more appropriate than CDS since nodes in sleep mode can be mapped to red nodes while blue nodes would preserve the ability of the network to forward messages. Despite the many natural applications of CRBDS, the problem seems to have been neglected prior to this work. The connectivity property required in both CDS and CRBDS makes them part of the family of non-local problems which are usually harder to solve exactly. Nevertheless, we shall first present an exact algorithm that solves the CRBDS problem in $\mathcal{O}^*(1.3645^n)$ time and polynomial space and then prove that CDS is solvable in $\mathcal{O}^*(1.8619^n)$ by reduction to CRBDS.

2. Preliminaries

Throughout this paper we denote by $G = (R \cup B, E)$ a red–blue graph whose vertex set $V = R \cup B$. For a vertex $v \in B$, we denote by $N_R(v)$ the set of red neighbors of v. The set $N_B(v)$ is defined analogously when $v \in R$. The Red–Blue Dominating Set problem, henceforth RBDS, is defined formally as follows:

Given: a red–blue graph $G = (R \cup B, E)$ and a positive integer k.
Question: does B have a subset S such that $|S| \leqslant k$ and $N_R(S) = R$?

RBDS is $\mathcal{NP}$-complete [6], even if the input is restricted to planar graphs [1]. We note here that the technique described in [13] for Dominating Set can also be adopted for RBDS and yields a worst-case running time in $\mathcal{O}^*(1.2302^n)$. In fact, the mentioned algorithm reduces the Dominating Set problem to Minimum Set Cover and has to keep track of an instance of size $2n$ that contains both the vertices and their neighborhoods. In the case of RBDS, the run-time is in $\mathcal{O}^*(1.2302^n)$ because the number of vertices and neighborhoods is bounded by n only. A slightly better asymptotic bound was obtained recently by van Rooij et al. in [14].

In this paper, we consider a variant of both CDS and RBDS, namely CRBDS, in which the required dominating set induces a connected subgraph. CRBDS is $\mathcal{NP}$-complete since it is equivalent to RBDS when B, the set of blue vertices, forms a fully connected subset. We shall see that CRBDS can be reduced to the Weighted Steiner Tree problem WST, which can be solved in $\mathcal{O}^*(2^k)$ and polynomial space [12] where k is the number of terminal nodes (the red nodes in a CRBDS instances). Our proposed algorithm uses the following "either-or" approach.

- While favorable branching rules exist, the algorithm proceeds by applying the branch and reduce paradigm. Favorable branching rules imply that the $\mathcal{O}^*(1.3645^n)$ upper bound will never be exceeded.
- When the algorithm reaches a search state where favorable branching rules are no longer possible, an instance of the Weighted Steiner Tree problem is generated to solve the remaining part of the problem.

This either-or approach can be applied, potentially, to other problems whenever the absence of favorable branching conditions leads to a reduction of the input instance to some manageable size. To simplify our algorithm and its analysis, in the sequel, we assume that:

- R is an independent set.
- B is a dominating set of G (otherwise, we have a no instance).
- B induces a connected component of G.

If G contains distinct blue connected components, the algorithm is applied to each connected component that dominates all red vertices.

The following lemma is needed in the analysis of our algorithm. It does provide us with a breakpoint, as to when to stop branching and apply the WST algorithm.

Lemma 1. *Let $G = (R \cup B, E)$ be a red–blue bipartite graph such that*:

1. *Every blue vertex has degree four or less and every red vertex has degree two or more.*
2. *For $d \in \{2, 3, 4\}$, every red vertex of degree d has at most one blue neighbor of degree d while every other blue neighbor has a smaller degree.*
3. *No two degree-two vertices in R can have a common neighbor in B.*

Download English Version:

https://daneshyari.com/en/article/431040

Download Persian Version:

https://daneshyari.com/article/431040

Daneshyari.com