Information Processing Letters 115 (2015) 119–124

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On the maximum acyclic subgraph problem under disjunctive constraints

Sílvia Maria Santana Mapa [a,b,*,1], Sebastián Urrutia [a,2]

[a] Computer Science Department, Federal University of Minas Gerais, Belo Horizonte, Brazil
[b] Production Engineering Department, Federal Institute of Minas Gerais, Congonhas, Brazil

ARTICLE INFO

Article history:
Received 12 December 2013
Received in revised form 30 May 2014
Accepted 25 July 2014
Available online 1 August 2014
Communicated by Ł. Kowalik

Keywords:
Maximum acyclic subgraph problem
Disjunctive constraints
Approximation algorithms
Computational complexity

ABSTRACT

Disjunctively constrained versions of classic problems in graph theory such as shortest paths, minimum spanning trees and maximum matchings were recently studied. In this article we introduce disjunctive constrained versions of the Maximum Acyclic Subgraph problem. Negative disjunctive constraints state that a certain pair of edges cannot be contained simultaneously in a feasible solution. Positive disjunctive constraints enforces that at least one arc for the underlying pair is in a feasible solution. It is convenient to represent these disjunctive constraints in terms of an undirected graph, called constraint graph, whose vertices correspond to the arcs of the original graph, and whose edges encode the disjunctive constraints. For the Maximum Acyclic Subgraph problem under Negative Disjunctive Constraints we develop 1/2-approximative algorithms that are polynomial for certain classes of constraint graphs. We also show that determining if a feasible solution exists for an instance of the Maximum Acyclic Subgraph problem under Positive Disjunctive Constraints is an NP-Complete problem.

1. Introduction

A directed graph may contain directed cycles which may be undesirable or not allowed at all in some practical applications. Therefore, in order to withdraw as few arcs as possible, a maximum directed acyclic graph should be found. The problem of finding the Maximum Acyclic Subgraph (MAS) of a given directed graph $G=(V,A)$ consists in determining a maximum subset $A' \subseteq A$ for which the subgraph $G'=(V,A')$ is cycle free. This problem is known to be NP-hard [5]. Simple approximation algorithms for this problem produce solutions with at least half of the number of arcs of an optimal solution [4].

Negative disjunctive constraints forbid a certain pair of entities of the problem to be simultaneously part of any feasible solution. On the other hand, positive disjunctive constraints force at least one of the entities of a given pair to be part of any feasible solution.

Disjunctively constrained versions of classic problems in graph theory such as shortest paths, minimum spanning trees and maximum matchings were recently studied [2]. All these problems, which are polynomially solvable in their basic form, turn NP-Hard in the presence of such disjunctive constraints. The maximum flow problem under positive and negative disjunctive constraints [8] and the classical 0–1 knapsack problem under negative disjunctive constraints [7], where also recently tackled in the literature.

In this paper we consider the Maximum Acyclic Subgraph problem under Negative Disjunctive Constraints (MASNDC) and the Maximum Acyclic Subgraph problem under Positive Disjunctive Constraints (MASPDC).

* Corresponding author. Tel.: +55 0313137318100.
E-mail addresses: silvia.mapa@ifmg.edu.br (S.M.S. Mapa), surrutia@dcc.ufmg.br (S. Urrutia).

1 The author thanks CAPES and IFMG for financial support.
2 The author was partially supported by CNPq Grant 303442/2010-7.

http://dx.doi.org/10.1016/j.ipl.2014.07.013
0020-0190/

To formally define the problems, let $G = (V, A)$ be a directed graph and let $\bar{G} = (A, E)$ be an undirected graph, called constraint graph.

The Maximum Acyclic Subgraph problem under Negative Disjunctive Constraints consists in determining a subset $A' \subseteq A$ of maximum cardinality for which the subgraph $G' = (V, A')$ is cycle free and A' is an independent set in $\bar{G}$. On the other hand, the Maximum Acyclic Subgraph problem under Positive Disjunctive Constraints consists in determining a subset $A' \subseteq A$ of maximum cardinality for which the subgraph $G' = (V, A')$ is cycle free and A' is a vertex cover in $\bar{G}$. Since MASNDC and MASPDC have MAS as a special case (when $E = \emptyset$), both are NP-Hard.

In this paper we propose simple modifications of existing approximation algorithms for MAS that produce solutions to MASNDC maintaining the 1/2-approximation ratio of the original algorithms. Moreover, we show that determining the existence of a feasible solution for a given MASPDC instance is NP-Hard.

The rest of the paper is organized as follows. In the next section we present three known algorithms for approximating MAS. In Section 3 we propose six new approximation algorithms for MASNDC adapting those from the MAS literature and following two distinct approaches. In Section 4 we prove that the feasibility of MASPDC is an NP-Complete problem. Finally, in Section 5 we give some concluding remarks.

2. Approximation algorithms for maximum acyclic subgraph

Concerning MAS, we present in this section three algorithms that produce solutions with at least $|A|/2$ arcs. Since the cardinality of A is an upper bound to the optimal solution value, the algorithms are 1/2-approximative. No better constant approximation polynomial algorithm exists for the Maximum Acyclic Subgraph problem and none is expected to exist assuming the Unique Games Conjecture and $P \neq$ NP [1,6].

Algorithm Sort(G)

This straightforward algorithm first sorts the vertices of the graph arbitrarily assigning a distinct label π_i to every vertex $i \in V$. Then, it divides the arcs in A into two subsets: in A_f it stores all arcs $a = (i, j) \in A$ such that $\pi_i < \pi_j$ and in A_b all arcs $a = (i, j) \in A$ such that $\pi_i > \pi_j$. The algorithm just picks the subset (A_f or A_b) with maximum cardinality and returns the subgraph with the selected set of arcs as the solution.

Theorem 1. *The solution of Sort*(G) *is an acyclic graph with at least* $|A|/2$ *arcs.*

Proof. Observe that both subgraphs $G_f = (V, A_f)$ and $G_b = (V, A_b)$ are cycle free. A cycle $v_1, v_2, \ldots, v_1$ would imply in the presence of an arc (i, j) with $\pi_i < \pi_j$ and of an arc (k, l) with $\pi_k > \pi_l$. Also, as each arc in A is either in A_f or A_b, one of those subsets must have at least $|A|/2$ arcs.

Algorithm Greedy(G)

This algorithm uses a greedy strategy while considering arcs for inclusion in an acyclic subgraph. The algorithm maintains two sets S and T. It greedily scans all arcs in A, adding an arc to S if with its addition the induced subgraph $G[S]$ remains acyclic. Otherwise it adds the arc to T. When all arcs have been processed, the subset of largest cardinality, S or T, is selected to form the acyclic subgraph.

Theorem 2. *The solution of Greedy*(G) *is an acyclic graph with at least* $|A|/2$ *arcs.*

Proof. $G[S]$ is acyclic by construction. To show that $G[T]$ is also acyclic consider the vertices in V sorted respecting a topological sort of $G[S]$. Each arc in T forms a cycle with a subset of arcs in S and in consequence it is a backward edge considering the topological sort of $G[S]$. Since all arcs in T are backward considering that order of the vertices, the argument for G_b to be acyclic in the previous algorithm also holds for $G[T]$. As in the previous proof, each arc in A is either in S or T and one of them must have at least $|A|/2$ arcs.

Algorithm Degree(G)

The third algorithm works as follows. It processes all the vertices of G, in any order, analyzing the incoming and outgoing arcs. If a vertex has more incoming arcs than outgoing ones, the incoming arcs are removed from the graph and added to a set A' and the outgoing arcs are also removed but discarded. If a vertex has the same number of incoming and outgoing arcs, the outgoing arcs are added to the set A' and the incoming arcs are discarded. When all vertices are processed, the algorithm returns set A' to form the acyclic subgraph.

Theorem 3. *The solution of Degree*(G) *is an acyclic graph with at least* $|A|/2$ *arcs.*

Proof. Consider a cycle $x_1, \ldots, x_k, x_1$ in the subgraph $G[A']$ and, without loss of generality, let x_1 be the vertex in this cycle that was processed by the algorithm before every other vertex in the cycle. Then, either arc (x_k, x_1) or arc (x_1, x_2) would have been discarded and therefore no cycle is possible. Also note that $|A'| \geq |A|/2$, since each arc in A is examined once and, while processing a certain vertex, the algorithm always adds to $|A'|$ at least as many arcs as it discards.

For several other approximative algorithms for MAS the reader is referred to [4].

3. Approximation algorithms for maximum acyclic subgraph under negative disjunctive constraints

The direct application of the algorithms presented above to the MASNDC does not guarantee a solution with at least half of the number of arcs in an optimal solution. In fact, the presence of disjunctive constraints may