Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Spanning cactus: Complexity and extensions

Alak Kumar Datta *, Chinmay Debnath

Department of Computer & System Sciences, Visva-Bharati University, Santiniketan - 731235, India

ARTICLE INFO

Article history:
Received 15 September 2016
Accepted 23 August 2017
Available online xxxx

Keywords:
Approximation algorithms
Minimum spanning cactus
NP-Completeness
Minimum spanning tree

ABSTRACT

Minimum spanning cactus and minimum spanning cactus extension problems are studied. Both problems are NP-Complete. We present an approximation algorithm for the minimum spanning cactus extension of a forest, a hardness of approximation result of the general minimum spanning cactus problem. For the minimum spanning cactus extension problem, Kabadi and Punnen presented polynomial time algorithms for extending quasi-stars, spanning trees (Kabadi and Punnen, 2013). We present improved analysis of their algorithms in both cases. We further show that their algorithm for the extension of spanning trees can be generalized to extend any connected spanning partial cactus. As a requirement of improved implementation, we have presented a new $O(n^3)$ algorithm to compute all minimum cost monotone paths with respect to a spanning tree.

1. Introduction

An undirected graph is called a *partial cactus* if every edge of the graph is contained in at most one cycle. A *cactus* is a connected bridgeless partial cactus [10]. This means that every edge is contained in exactly one cycle in a cactus. Consider a graph, $G = (V, E)$. A subgraph $H = (V, E_H)$ of G is a *spanning cactus* of G if H is a cactus. Let c_e be the cost of an edge $e \in E$. For a spanning cactus $H = (V, E_H)$ of G, the cost of the spanning cactus H, denoted by c_H, is defined by $\sum_{e \in E_H} c_e$. *Minimum Spanning Cactus Problem (MSCP)* is to compute a spanning cactus of minimum cost.

If $H = (V, E_H)$ is any spanning partial cactus of $G = (V, E)$, one may try to add edges in H from $E \setminus E_H$ to get a spanning cactus $\bar{H} = (V, E_{\bar{H}})$ of G. If such a spanning cactus exists then we say H is cactus extendable in G and $\bar{H}$ is called a *cactus extension* of H. It is easy to observe that a cactus extension may not exist for some spanning partial cacti. The optimization version of the problem finds $\bar{H}$ so as to minimize $c_{\bar{H}}$. This is known as the *Minimum Cactus Extension Problem (MCEP).*

Minimum Spanning Tree (MST) of a graph is a minimum connected substructure of a graph. In network applications, it is often important to maintain such substructures. However, an MST substructure has an edge connectivity one; hence not reliable. On the other hand, a cactus substructure has edge connectivity two; therefore more reliable. For this reason, *MSCP, including MCEP, find applications in the design of reliable communication and transportation networks* [10,17]. *This motivates the design of more efficient algorithms for these problems.* The problems also find applications in genome comparison [6], representation of cuts in a graph [7–9], traffic estimation [13,20]. The reader may refer to [3–5,12,19] for some basic graph problems on a cactus graph.

The directed version of MSCP is studied in [15]. It is shown that the directed version is NP-Complete. Further, if the edge costs satisfy the triangle inequality, the problem becomes equivalent to the asymmetric traveling salesman problem.

The undirected version of the problem, defined above, is studied in [10]. It is proved that the problem is NP-Complete. Further, it is shown that if the edge costs satisfy the triangle inequality then the problem becomes equivalent to the traveling salesman problem, and both of them have the same approximation hardness.

* Corresponding author.
E-mail addresses: alak.datta@visva-bharati.ac.in (A.K. Datta), chinmaydebnath.rs@visva-bharati.ac.in (C. Debnath).

http://dx.doi.org/10.1016/j.dam.2017.08.020
0166-218X/

Please cite this article in press as: A.K. Datta, C. Debnath, Spanning cactus: Complexity and extensions, Discrete Applied Mathematics (2017), http://dx.doi.org/10.1016/j.dam.2017.08.020.

MCEP becomes equivalent to MSCP when the given spanning subgraph $H = (V, E_H)$ contains no edge. This implies MCEP is NP-Complete [10]. Kabadi and Punnen present a series of results mentioned below [10]. They provide a polynomial time algorithm to test if H is cactus extendable in a complete graph. Further, they prove that if H is a spanning tree then it is cactus extendable if and only if it has at least one vertex of even degree. Polynomial time algorithms are also presented for MCEP when H are paths, quasi-stars or cactus extendable spanning trees. For paths and quasi-stars the algorithms take $O(n^2)$ and $O(n^4)$ time, respectively. For spanning trees, it is argued that the algorithm works in polynomial time; but no detailed time complexity analysis is provided.

In this paper, we present a new analysis of the kabadi–punnen MCEP algorithm on a quasi-star. *We show that the algorithm works in $O(n^3)$ time, reducing the time complexity from $O(n^4)$* as claimed in [10]. An analysis of the kabadi–punnen MCEP algorithm for cactus extension of a spanning tree has been considered. We show that the analysis technique used by Kabadi et al. actually results in $O(n^5)$ time complexity. Further, *we present a way to implement the algorithm in $O(n^3)$ time.* For the new implementation, we need all minimum cost monotone paths between any two vertices with respect to a spanning tree. *We propose a new efficient algorithm to compute all these paths in $O(n^3)$ time. Next we show that the kabadi–punnen algorithm can be generalized to obtain a cactus extension of any arbitrary connected spanning cactus in $O(n^3)$ time.*

Although MCEP on a tree can be solved in polynomial time, it seems that MCEP for a forest cannot be solved in polynomial time. *We present an approximation algorithm for the problem with a ratio error two if the edge costs satisfy the triangle inequality.* It has been shown in [10] that for a graph satisfying the triangle inequality, MSCP has the same approximation hardness as the traveling salesman problem satisfying the triangle inequality. However, no result is known for the approximation of the general MSCP. In this paper, *we show that if $P \neq NP$, there exists no approximation algorithm for the general MSCP with ratio error bound $r(\geq 1)$.*

In Section 2, we present briefly kabadi–punnen MCEP algorithms for paths, quasi-stars, and spanning trees. Section 3 includes the improved time complexity analysis of the kabadi–punnen MCEP algorithm for a quasi-star. In Section 4, we present a simple $O(n^3)$ time algorithm to compute all minimum cost monotone paths with respect to a spanning tree. Section 5 presents the implementation of the kabadi–punnen algorithm for an extension of a spanning tree in $O(n^3)$ time. In Section 6, we present a way to extend this algorithm to solve MCEP for an extension of a connected partial spanning cactus. Section 7 presents the approximation algorithm for the extension of a forest. The hardness of approximation of MSCP has been studied in Section 8. Finally, Section 9 presents the conclusion and scopes for future studies.

2. kabadi–punnen algorithms

In this section we present the kabadi–punnen MCEP algorithms for extensions of Hamiltonian paths, quasi-stars, and cactus extendable spanning trees. We use some definitions from [10] for description of the algorithm. A vertex h with degree three or more in a tree T is called a *head* of a tree T if $T \setminus \{h\}$ has at most one component, called the body of T at h, that is neither a path nor an isolated vertex. Let T_h^0 denote the body of T at h. $T \setminus T_h^0$ defines a subtree of T, called the *tentacle system*, TS_h, at h. A path from h to a pendant vertex in TS_h is called a *tentacle* at h. This tentacle system is a tree in which there is exactly one head vertex. Such a tree, called *quasi-star*, is homeomorphic to a star. Fig. 1 shows a quasi-star with head vertex h.

Without any loss of generality we assume that the given graph $G = (V, E)$ is a complete graph K_n of n vertices. First we assume that H is a path of all vertices in K_n. The algorithm for an extension of the path H uses the notion of monotone path introduced in [1]. Let $P_{a,b} = P_{b,a} = a - u_1 - u_2 \ldots. u_{n-2} - b$ be a Hamiltonian path in K_n. A *monotone path* with respect to $P_{a,b}$ is a path $\Pi_{a,b} = a - u_{\Pi_1} - u_{\Pi_2} \ldots.. u_{\Pi_k} - b$, where $2 < \Pi_i + 1 < \Pi_{i+1} < n - 2$, for $i = 1, 2, \ldots.., k - 1$. The cost of the monotone path $\Pi_{a,b}$ is the sum of the costs of the edges on the path. The *minimum cost monotone path* with respect to the path $P_{a,b}$, denoted by $\beta_{P_{a,b}}(a, b)$, can be computed in $O(n^2)$ time [1]. It is straightforward to observe that $P_{a,b} \cup \beta_{P_{a,b}}(a, b)$ is the required minimum spanning cactus extension [10]. So the MCEP on a Hamiltonian path can be solved in $O(n^2)$ time.

2.1. Extension of quasi-star

The idea has been extended to solve MCEP on a quasi-star [10]. Consider a quasi-star T of n vertices as in Fig. 1. Let S be the set of all odd degree vertices in T. Note that S contains the pendant vertices $p_1, p_2, \ldots. p_k$. $h \in S$ if its degree is odd. The algorithm for MCEP on the quasi-star T in K_n, given by Kabadi et al., constructs a complete graph G' of the vertices in S. Let $\beta_T(i, j)$ be the minimum cost monotone path with respect to the unique path between i and j in T. The cost of the edge (i, j) in G' is the cost of $\beta_T(i, j)$, the minimum cost monotone path from i to j. Note that the cost of $\beta_T(h, j) = \infty$, if $(h, j) \in T$. Let M^* be the minimum cost perfect matching of the vertices in G'. They proved that $\bar{H} = T \cup \{\beta_T(u, v) | (u, v) \in M^*\}$ is the minimum cost cactus extension of T in K_n [10].

They also proved that their algorithm takes $O(n^4)$ time. The computation of cost of an edge (i, j) in G' needs computation of minimum cost monotone path $\beta_T(i, j)$; therefore can be performed in $O(n^2)$ time [1]. There can be $O(n^2)$ edges in G' in the worst case. So, the construction of G' takes $O(n^4)$ time. Further, the computation of the minimum cost perfect matching in G' takes $O(n^3)$ time [14]. Total time required, as claimed by Kabadi et al., is $O(n^4)$.

Please cite this article in press as: A.K. Datta, C. Debnath, Spanning cactus: Complexity and extensions, Discrete Applied Mathematics (2017), http://dx.doi.org/10.1016/j.dam.2017.08.020.