



Note

# The approximability of the weighted Hamiltonian path completion problem on a tree

Quincy Wu<sup>a</sup>, Chin Lung Lu<sup>b,\*</sup>, Richard Chia-Tung Lee<sup>c</sup>

<sup>a</sup>*Graduate Institute of Communication Engineering, National Chi-Nan University, Puli, Nantou Hsien 545, Taiwan, ROC*

<sup>b</sup>*Department of Biological Science and Technology, National Chiao Tung University, Hsinchu 300, Taiwan, ROC*

<sup>c</sup>*Department of Computer Science and Information Engineering, National Chi-Nan University, Puli, Nantou Hsien 545, Taiwan, ROC*

Received 7 February 2004; received in revised form 21 February 2005; accepted 2 March 2005

Communicated by D.-Z. Du

---

## Abstract

Given a graph, the Hamiltonian path completion problem is to find an augmenting edge set such that the augmented graph has a Hamiltonian path. In this paper, we show that the Hamiltonian path completion problem will unlikely have any constant ratio approximation algorithm unless  $NP = P$ . This problem remains hard to approximate even when the given subgraph is a tree. Moreover, if the edge weights are restricted to be either 1 or 2, the Hamiltonian path completion problem on a tree is still NP-hard. Then it is observed that this problem is strongly NP-hard, so it does not have any fully polynomial-time approximation scheme (FPTAS) unless  $NP = P$ . When the given tree is a  $k$ -tree, we give an approximation algorithm with performance ratio 1.5.

© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Hamiltonian path completion problem; Trees; Strongly NP-hard; Approximation algorithm; Fully polynomial-time approximation scheme

---

---

\* Corresponding author. Tel.: +886 3 5712121x56949; fax: +886 3 5729288.

*E-mail addresses:* [solomon@ipv6.club.tw](mailto:solomon@ipv6.club.tw) (Q. Wu), [cllu@mail.nctu.edu.tw](mailto:cllu@mail.nctu.edu.tw) (C.L. Lu), [rctlee@ncnu.edu.tw](mailto:rctlee@ncnu.edu.tw) (R.C.-T Lee).

## 1. Introduction

Given a graph  $G$ , a Hamiltonian path is a simple path on  $G$  which traverses each vertex exactly once. Finding a Hamiltonian path is often required in problems involving routing and the periodic updating of data structures. In the past, the Hamiltonian path completion problem was defined on unweighted graphs. For a given unweighted graph  $G = (V, E_0)$ , the Hamiltonian path completion problem is to find an augmenting edge set  $E_2$  such that  $G' = (V, E_0 \cup E_2)$  has a Hamiltonian path. Such an edge set  $E_2$  is called an *augment*. It was shown that to find an edge set  $E_2$  with minimum cardinality is an NP-hard problem [9, p. 198]. If the given graph  $G$  is a tree [6,10,11,13,16], a forest [20], an interval graph [1], a circular-arc graph [7], a bipartite permutation graph [21], a block graph [21,22,24], or a cograph [15], it was shown that there exist polynomial-time algorithms for this problem by computing the *path covering number*, which is the minimum number of vertex-disjoint paths covering all vertices. In general, for a complete bipartite graph  $K_{1,n}$ , the path covering number is  $n - 1$ , and at least  $n - 2$  edges must be added to make it have a Hamiltonian path. In other words, for  $K_{1,n}$  to have a Hamiltonian path, any optimal augment must have exactly  $n - 2$  edges [6]. For example, in Fig. 1, the graph can be covered by three paths  $\{(4, 1, 5), (2), (3)\}$ , so two edges, say  $(2,3)$  and  $(3,4)$ , may be added to make the original graph have a Hamiltonian path.

In this paper, we shall discuss the weighted Hamiltonian path completion problem, whose formal definition is given below.

**Weighted Hamiltonian Path Completion Problem (WHPCP).** *Given a complete graph  $G = (V, E)$  with edge weights  $w : E \rightarrow R^+$ , and an edge subset  $E_0 \subseteq E$ , find an augment  $E_2 \subseteq E$  such that  $G' = (V, E_0 \cup E_2)$  has a Hamiltonian path and  $\sum_{e \in E_2} w(e)$  is minimized.*

For example, in Fig. 2, given  $E_0 = \{(1, 2), (3, 4), (5, 6)\}$  and the weight of each edge assigned as in the table aside, the optimal augment is  $E_2 = \{(1, 7), (2, 5), (3, 6)\}$  with weight 10. However, this problem is hard to solve if  $E_0$  is arbitrary. Throughout this paper, we shall assume that  $E_0$  constitutes a tree. The formal definition of the problem is given below.

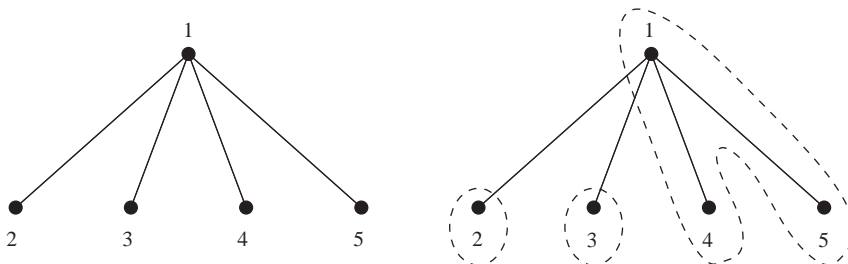


Fig. 1. A bipartite graph  $K_{1,4}$  which can be covered by 3 paths.

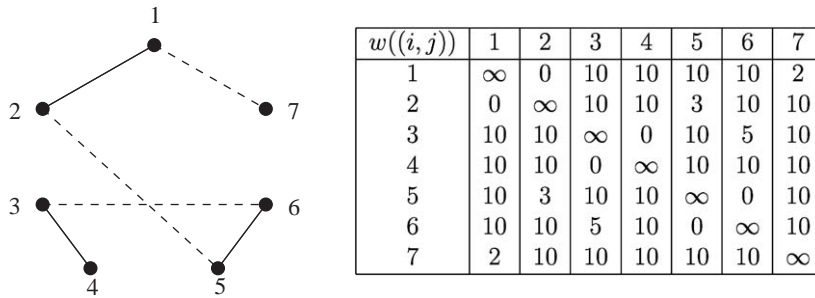


Fig. 2. An example for the weighted Hamiltonian path completion problem, where the given edge set  $E_0$  is represented by solid lines and augment  $E_2$  by dashed lines, and the weights of all the edges are shown in the table aside.

**Weighted Hamiltonian Path Completion Problem on a Tree (WHPCT).** *Given a complete graph  $G = (V, E)$  with edge weights  $w : E \rightarrow R^+$ , and an edge subset  $E_0$  constituting a spanning tree on  $G$ , find an augment  $E_2 \subseteq E$  such that  $G' = (V, E_0 \cup E_2)$  has a Hamiltonian path and the weight  $\sum_{e \in E_2} w(e)$  is minimized.*

In this paper, we shall first show that the weighted Hamiltonian path completion problem on a tree (WHPCT) will unlikely have any constant ratio approximation algorithm and it is still NP-hard even when the edge weights are restricted to be either 1 or 2. We then observe that this problem is strongly NP-hard, so it has no fully polynomial-time approximation scheme (FPTAS) unless  $NP = P$ . Furthermore, when the given tree has only one internal node, we give an approximation algorithm with performance ratio 1.5 and then extend this algorithm to trees with  $k$  internal nodes.

## 2. Non-approximability of WHPCT

To prove that WHPCT is hard to approximate, we adopt a technique which is similar to the one applied to prove that the traveling salesperson problem will unlikely have any  $\alpha$ -approximation algorithm as in [19]. In our proof, we shall reduce the Hamiltonian path problem, which is a well-known NP-complete problem [9, pp. 199–200], to the WHPCT problem. Let us define the problem first.

**Hamiltonian Path Problem (HPP).** *Given  $G = (V, E)$ , where  $|V| = n$ , determine whether  $G$  has a path of length  $n - 1$ .*

An approximation algorithm  $\mathcal{A}$  is said to be  $\alpha$ -approximation if for any problem instance, the weight of the approximate solution obtained by  $\mathcal{A}$  is bounded by  $\alpha$  times the weight of the optimal solution. We then have the following theorem:

**Theorem 2.1.** *For any  $\alpha > 1$ , if there exists a polynomial-time  $\alpha$ -approximation algorithm for WHPCT, then  $NP = P$ .*

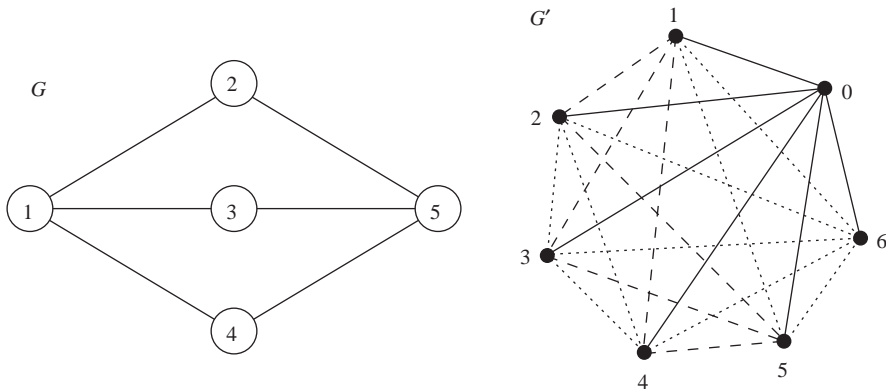


Fig. 3. Reduction from  $G$  in HPP to  $G'$  in WHPCT. In  $G'$ , the dashed edges have weight 1 and other edges have weight  $24\alpha$ .

**Proof.** Suppose there exists an approximation algorithm  $\mathcal{A}$  of WHPCT that will find an approximation solution of ratio  $\alpha$  in polynomial time. We then reduce HPP to WHPCT as follows: Given an instance of HPP, say  $G = (V, E)$  with  $V = \{v_1, v_2, \dots, v_n\}$ , we construct an instance of WHPCT, say  $G' = (V', E')$  and  $E_0$ , as follows.

- $G'$  is the complete graph with vertex set  $V' = V \cup \{v_0, v_{n+1}\}$ .
- For each  $e \in E'$ ,  $w(e) = \begin{cases} 1 & \text{if } e \in E, \\ \alpha|E|(n-1) & \text{otherwise.} \end{cases}$
- $E_0 = \{(v_0, v_i) \mid 1 \leq i \leq n+1\}$ .

Let us see Fig. 3 for an example of the reduction.

First, we claim that  $G$  has a Hamiltonian path if and only if  $G'$  has an optimal augment with weight  $n-1$ . It is easy to see that if  $G$  has a Hamiltonian path, then by the construction of  $G'$ ,  $G'$  has an augment with  $n-1$  edges, and each of these edges has weight 1. Conversely, suppose  $E_0$  has an optimal augment  $E_2$  of weight  $n-1$ . According to the discussion in the previous section, since the given  $E_0$  constitutes a tree  $K_{1,n+1}$ ,  $E_2$  must have exactly  $n-1$  edges. Since  $E_2$  is of weight  $n-1$ , each of the  $n-1$  edges in  $E_2$  must have weight exactly equal to 1 and hence  $E_2 \subseteq E$ . Let  $P \subseteq E_0 \cup E_2$  be a Hamiltonian path in  $G'' = (V', E_0 \cup E_2)$ . Since  $V'$  has  $n+2$  vertices,  $|P| = n+1$ . Because the path is Hamiltonian, the vertex  $v_{n+1}$  must be covered by the path. Since vertex  $v_{n+1}$  is of degree 1 and adjacent to  $v_0$  in  $G''$ ,  $P$  must pass through  $v_0$  to visit  $v_{n+1}$  and then stop. In other words, if we delete  $v_0$  and  $v_{n+1}$  from  $P$ , we can obtain a path  $P' \subseteq E_2$  which visits all vertices in  $\{v_1, v_2, \dots, v_n\}$  exactly once. Since  $E_2 \subseteq E$ ,  $P'$  is a Hamiltonian path for  $G$ . Therefore,  $G$  has a Hamiltonian path if and only if  $G'$  has an optimal augment with weight  $n-1$ .

Second, we claim that there is an optimal augment for  $G'$  with weight  $n-1$  if and only if  $\mathcal{A}$  cannot generate a solution containing any edge with weight  $\alpha|E|(n-1)$ . If the approximate solution contains any such edge, the weight of the augment would be at least  $\alpha|E|(n-1)$ . Since  $(\alpha|E|(n-1))/(n-1) = \alpha|E| > \alpha$ , this violates the assumption that  $\mathcal{A}$  is an  $\alpha$ -approximation algorithm. Conversely, if  $\mathcal{A}$  generates an augment with weight less than or equal to  $|E|$ , then we know this augment does not contain any edge in  $G'$  with

weight  $\alpha|E|(n-1)$ . In other words, all the edges in this augment are contained in  $E$  of the original graph  $G$ . Since such a subset in  $E$  constitutes a Hamiltonian path in  $G$ , it leads to the conclusion that the graph  $G$  has a Hamiltonian path.

By the above discussion,  $G$  has a Hamiltonian path if and only if  $\mathcal{A}$  generates an augment with weight less than or equal to  $|E|$  for  $G'$ . Hence we can use  $\mathcal{A}$  to solve the HPP in polynomial time, by examining the weight of the solution returned by  $\mathcal{A}$ . Thus, if WHPCT has a polynomial-time  $\alpha$ -approximation algorithm, then  $\text{NP} = \text{P}$ .  $\square$

### 3. Hardness results of (1,2)-HPCT

In this section, we shall discuss the (1,2)-HPCT problem, which is the WHPCT problem whose edge weights are restricted to be either 1 or 2.

**(1,2)-Hamiltonian Path Completion Problem on a Tree ((1,2)-HPCT).** *Given a complete graph  $G = (V, E)$  with edge weights  $w : E \rightarrow \{1, 2\}$ , and an edge subset  $E_0$  constituting a spanning tree on  $G$ , find an augment  $E_2 \subseteq E$  such that  $G' = (V, E_0 \cup E_2)$  has a Hamiltonian path and the weight  $\sum_{e \in E_2} w(e)$  is minimized.*

In [3], an NP-hard problem is defined to be *strongly NP-hard* if it remains to be NP-hard even when the value of the maximum number occurring in the input is bounded by some polynomial in the length of the input. That is, for any input  $x$ ,  $\max(x) \leq p(|x|)$ . Since the edge weights in any instance  $x$  of (1,2)-HPCT are either 1 or 2,  $\max(x) = 2$ , which is a constant and certainly bounded by any polynomial. It can also be shown that (1,2)-HPCT is NP-hard with a similar technique applied in the first part of Theorem 2.1. The only difference is that in the construction of  $G'$ , the weight of those edges not in  $E$  is set to 2, instead of  $\alpha|E|(n-1)$ . Therefore, we have the following observation immediately.

**Observation 3.1.** *(1,2)-HPCT is strongly NP-hard.*

A *polynomial-time approximation scheme* (PTAS) is a family of algorithms such that for any rational value  $\varepsilon > 0$ , there is a corresponding approximation algorithm whose solution is within ratio  $1 + \varepsilon$ , and the time complexity of this approximation algorithm is polynomial in the size of its input. Furthermore, when the running time of a PTAS is polynomial both in the size of the input and in  $1/\varepsilon$ , the scheme is called a *fully polynomial-time approximation scheme* (FPTAS) [3]. Some problems, like the maximum independent set problem on planar graphs and the Euclidean traveling salesperson problem are found to have PTASs [2,4], while the 0–1 knapsack problem admits an FPTAS [12]. On the contrary, some problems, such as the maximum 3-satisfiability problem, the maximum leaves spanning tree problem, the superstring problem, and the traveling salesperson problem with distances one and two, are proven by a reduction from the MAX SNP-complete class that they do not have any PTAS unless  $\text{NP} = \text{P}$  [5,8,17,18].

The following lemma states the relationship between strongly NP-hard problems and FPTAS solutions.

**Lemma 3.1** ([3], Corollary 3.19). *Let  $\mathcal{P}$  be a strongly NP-hard problem that admits a polynomial  $p$  such that  $m^*(x) \leq p(|x|, \max(x))$  for any input  $x$ , where  $m^*(x)$  denotes the value of any optimal solution of  $x$  and  $|x|$  denotes the length of  $x$ . If  $\mathbf{P} \neq \mathbf{NP}$ , then  $\mathcal{P}$  does not have any FPTAS.*

**Theorem 3.1.** *If (1,2)-HPCT has an FPTAS, then  $\mathbf{NP} = \mathbf{P}$ .*

**Proof.** Given any instance  $x$  of (1,2)-HPCT with  $G = (V, E)$ , the optimal augment  $E^*$  will not contain all edges in  $E$ , which implies that  $m^*(x) < 2|E|$ . Since it is bounded by a polynomial, by Observation 3.1 and Lemma 3.1, if (1,2)-HPCT has an FPTAS, then  $\mathbf{NP} = \mathbf{P}$ .  $\square$

#### 4. A 1.5-approximation algorithm for (1,2)-hamiltonian path completion problem on 1-star

The *1-star* is a complete bipartite graph  $K_{1,n}$ , which is a tree with  $n$  leaf vertices and 1 non-leaf vertex. In this section, we shall give a 1.5-approximation algorithm for the (1,2)-Hamiltonian path completion problem on a 1-star. Before that, we would like to mention here that the minimum-weight maximal matching problem in a weighted complete graph, by which we adopt to design the approximation algorithms throughout the rest of this paper, can be solved in polynomial time, in contrast to the fact that the minimum-weight maximal matching problem is NP-hard for general graphs [9].

**Lemma 4.1.** *The minimum-weight maximal matching problem in a weighted complete graph can be solved in polynomial time.*

**Proof.** Suppose that the minimum-weight maximal matching problem is given in a weighted complete graph  $G = (V, E)$ , with  $w(e)$  denoting the weight on each edge  $e \in E$ . Let  $\Delta$  be a positive real number greater than the weight of any edge in  $G$ . This problem can be reduced to the maximum-weight maximal matching problem in a weighted complete graph  $G' = (V', E')$  with  $V' = V$ ,  $E' = E$  and  $w' = \Delta - w(e)$ . Clearly, a maximal matching in  $G$  is also a maximal matching in  $G'$ , and vice versa. Moreover, all the maximal matchings in  $G$  and  $G'$  contain exactly  $\lfloor n/2 \rfloor$  edges, where  $n = |V|$ . Let  $w(M) = \sum_{e \in M} w(e)$  for a maximal matching  $M$ . For any two maximum matchings  $M_1$  and  $M_2$  in  $G$ , we use  $M'_1$  and  $M'_2$  to denote their corresponding maximal matchings in  $G'$ , respectively. Then for  $i \in \{1, 2\}$ ,  $w(M'_i) = \lfloor n/2 \rfloor \times \Delta - w(M_i)$ . Consequently,  $w(M'_1) \geq w(M'_2)$  if and only if  $w(M_1) \leq w(M_2)$ . In other words, the maximal matching with maximum weight in  $G'$  corresponds to a maximal matching with minimum weight in  $G$ , and vice versa. Note that the maximum-weight maximal matching problem for general graphs can be solved in  $O(n^3)$  time [14, Chapter 6]. Hence, as discussed above, the minimum-weight maximal matching problem in a weighted complete graph is solvable in  $O(n^3)$  time.<sup>1</sup>  $\square$

<sup>1</sup> It is worth noticing that if the considered graph is not complete, then the above reduction does not work since not all the maximal matchings have the same cardinalities.

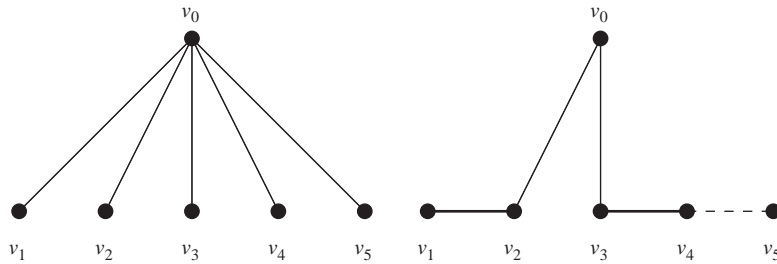


Fig. 4. Finding the minimum-weight maximal matching on a 1-star.

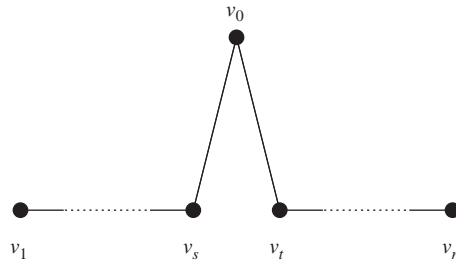


Fig. 5. A Hamiltonian path in  $G' = (V, E_0 \cup E_2^*)$ .

Now, we describe our approximation algorithm for solving the (1,2)-Hamiltonian path completion problem on a 1-star. Suppose that  $G = (V, E)$  is the given graph of the problem, where  $V = \{v_0, v_1, v_2, \dots, v_n\}$  and  $v_0$  is the root of the given 1-star. Our approximation algorithm will first find a minimum-weight maximal matching in the induced complete graph by  $V' = V \setminus \{v_0\}$ , and then add edges to concatenate these matching pairs into a Hamiltonian path. For example, in Fig. 4, the matching may find  $\{(v_1, v_2), (v_3, v_4)\}$ , then in the second step  $(v_4, v_5)$  is added to form a Hamiltonian path. The formal description of our algorithm is as follows.

**Algorithm 1**

1. **if**  $n \leq 2$ , **then** return  $E_2 = \emptyset$ ;
2. Perform a minimum-weight maximal matching algorithm in the induced graph by  $V'$ ;  
Suppose the matching is  $\{(v_{i_1}, v_{j_1}), (v_{i_2}, v_{j_2}), \dots, (v_{i_{\lfloor n/2 \rfloor}}, v_{j_{\lfloor n/2 \rfloor})\}$  and denote by  $v_{i_{\lfloor n/2 \rfloor}}$  the unmatched vertex if  $n$  is odd.
3. Return  $E_2 = \left(\bigcup_{k=1}^{\lfloor n/2 \rfloor} \{(v_{i_k}, v_{j_k})\}\right) \cup \left(\bigcup_{k=2}^{\lfloor n/2 \rfloor - 1} \{(v_{j_k}, v_{i_{k+1}})\}\right)$ ;

**Lemma 4.2.** *If the optimal augment  $E_2^*$  contains  $k$  edges with weight 1, then our algorithm finds at least  $k/2$  of edges with weight 1 in the solution.*

**Proof.** It can be seen that for  $K_{1,n}$ , any optimal augment  $E_2^*$  always has exactly  $n - 2$  edges. Moreover, in  $G' = (V, E_0 \cup E_2^*)$ , there exist two edges  $(v_0, v_s)$  and  $(v_0, v_t)$  in  $E_0$  such that they together with these  $n - 2$  edges in  $E_2^*$  constitute a Hamiltonian path, as illustrated in Fig. 5. By deleting the vertex  $v_0$  from the path, we obtain two vertex-disjoint

paths to cover  $V' = \{v_1, v_2, \dots, v_n\}$ . (Please note that a single vertex is also regarded as a degenerated path here.) Connecting these two paths with “head to head” and “tail to tail” by adding the two corresponding edges, we obtain a cycle  $C$ . Obviously  $E_2^* \subset C$ . Without loss of generality, let us label this cycle as  $\{(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n), (v_n, v_1)\}$ , where  $(v_s, v_t)$  and  $(v_n, v_1)$  are the two added edges. Let MM be the minimum-weight maximal matching obtained in our algorithm.

*Case 1.* If  $n$  is even, then each maximal matching has  $n/2$  edges since the given graph is complete. It can be seen that  $M_1 = \{(v_1, v_2), (v_3, v_4), \dots, (v_{n-1}, v_n)\}$  is a maximal matching and  $M_2 = \{(v_2, v_3), (v_4, v_5), \dots, (v_n, v_1)\}$  is also a maximal matching. Suppose MM contains  $k_1$  edges with weight 1. Then both  $M_1$  and  $M_2$  contain at most  $k_1$  edges with weight 1; otherwise, MM cannot be minimum. Thus,  $C = M_1 \cup M_2$  contains at most  $2k_1$  edges with weight 1. If  $k_1 < k/2$ , then  $2k_1 < k$  and  $C$  contains fewer than  $k$  edges with weight 1. This contradicts the assumption that the subset  $E_2^*$  of  $C$  already has  $k$  edges with weight 1. Therefore,  $k_1 \geq k/2$ .

*Case 2.* If  $n$  is odd, then each maximal matching has  $n - 1/2$  edges. It can be seen that  $M_1 = \{(v_1, v_2), (v_3, v_4), \dots, (v_{n-2}, v_{n-1})\}$  is a maximal matching and  $M_2 = \{(v_2, v_3), (v_4, v_5), \dots, (v_{n-1}, v_n)\}$  is a maximal matching, too. Suppose MM contains  $k_1$  edges with weight 1. Then with similar reasoning, both  $M_1$  and  $M_2$  will have less than or equal to  $k_1$  edges with weight 1. Again,  $k_1 < k/2$  implies that the path  $\{(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)\}$  contains fewer than  $k$  edges with weight 1. This also causes a contradiction because its subset  $E_2^*$  already contains  $k$  edges with weight 1. Therefore,  $k_1 \geq k/2$ .

In either case, our algorithm finds at least  $k/2$  edges with weight 1 in the minimum-weight maximal matching MM.

**Theorem 4.1.** *The performance ratio of Algorithm 1 is  $\frac{3}{2}$ .*

**Proof.** The augment  $E_2$  obtained by our algorithm contains  $\lfloor n/2 \rfloor + \lceil n/2 \rceil - 2 = n - 2$  edges, just the same as the optimal augment. Suppose the optimal solution  $E_2^*$  contains  $k$  edges with weight 1, and  $h$  edges with weight 2 (i.e.,  $k + h = n - 2$ ). According to Lemma 4.2, our approximation algorithm which performs minimum-weight maximal matching to get a partial result will choose at least  $k/2$  edges with weight 1. Even in the worst case that all the other edges added later are with weight 2, the performance ratio of our approximate solution will be

$$\frac{\frac{k}{2} + 2(\frac{k}{2} + h)}{k + 2h} = \frac{\frac{3k}{2} + 2h}{k + 2h} \leq \frac{3}{2}.$$

**Remarks.** Precisely, by Lemma 4.2, the above formula should be written as

$$\frac{\lceil \frac{k}{2} \rceil + 2(k + h - \lceil \frac{k}{2} \rceil)}{k + 2h}.$$

When  $k$  is odd, it becomes

$$\frac{\frac{k+1}{2} + 2(k + h - \frac{k+1}{2})}{k + 2h} = \frac{\frac{3k-1}{2} + 2h}{k + 2h} < \frac{\frac{3k}{2} + 2h}{k + 2h} \leq \frac{3}{2}. \quad \square$$



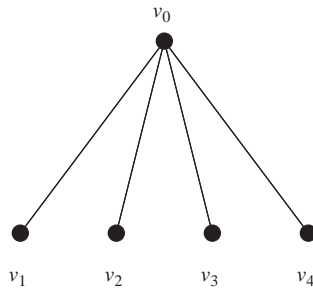


Fig. 6. An example showing that ratio 1.5 is tight.

The time complexity of Algorithm 1 is analyzed as follows. Let  $n + 1$  be the number of vertices in  $G$ . The minimum-weight maximal matching in Step 2 takes  $O(n^3)$  time, as mentioned in the beginning of this section. Therefore, we obtain a 1.5-approximation algorithm to solve the (1,2)-Hamiltonian path completion problem on 1-star that runs in  $O(n^3)$  time.

To show that the analysis for the performance ratio of this algorithm is tight, let us see the example in Fig. 6, where the edge weights are

$$w(e) = \begin{cases} 1 & \text{if } e \in \{(v_1, v_2), (v_1, v_3)\}, \\ 2 & \text{otherwise.} \end{cases}$$

Our approximation algorithm will find a minimum-weight maximal matching first, say  $\{(v_1, v_2), (v_3, v_4)\}$ , and adopts this as the approximate solution. Its weight will be 3. However, the optimal augment is  $\{(v_1, v_2), (v_1, v_3)\}$ , whose weight is 2. The performance ratio then is  $\frac{3}{2} = 1.5$ .

### 5. A 1.5-approximation algorithm for (1,2)-Hamiltonian path completion problem on $k$ -star

A tree with  $k$  internal vertices is called a  $k$ -star. In this section, we are going to show that the approximation algorithm developed in the previous section can be extended to obtain similar results on  $k$ -stars. Let us introduce some notation first.

**Definition 5.1.** Suppose that  $G = (V, E)$  is a weighted graph with weight function  $w(e)$  defined on all edges  $e$  in  $E$  and let  $P = (u_1, u_2, \dots, u_k)$  be a path in  $G$ . Then the new weighted graph  $G' = (V', E')$  obtained from  $G$  by shrinking  $P$  into a single vertex is defined as follows, where  $d(u, v) = w(e)$  for an edge  $e = (u, v)$ .

- $V' = V \setminus \{u_2, u_3, \dots, u_k\}$ .
- $E' = E \setminus \bigcup_{v \in V, u \in \{u_2, u_3, \dots, u_k\}} \{(u, v)\}$ .
- $w'(e) = \begin{cases} \min\{d(u_1, v), d(u_k, v)\} & \text{if } e = (u_1, v) \text{ for some } v \in V', \\ w(e) & \text{otherwise.} \end{cases}$

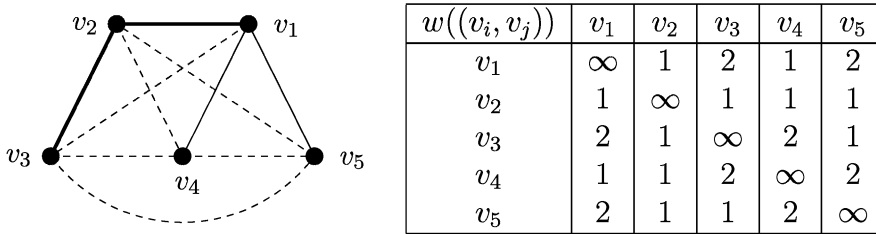


Fig. 7. The path  $(v_1, v_2, v_3)$  of  $G$  is going to be shrunk to a vertex  $v_1$ .

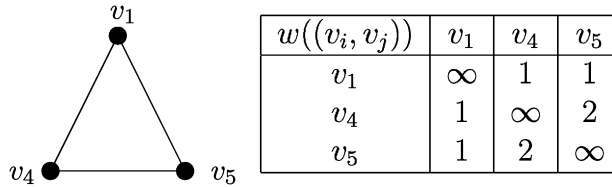


Fig. 8. The resulting graph  $G_{\{P\}}$ , where  $P$  is the path  $(v_1, v_2, v_3)$  of Fig. 7.

Let  $G_{\{P\}}$  denote the graph obtained from  $G$  by shrinking the path  $P$ . For two vertex-disjoint paths  $P_1$  and  $P_2$ , the result of shrinking will be the same no matter  $P_1$  is shrunk first or  $P_2$  is shrunk first, because the  $\min$  operation satisfies the associative law. That is,  $(G_{\{P_1\}})_{\{P_2\}} = (G_{\{P_2\}})_{\{P_1\}}$ . Therefore, it is sound to simply write it as  $G_{\{P_1, P_2\}}$ . For example, consider the complete graph  $G$  as shown in Fig. 7, whose distances between any two vertices are specified in the table aside. After shrinking the path  $P = (v_1, v_2, v_3)$  of  $G$ , the resulting graph  $G_{\{P\}}$  is shown in Fig. 8.

In the rest of this section, we suppose that  $G = (V, E)$  and  $E_0$  are the instance of the (1,2)-HPCT problem, where  $G$  is a weighted complete graph and  $E_0$  constitutes a  $k$ -tree of  $G$ . In  $G$ , we call the edges in the given subset  $E_0$  the  $e_0$ -edges, and edges in  $E \setminus E_0$  with weight 1 the  $e_1$ -edges, and edges in  $E \setminus E_0$  with weight 2 the  $e_2$ -edges.

**Observation 5.1.** Suppose that  $H$  is a Hamiltonian path in  $G$  and  $P_1, P_2, \dots, P_i$  are the vertex-disjoint paths in  $H \cap E_0$ . If we focus on edges in  $H$ , it can be observed that during the process of shrinking  $P_1, P_2, \dots, P_i$ , only  $e_0$ -edges will be deleted, and some  $e_2$ -edges may be turned to  $e_1$ -edges. Therefore,  $H_{\{P_1, P_2, \dots, P_i\}}$  contains at least  $n_1$  edges with weight 1 if  $H$  contains  $n_1$   $e_1$ -edges.

**Lemma 5.1.** If a Hamiltonian path  $H$  in graph  $G$  contains  $n_1$   $e_1$ -edges, then a minimum-weight maximal matching in  $G$  contains at least  $n_1/2$   $e_1$ -edges.

**Proof.** The proof is similar to that of Lemma 4.2.  $\square$

**Theorem 5.1.** If a Hamiltonian path  $H$  contains  $n_1$   $e_1$ -edges and  $H \cap E_0$  consists of some vertex-disjoint paths  $P_1, P_2, \dots, P_i$ , then a minimum-weight maximal matching on  $G_{\{P_1, P_2, \dots, P_i\}}$  contains at least  $\frac{n_1}{2}$   $e_1$ -edges.

**Proof.** By Observation 5.1 and Lemma 5.1.  $\square$

Informally speaking, our approximation algorithm will first find a minimum-weight maximal matching on the shrunk complete graph  $G_{\{P_1, P_2, \dots, P_i\}}$ , and then map these matching edges back to  $G$ . By our construction of  $G_{\{P_1, P_2, \dots, P_i\}}$ , each edge in  $G_{\{P_1, P_2, \dots, P_i\}}$  corresponds to an edge in  $G$ , and each vertex in  $G_{\{P_1, P_2, \dots, P_i\}}$  corresponds to either a single vertex or a path in  $G$ . Therefore, each path in  $G_{\{P_1, P_2, \dots, P_i\}}$  will be mapped to a path (containing one or more edges) in  $G$ . These paths are then concatenated serially to form a Hamiltonian path. Note that if  $u$  and  $v$  are the terminals of path  $P_j$  for  $1 \leq j \leq i$ , the mapping of the matching edge  $e' = (x, u)$  in  $G_{\{P_1, P_2, \dots, P_i\}}$  back to the edge  $e$  in  $G$  is done according to the following equation.

$$e = \begin{cases} (x, u) & \text{if } d(x, u) \text{ in } G \text{ is equal to } d(x, u) \text{ in } G_{\{P_1, P_2, \dots, P_i\}}, \\ (x, v) & \text{otherwise.} \end{cases}$$

Therefore, if  $H$  contains  $n_0$   $e_0$ -edges,  $n_1$   $e_1$ -edges,  $n_2$   $e_2$ -edges, then the result obtained by our approximation algorithm will also contain the same  $n_0$   $e_0$ -edges, and at least  $n_1/2$  edges with weight 1 (contributed by the minimum-weight maximal matching) according to Theorem 5.1. As we have seen in the previous section, the weight of this approximate result will be less than or equal to  $n_1/2 + 2 \times (n_1 + n_2 - n_1/2) = \frac{3}{2}n_1 + 2n_2$ . Therefore, the performance ratio is

$$\frac{\frac{3}{2}n_1 + 2n_2}{n_1 + 2n_2} \leq \frac{3}{2}.$$

Now we have a natural question: How does our approximation algorithm know what paths in  $E_0$  must be chosen to shrink? If we could choose the ones exactly as an optimal path  $H^*$  contains, then the cost of the solution obtained by our approximation algorithm is guaranteed to be within  $\frac{3}{2}$  times the one of the optimal solution. However, we do not know what edges may be contained in  $H^*$  and what may not. If we have to try each possibility, since there are  $O(n)$   $e_0$ -edges, trying all the  $O(2^n)$  combinations will lead to an exponential algorithm. Fortunately, we have the following lemma:

**Lemma 5.2.** *If  $H$  is a Hamiltonian path in  $G$ , then  $H$  contains at most  $2k$   $e_0$ -edges.*

**Proof.** Since  $E_0$  constitutes a  $k$ -star in  $G$ , all  $e_0$ -edges must be incident to the  $k$  internal vertices of this  $k$ -star. However, in Hamiltonian path  $H$ , at most two edges are incident to each internal vertex, so at most  $2k$   $e_0$ -edges are contained in this path.  $\square$

Therefore, we only have to test all combinations that contain 0  $e_0$ -edges, 1  $e_0$ -edges, 2  $e_0$ -edges, . . . ,  $2k$   $e_0$ -edges. For each combination, we apply the shrink operation and find an approximate solution of it. The minimum of these solutions will be chosen as our final approximate solution to be reported. Therefore, let us state our algorithm formally below.

**Algorithm 2**

1.  $W \leftarrow \infty, AUG \leftarrow \emptyset$ .
2. **for** all subsets of  $E_0$  with no more than  $2k$  edges **do**

- 2.1. **if** the subset has 3 or more edges incident to the same vertex **then**  
 /\* Not vertex-disjoint paths. Do nothing. \*/
- 2.2. **else**  
 Suppose the subset consists of vertex-disjoint paths  $P_1, P_2, \dots, P_i$ .
- 2.2.1. Shrink paths  $P_1, P_2, \dots, P_i$  to obtain  $G_{\{P_1, P_2, \dots, P_i\}}$ ;
- 2.2.2. Find a minimum-weight maximal matching MM on  $G_{\{P_1, P_2, \dots, P_i\}}$ ;
- 2.2.3. Map these matching edges to paths in  $G$ ;  
 /\* Let MM be mapped to MM'. \*/
- 2.2.4. Add edges (denoted by  $S$ ) to concatenate these paths serially to form a Hamiltonian path;
- 2.2.5. **if** the weight of  $S \cup MM'$  is smaller than  $W$  **then**  
 $W \leftarrow w(S \cup MM')$ ,  $AUG \leftarrow S \cup MM'$ ;
3. Report AUG as the solution and stop;

There are  $O(n^{2k})$  iterations in Step 2 of Algorithm 2, and each iteration takes  $O(n^3)$  time as we have seen in the previous section. Hence, we obtain a 1.5-approximation algorithm to solve the (1,2)-Hamiltonian path completion problem on  $k$ -stars that runs in  $O(n^{2k+3})$  time.

## 6. Conclusions

In this paper, we introduced the weighted Hamiltonian path completion problem and showed that this problem is hard to approximate, even if the given edge set is a tree. We also showed that the weighted Hamiltonian path completion problem remains NP-hard when the edge weights are restricted to be either 1 or 2. We then observed that this problem is strongly NP-hard, so it is unlikely to have any FPTAS. When the given tree has  $k$  internal vertices, we gave an approximation algorithm with performance ratio  $\frac{3}{2}$  whose time complexity is polynomial when  $k$  is fixed.

Some version of Hamiltonian completion problems finds the augment to make the given graph having a Hamiltonian circuit instead of a Hamiltonian path. It can be checked that our results can be applied to obtain the same results on Hamiltonian cycle completion problems.

Although (1,2)-HPCT is unlikely to have any FPTAS, it is still unknown whether it has a PTAS. Another variation which deserves further study is for general trees. In [23], an approximation algorithm for (1,2)-HPCT with performance ratio 2 was proposed for general trees. However, whether there exists a PTAS or an approximation algorithm with performance ratio less than 2 as we derived for  $k$ -stars is still unknown.

## Acknowledgements

The authors would like to thank the anonymous referees for many constructive suggestions for the presentation of this paper.

## References

- [1] S.R. Arikati, C. Pandu Rangan, Linear algorithm for optimal path cover problem on interval graphs, *Inform. Process. Lett.* 35 (1990) 149–153.
- [2] S. Arora, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, in: *Proc. 37th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Comp. Soc. 1996, pp. 2–11.
- [3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, Springer, Berlin Heidelberg, 1999.
- [4] B.S. Baker, Approximation algorithms for NP-complete problems on planar graphs, *J. ACM* 41 (1994) 153–180.
- [5] A. Blum, M. Li, J. Tromp, M. Yannakakis, Linear approximation of shortest superstrings, *J. ACM* 41 (1994) 630–647.
- [6] F.T. Boesch, S. Chen, J.A.M. McHugh, On covering the points of a graph with point disjoint paths, in: R.A. Bari, F. Harary (Eds.), *Proc. Capital Conf. on Graph Theory and Combinatorics*, Lecture Notes in Math. Vol. 46, Springer, Berlin, 1974, pp. 201–212.
- [7] M.A. Bonucelli, D.P. Bovet, Minimum node disjoint path covering for circular-arc graphs, *Inform. Process. Lett.* 8 (1979) 159–161.
- [8] G. Galbiati, F. Maffioli, A. Morzenti, A short note on the approximability of the maximum leaves spanning tree problem, *Inform. Process. Lett.* 52 (1994) 45–49.
- [9] M. Garey, D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.
- [10] S.E. Goodman, S.T. Hedetniemi, On the Hamiltonian completion problem, in: R.A. Bari, F. Harary (Eds.), *Proc. Capital Conf. on Graph Theory and Combinatorics*, Lecture Notes in Math., Vol. 46, Springer, Berlin, 1974, pp. 262–272.
- [11] S.E. Goodman, S.T. Hedetniemi, P.J. Slater, Advances on the Hamiltonian completion problem, *J. ACM* 22 (1975) 352–360.
- [12] O.H. Ibarra, C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *J. ACM* 22 (1975) 463–468.
- [13] S. Kundu, A linear algorithm for the Hamiltonian completion number of a tree, *Inform. Process. Lett.* 5 (1976) 55–57.
- [14] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, New York, 1976.
- [15] R. Lin, S. Olariu, G. Pruesse, An optimal path cover algorithm for cographs, *Comput. Math. Appl.* 30 (1995) 75–83.
- [16] J. Misra, R.E. Tarjan, Optimal chain partitions of trees, *Inform. Process. Lett.* 4 (1975) 24–26.
- [17] C.H. Papadimitriou, M. Yannakakis, Optimization approximation and complexity classes, *J. Comput. System Sci.* 43 (1991) 425–440.
- [18] C.H. Papadimitriou, M. Yannakakis, The traveling salesman problem with distances one and two, *Math. Oper. Res.* 18 (1993) 1–11.
- [19] S. Sahni, T. Gonzalez, *P*-complete approximation problems, *J. ACM* 23 (1976) 555–565.
- [20] Z. Skupien, Path partitions of vertices and Hamiltonicity of graphs, in: *Proc. second Czechoslovakian Symposium on Graph Theory*, Prague, 1974, pp. 481–491.
- [21] R. Srikant, R. Sundaram, K.S. Singh, C.P. Rangan, Optimal path cover problem on block graphs and bipartite permutation graphs, *Theoret. Comput. Sci.* 115 (1993) 351–357.
- [22] P.-K. Wong, Optimal path cover problem on block graphs, *Theoret. Comput. Sci.* 225 (1999) 163–169.
- [23] Q. Wu, *On the Complexity and Approximability of Some Hamiltonian Path Problems*, Ph.D. Thesis, National Tsing Hua University, Taiwan, 2000.
- [24] J.H. Yan, G.J. Chang, The path-partition problem in block graphs, *Inform. Process. Lett.* 52 (1994) 317–322.