Algorithmica manuscript No.
(will be inserted by the editor)

Route-Enabling Graph Orientation Problems

Takehiro Ito · Yuichiro Miyamoto ·
Hirotaka Ono · Hisao Tamaki ·
Ryuhei Uehara

Received: date / Accepted: date

Abstract Given an undirected and edge-weighted graph G together with a set of ordered vertex-pairs, called st-pairs, we consider two problems of finding an orientation of all edges in G: MIN-SUM ORIENTATION is to minimize the sum of the shortest directed distances between all st-pairs; and MIN-MAX ORIENTATION is to minimize the maximum shortest directed distance among all st-pairs. Note that these shortest directed paths for st-pairs are not necessarily edge-disjoint. In this paper, we first show that both problems are strongly NP-hard for planar graphs even if all edge-weights are identical, and that both problems can be solved in polynomial time for cycles. We then consider the problems restricted to cacti, which form a graph class that contains trees and cycles but is a subclass of planar graphs. Then, MIN-SUM ORIENTATION is solvable

An extended abstract of this paper has been presented at the 20th International Symposium on Algorithms and Computation (ISAAC 2009) [7].

This work is partially supported by Grant-in-Aid for Scientific Research.

T. Ito
Graduate School of Information Sciences, Tohoku University,
Aoba-yama 6-6-05, Sendai, 980-8579, Japan
E-mail: takehiro@ecei.tohoku.ac.jp

Y. Miyamoto
Faculty of Science and Technology, Sophia University,
Kioi-cho 7-1, Chiyoda-ku, Tokyo, 102-8554, Japan
E-mail: miyamoto@sophia.ac.jp

H. Ono
Faculty of Economics, Kyushu University,
6-19-1 Hakozaki, Higashi-ku, Fukuoka, 812-8581, Japan
E-mail: ono@csce.kyushu-u.ac.jp

H. Tamaki
School of Science and Technology, Meiji University,
Higashi-mita 1-1-1, Tama-ku, Kawasaki-shi, Kanagawa, 214-8571, Japan
E-mail: tamaki@cs.meiji.ac.jp

R. Uehara
School of Information Science, JAIST,
Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan
E-mail: uehara@jaist.ac.jp

in polynomial time, whereas MIN-MAX ORIENTATION remains NP-hard even for two st-pairs. However, based on LP-relaxation, we present a polynomial-time 2-approximation algorithm for MIN-MAX ORIENTATION. Finally, we give a fully polynomial-time approximation scheme (FPTAS) for MIN-MAX ORIENTATION on cacti if the number of st-pairs is a fixed constant.

Keywords approximation algorithm · cactus · dynamic programming · fully polynomial-time approximation scheme · graph orientation · planar graph · reachability

1 Introduction

Consider the situation in which we wish to assign one-way restrictions to (narrow) aisles in a limited area, such as in an industrial factory, with keeping reachability between several sites. Since traffic jams rarely occur in industrial factories, the distances of routes between important sites directly affect transit time, productivity, etc. This situation frequently appears in the context of the scheduling of automated guided vehicles without collision [8,9]. In this paper, we model this situation as graph orientation problems, in which we wish to find an orientation so that the distances of (directed) routes are not so long for given multiple st-pairs.

Let $G = (V, E)$ be an undirected graph together with an assignment of a non-negative integer, called the *weight* $\omega(e)$, to each edge e in G. Assume that we are given q ordered vertex-pairs (s_i, t_i), $1 \le i \le q$, called *st-pairs*. Then, an *orientation* of G is an assignment of exactly one direction to each edge in G so that there exists a directed (s_i, t_i)-path (*i.e.*, a directed path from s_i to t_i) for every st-pair (s_i, t_i), $1 \le i \le q$. Note that these directed (s_i, t_i)-paths, $1 \le i \le q$, are not necessarily edge-disjoint, that is, some of directed (s_i, t_i)-paths may share an edge (passing through the same direction). We denote by $\mathbf{G}$ an orientation of G. For an orientation $\mathbf{G}$ of G and an st-pair (s_i, t_i), we denote by $\omega(\mathbf{G}, s_i, t_i)$ the total weight of a shortest directed (s_i, t_i)-path in $\mathbf{G}$, that is,

$$\omega(\mathbf{G}, s_i, t_i) = \min\{\omega(P) \mid P \text{ is a directed } (s_i, t_i)\text{-path in } \mathbf{G}\}$$

where $\omega(P)$ is the sum of weights of all edges in a path P. We introduce two objective functions for orientations $\mathbf{G}$ of a graph G, and study the corresponding two minimization problems. The first objective is of SUM-type, defined as follows: $g(\mathbf{G}) = \sum_{1 \le i \le q} \omega(\mathbf{G}, s_i, t_i)$. Its corresponding problem, called the MIN-SUM ORIENTATION problem, is to find an orientation $\mathbf{G}$ of G such that $g(\mathbf{G})$ is minimum; we denote by $g^*(G)$ the optimal value for G. The second objective is of MAX-type, defined as follows:

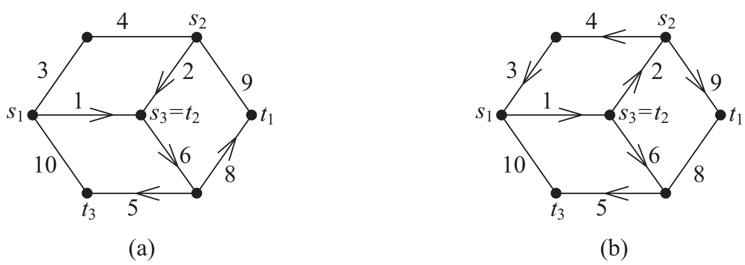

Fig. 1 (a) Solution for MIN-SUM ORIENTATION and (b) solution for MIN-MAX ORIENTATION.

Table 1 Summary of our results.

	MIN-SUM ORIENTATION	MIN-MAX ORIENTATION
planar graphs	• strongly NP-hard	• strongly NP-hard • no $(2-\varepsilon)$-approximation
cacti	$O(nq^2)$	• NP-hard even for $q = 2$ • polynomial-time 2-approximation • FPTAS for a fixed constant q
cycles	$O(n + q^2)$	$O(n + q^2)$

$h(\mathbf{G}) = \max\{\omega(\mathbf{G}, s_i, t_i) \mid 1 \leq i \leq q\}$. Its corresponding problem, called the MIN-MAX ORIENTATION problem, is to find an orientation $\mathbf{G}$ of G such that $h(\mathbf{G})$ is minimum; we denote by $h^*(G)$ the optimal value for G. Let $g^*(G) = +\infty$ and $h^*(G) = +\infty$ if there is no orientation for G that contains a directed (s_i, t_i)-path for every st-pair (s_i, t_i), $1 \leq i \leq q$.

Figure 1 illustrates two orientations of the same graph G for the same set of st-pairs, where the weight $\omega(e)$ is attached to each edge e and the direction assigned to an edge is indicated by an arrow (but the directions are not indicated for the edges that are not used in any shortest directed (s_i, t_i)-path, $1 \leq i \leq 3$). The orientation $\mathbf{G}$ in Fig. 1(a) is an optimal solution for MIN-SUM ORIENTATION, where $g^*(G) = g(\mathbf{G}) = (1 + 6 + 8) + 2 + (6 + 5) = 28$. On the other hand, Fig. 1(b) illustrates an optimal solution for MIN-MAX ORIENTATION, in which the st-pair (s_1, t_1) has the maximum distance; $h^*(G) = \max\{1 + 2 + 9,\ 4 + 3 + 1,\ 6 + 5\} = 12$.

Obviously, both problems can be solved in polynomial time if we are given a single st-pair (s_1, t_1); in this case, we simply seek a shortest path between s_1 and t_1. Robbins [12] showed that every 2-edge-connected graph can be directed so that the resulting digraph is strongly connected. Therefore, a graph G has at least one orientation for any set of st-pairs if G is 2-edge-connected. Chvátal and Thomassen [2] showed that it is NP-complete to determine whether a given unweighted graph can be directed so that the resulting digraph is strongly connected and whose (directed) diameter is 2. This implies that our MIN-MAX ORIENTATION is NP-hard in general. In contrast, Eggemann and Noble [3] showed that, for every fixed constant l, it can be determined in linear time whether a given planar graph has an orientation such that the resulting graph is strongly connected with directed diameter at most l. (The hidden coefficient of their running time is exponential in l.) Medvedovsky *et al.* [10] studied the problem of directing a 1-edge-connected graph so as to maximize the number of st-pairs (s_i, t_i) having a directed (s_i, t_i)-path for a given set of st-pairs. They showed that the problem is NP-hard in general, while Hakimi *et al.* [6] proposed a quadratic-time algorithm for the case where the given set of st-pairs consists of all ordered vertex-pairs $V \times V$.

In this paper, we mainly give the following three results. (Table1 summarizes our results, where n is the number of vertices in a graph.) The first is to show the computational hardness of our problems. Specifically, we show that both problems are strongly NP-hard for planar graphs even if all edge-weights are identical. We remark that the known result of [2] does not imply NP-hardness for planar graphs. The second is to show that both problems can be solved in polynomial time for cycles. By extending the algorithm for cycles, we then show that MIN-SUM ORIENTATION is solvable in polynomial time for cacti, whereas MIN-MAX ORIENTATION remains NP-hard even for cacti with $q = 2$. (Cacti form a graph class that contains trees and cycles, but is a subclass of planar graphs; a formal definition of cacti will be given in Section 2.2.) The third is to give a fully polynomial-time approximation scheme (FPTAS) for MIN-MAX ORIENTATION on cacti if q is a fixed constant; the polynomial running time depends exponentially on q.

In addition, we give several results on the way to the three main results above. Firstly, our proof of strong NP-hardness implies that, for any constant $\varepsilon > 0$, MIN-MAX ORIENTATION admits no polynomial-time $(2-\varepsilon)$-approximation algorithm unless P = NP. Secondly, in order to obtain both lower and upper bounds on $h^*(G)$ for a cactus G, we present a polynomial-time 2-approximation algorithm based on LP-relaxation; we remark that q is not required to be a fixed constant for this 2-approximation algorithm. We finally remark that our complexity analysis for MIN-MAX ORIENTATION on cacti is tight in the following sense: the problem is in P if $q = 1$, but is NP-hard for $q = 2$; moreover, our third result implies that the problem for cacti cannot be strongly NP-hard if q is a fixed constant [11, p. 307].

2 Computational Hardness

In this section, we show the computational hardness of our problems. In Section 2.1, we first show that our two problems are both strongly NP-hard for planar graphs. We then show in Section 2.2 that MIN-MAX ORIENTATION remains NP-hard even for cacti with $q = 2$.

2.1 Strongly NP-hardness for planar graphs

We first give the following theorem for MIN-MAX ORIENTATION.

Theorem 1 MIN-MAX ORIENTATION *is strongly NP-hard for planar graphs of maximum degree* 4 *even if all edge-weights are identical.*

Proof We show that the PLANAR 3-SAT problem, which is known to be strongly NP-complete [4, p. 259], can be reduced in polynomial time to the MIN-MAX ORIENTATION problem for planar graphs.

In PLANAR 3-SAT, we are given a Boolean formula ϕ in conjunctive normal form, say with set U of n variables $u_1, u_2, \ldots, u_n$ and set C of m clauses $c_1, c_2, \ldots, c_m$, such that each clause $c_j \in C$ contains exactly three literals and the following bipartite graph $B = (V', E')$ is planar: $V' = U \cup C$ and E' contains exactly those pairs $\{u_i, c_j\}$ such that either u_i or $\bar{u}_i$ appears in c_j. The PLANAR 3-SAT problem is to determine whether there is a satisfying truth assignment for ϕ.

Given an instance of PLANAR 3-SAT, we construct the corresponding instance of MIN-MAX ORIENTATION. We first make a *flower gadget* $F_i(M)$ for each variable $u_i \in U$, and then construct the whole graph G_ϕ corresponding to ϕ.

Flower gadget $F_i(M)$

We first define a flower gadget $F_i(M)$ for each variable $u_i \in U$. Let M be a fixed constant (integer) such that $M \geq 3$. (We here introduce the constant M, instead of specifying $M = 3$, to prove Corollary 1 later.) The flower gadget $F_i(M) = (V_i, E_i)$ consists of $2m$ hexagonal elementary cycles, as illustrated in Fig. 2(a). (Remember that m is the number of clauses in ϕ.) More precisely, $V_i = \{a_k, b_k, c_k, d_k \mid 1 \leq k \leq 2m\}$ and $E_i = \{\{a_{k+1}, a_k\}, \{a_k, b_k\}, \{b_k, c_k\}, \{c_k, d_k\}, \{d_k, b_{k+1}\} \mid 1 \leq k \leq 2m\}$, where

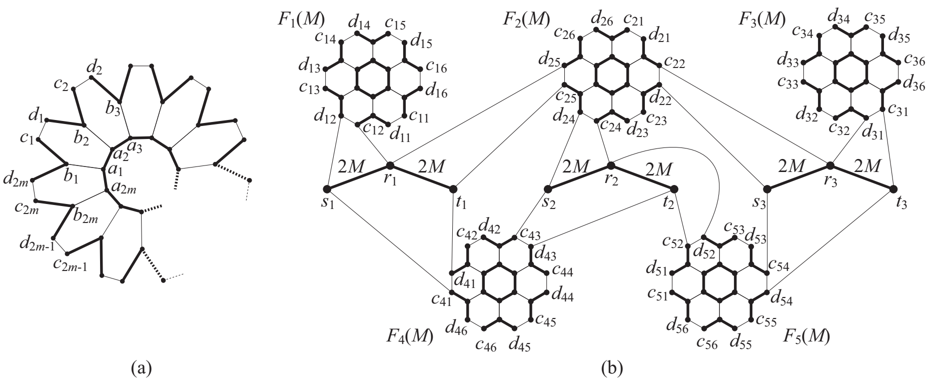

Fig. 2 (a) Flower gadget $F_i(M)$, and (b) planar graph G_ϕ corresponding to a Boolean formula ϕ with three clauses $c_1 = (u_1 \vee \bar{u}_2 \vee u_4)$, $c_2 = (u_2 \vee u_5 \vee u_4)$ and $c_3 = (u_2 \vee \bar{u}_3 \vee \bar{u}_5)$.

$a_{2m+1} = a_1$ and $b_{2m+1} = b_1$. The edge-weights are defined as follows: for each k, $1 \le k \le 2m$, $\omega(\{a_{k+1}, a_k\}) = \omega(\{b_k, c_k\}) = \omega(\{d_k, b_{k+1}\}) = M$ and $\omega(\{a_k, b_k\}) = \omega(\{c_k, d_k\}) = 1$. (In Fig. 2(a), the weight-M edges are depicted by thick lines.) Finally, we define the set ST_i of $12m$ st-pairs, as follows:

$$ST_i = \{(a_k, d_k), (d_k, a_k), (b_k, b_{k+1}), (b_{k+1}, b_k), (c_k, a_{k+1}), (a_{k+1}, c_k) \,|\, 1 \le k \le 2m\}.$$

For each k, $1 \le k \le 2m$, the kth hexagonal elementary cycle $a_k b_k c_k d_k b_{k+1} a_{k+1}$ is called the *kth petal* P_k; P_k is called an *odd petal* if k is odd, while is called an *even petal* if k is even. We call the edge $\{c_k, d_k\}$ in each petal P_k, $1 \le k \le 2m$, an *external edge* of P_k. For the sake of convenience, we fix the embedding of $F_i(M)$ such that the outer face consists of b_k, c_k, d_k, $1 \le k \le 2m$, which are placed in a clockwise direction, as illustrated in Fig. 2(a).

It is easy to see that $F_i(M)$ has only two optimal orientations for ST_i: the one is to direct each odd petal in a clockwise direction and to direct each even petal in a counterclockwise direction; and the other is the reversed one. In the first optimal orientation, the external edges $\{c_k, d_k\}$ are directed from c_k to d_k in all odd petals P_k, while directed from d_k to c_k in all even petals; we call this optimal orientation of $F_i(M)$ a *true-orientation*, which corresponds to assigning TRUE to the variable u_i. On the other hand, the other optimal orientation of $F_i(M)$ is called a *false-orientation*, which corresponds to assigning FALSE to u_i. Clearly, $h^*(F_i(M)) = 2M + 1$.

Corresponding graph G_ϕ

We now construct the planar graph G_ϕ corresponding to the formula ϕ, as follows. We fix a plane embedding of the bipartite graph $B = (V', E')$ arbitrarily. For each variable u_i, $1 \le i \le n$, we replace it with the flower gadget $F_i(M)$. For each clause c_j, $1 \le j \le m$, we replace it with a path consisting of three vertices s_j, r_j, t_j; let $\omega(\{s_j, r_j\}) = \omega(\{r_j, t_j\}) = 2M$. We then connect flower gadgets $F_i(M)$, $1 \le i \le n$, with paths $s_j r_j t_j$, $1 \le j \le m$, as follows. For each clause c_j, $1 \le j \le m$, let l_{j1}, l_{j2}, l_{j3} be the three literals in c_j whose corresponding flower gadgets $F_{j1}(M), F_{j2}(M), F_{j3}(M)$ are placed in a clockwise order around the path $s_j r_j t_j$. Assume that l_{jk}, $1 \le k \le 3$, is either u_i or $\bar{u}_i$. Then, we replace the edge of B joining variable u_i and clause c_j with a pair of weight-1 edges which, together with an external edge in $F_i(M)$, forms a path between two vertices chosen from $\{s_j, r_j, t_j\}$, according to the following rules (see Fig. 2(b) as an example):

(i) The endpoints of this path are s_j and r_j if $k = 1$; r_j and t_j if $k = 2$; and s_j and t_j if $k = 3$.
(ii) The external edge is from an even petal if $l_{j1} = u_i$, $l_{j2} = u_i$, or $l_{j3} = \bar{u}_i$; while it is from an odd petal if $l_{j1} = \bar{u}_i$, $l_{j2} = \bar{u}_i$, or $l_{j3} = u_i$.
(iii) From the viewpoint of variable u_i, we choose a distinct external edge for each clause containing u_i, honoring the order of those clauses around u_i and thereby preserving the planarity of the embedding.

Finally, we replace each edge e in G_ϕ with a path of length $\omega(e)$ in which all edges are of weight 1. (Remember that M is a fixed constant.) Clearly, the resulting graph G_ϕ is a planar graph of maximum degree 4, and can be constructed in polynomial time. The set of all st-pairs in this instance is defined as follows:

$$\left(\bigcup_{i=1}^{n} ST_i\right) \cup \{(s_j, t_j) \mid 1 \leq j \leq m\}.$$

Therefore, there are $(12mn + m)$ st-pairs in total. This completes the construction of the corresponding instance of MIN-MAX ORIENTATION.

We now show that $h^*(G_\phi) \leq 2M + 3$ if and only if there exists a satisfying truth assignment for ϕ, and hence MIN-MAX ORIENTATION is strongly NP-hard for planar graphs of maximum degree 4 even if all edge-weights are identical.

Consider any satisfying truth assignment for ϕ. Then, according to the truth assignment, we assign either the true-orientation or the false-orientation to each flower gadget in G_ϕ. Since each clause c_j contains at least one TRUE-literal, G_ϕ has an orientation $\mathbf{G}_\phi$ such that there exists a directed (s_j, t_j)-path of distance at most $2M + 3$ via the external edge in the flower gadget corresponding to the TRUE-literal. Therefore, $h^*(G_\phi) \leq 2M + 3$ if there exists a satisfying truth assignment for ϕ.

Conversely, consider any orientation $\mathbf{G}_\phi$ of G_ϕ such that $h(\mathbf{G}_\phi) \leq 2M + 3$. Then, each flower gadget $F_i(M)$ must be directed as either the true-orientation or the false-orientation; otherwise $h(\mathbf{G}_\phi) > 2M + 3$. Moreover, since the distance of a shortest directed (s_j, t_j)-path in $\mathbf{G}_\phi$ is at most $2M + 3$ for each j, $1 \leq j \leq m$, it must pass through at least one external edge. This means that each clause c_j, $1 \leq j \leq m$, contains at least one TRUE-literal, and hence there exists a satisfying truth assignment for ϕ. □

From the proof of Theorem 1, we obtain the following corollary.

Corollary 1 *For any constant $\varepsilon > 0$,* MIN-MAX ORIENTATION *admits no polynomial-time $(2 - \varepsilon)$-approximation algorithm for planar graphs of maximum degree* 4 *unless* P = NP.

Proof Notice that, if there is no satisfying truth assignment for a given instance ϕ of PLANAR 3-SAT, then $h^*(G_\phi) \geq 4M$ for the corresponding instance G_ϕ of MIN-MAX ORIENTATION. Suppose for a contradiction that the problem admits a polynomial-time $(2 - \varepsilon)$-approximation algorithm for some constant $\varepsilon > 0$. Let $M = 3 \cdot \lceil \frac{1}{\varepsilon} \rceil$. Then, $(2 - \varepsilon)(2M + 3) < 4M$, and hence one can distinguish either $h^*(G) \leq 2M + 3$ or $h^*(G) \geq 4M$ in polynomial time using the algorithm. This is a contradiction unless P = NP. □

We then give the following theorem for MIN-SUM ORIENTATION.

Theorem 2 MIN-SUM ORIENTATION *is strongly NP-hard for planar graphs of maximum degree* 3 *even if all edge-weights are identical.*

Proof The proof is analogous to that for Theorem 1, but we give a reduction from the PLANAR MAX 2-SAT problem which is known to be strongly NP-complete [5].

In PLANAR MAX 2-SAT, we are given a Boolean formula ϕ in conjunctive normal form, say with set U of n variables $u_1, u_2, \ldots, u_n$ and set C of m clauses $c_1, c_2, \ldots, c_m$, such that each clause $c_j \in C$ contains exactly *two* literals and the bipartite graph $B = (U \cup C, E')$ is planar. The PLANAR MAX 2-SAT problem is to find a truth assignment for ϕ which satisfies at least ℓ clauses, for a given integer ℓ.

Given an instance of PLANAR MAX 2-SAT, we construct the corresponding instance of MIN-SUM ORIENTATION. We construct the same flower gadget $F_i(M)$ for each variable $u_i \in U$. Then, each flower gadget $F_i(M)$ has only two optimal orientations for ST_i, that is, the true-orientation and the false-orientation, and hence $g^*(F_i(M)) = 18m(M+1)$. On the other hand, we simply introduce an edge $\{s_j, t_j\}$ for each clause $c_j \in C$, instead of the path consisting of three vertices s_j, r_j, t_j; let $\omega(\{s_j, t_j\}) = 3M + 2$. We analogously connect the gadgets, but let the weight of the edge joining s_j, $1 \le j \le m$, and the endpoint of external edge be M. Note that the resulting graph G_ϕ corresponding to ϕ is a planar graph of maximum degree 3 since each clause contains two literals.

We now show that $g^*(G_\phi) \le 18mn(M+1) + (M+2)\ell + (3M+2)(m-\ell)$ if and only if there exists a truth assignment for ϕ which satisfies at least ℓ clauses.

Consider any truth assignment for ϕ which satisfies at least ℓ clauses. Then, according to the truth assignment, we assign either the true-orientation or the false-orientation to each flower gadget in G_ϕ. If a clause $c_j \in C$ is satisfied by the truth assignment, then c_j contains at least one TRUE-literal and hence we can direct edges so that there exists a directed (s_j, t_j)-path of distance $M+2$ via the external edge in the flower gadget corresponding to the TRUE-literal. On the other hand, if a clause $c_j \in C$ is *not* satisfied by the truth assignment, then c_j contains no TRUE-literal; we direct the edge $\{s_j, t_j\}$ from s_j to t_j, and hence there is a directed (s_j, t_j)-path of distance $\omega(\{s_j, t_j\}) = 3M+2$. Since at least ℓ clauses are satisfied by the truth assignment, we have $g^*(G_\phi) \le 18mn(M+1) + (M+2)\ell + (3M+2)(m-\ell)$.

Conversely, consider any orientation $\mathbf{G}_\phi$ of G_ϕ such that $g(\mathbf{G}_\phi) \le 18mn(M+1) + (M+2)\ell + (3M+2)(m-\ell)$. Suppose for a contradiction that any truth assignment for ϕ satisfies at most $\ell - 1$ clauses. Remember that each flower gadget $F_i(M)$ has only two optimal orientations for ST_i, and $g^*(F_i(M)) = 18m(M+1)$. Then, since at most $\ell - 1$ clauses can be satisfied, some of the flower gadgets must be directed in $\mathbf{G}_\phi$ as neither the true-orientation nor the false-orientation so as to increase the number of st-pairs (s_j, t_j) having directed (s_j, t_j)-paths of distance $M+2$ via external edges. However, reversing the direction of one external edge would detour some st-pair in ST_i of additional distance at least $2M+1$; moreover, the additional distance would be at least $2M+3$ if the detour goes through outside the flower gadget. Therefore, each flower gadget must be directed in $\mathbf{G}_\phi$ as either the true-orientation or the false-orientation, and hence $g(\mathbf{G}_\phi) > 18mn(M+1) + (M+2)\ell + (3M+2)(m-\ell)$, a contradiction. □

2.2 NP-hardness for cacti

We then show that MIN-MAX ORIENTATION remains NP-hard even for cacti with $q = 2$. A graph G is a *cactus* if every edge is part of at most one cycle in G [1, p. 169][13]. (See Figs. 3 and 4(a) as examples of cacti.) Cacti form a subclass of planar graphs. However, we have the following theorem.

Theorem 3 MIN-MAX ORIENTATION *is NP-hard for cacti of maximum degree* 4 *even if* $q = 2$.

Proof We show that the PARTITION problem, which is known to be NP-complete [4, p. 223], can be reduced in polynomial time to the MIN-MAX ORIENTATION problem for cacti with $q = 2$.

In PARTITION, we are given a finite set $A = \{a_1, a_2, \ldots, a_n\}$ in which each element $a_i \in A$ has a positive integer size $s(a_i)$. Then, the PARTITION problem is to decide whether there is a subset $A' \subset A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a) = \frac{1}{2} \sum_{a \in A} s(a)$.

From a given instance A of PARTITION, we construct a graph $G = (V, E)$ as the corresponding instance of MIN-MAX ORIENTATION, as follows. The vertex set V consists of $2n + 1$ vertices $v_0, v_1, \ldots, v_n, u_1, u_2 \ldots, u_n$. The edge set E consists of $3n$ edges $\{u_i, v_{i-1}\}$, $\{v_{i-1}, v_i\}$ and $\{v_i, u_i\}$, $1 \le i \le n$; each elementary cycle C_i, $1 \le i \le n$, consisting of the three edges $\{u_i, v_{i-1}\}$, $\{v_{i-1}, v_i\}$ and $\{v_i, u_i\}$ is called the *i-th cycle* of G. (See Fig. 3.) The weights of edges are defined as follows: $\omega(\{u_i, v_{i-1}\}) = \omega(\{v_{i-1}, v_i\}) = 1$ and $\omega(\{v_i, u_i\}) = s(a_i)$ for each i, $1 \le i \le n$. Clearly, G is a cactus. Let $(s_1, t_1) = (v_0, v_n)$ and $(s_2, t_2) = (v_n, v_0)$, and hence $q = 2$. This completes the construction of the corresponding instance of MIN-MAX ORIENTATION.

We now show that $h^*(G) = n + \frac{1}{2} \sum_{a \in A} s(a)$ if and only if there exists a desired subset A' for A. Since every orientation of G must have both a directed (v_0, v_n)-path P_1 and a directed (v_n, v_0)-path P_2, any orientation of G satisfies the following two properties: for each i, $1 \le i \le n$,

(i) if the edge $\{v_{i-1}, v_i\}$ is directed from v_{i-1} to v_i, then the edge $\{v_i, u_i\}$ is directed from v_i to u_i and the edge $\{u_i, v_{i-1}\}$ is directed from u_i to v_{i-1}; and

(ii) conversely, if $\{v_{i-1}, v_i\}$ is directed from v_i to v_{i-1}, then $\{u_i, v_{i-1}\}$ is directed from v_{i-1} to u_i and $\{v_i, u_i\}$ is directed from u_i to v_i.

Therefore, we clearly have $E(P_1) \cup E(P_2) = E$, where $E(P_1)$ and $E(P_2)$ are the sets of edges in P_1 and P_2, respectively. Since $q = 2$ and $\omega(P_1) + \omega(P_2) = \sum_{e \in E} \omega(e) = 2n + \sum_{a \in A} s(a)$, we have

$$h(\mathbf{G}) \ge n + \frac{1}{2} \sum_{a \in A} s(a) \tag{1}$$

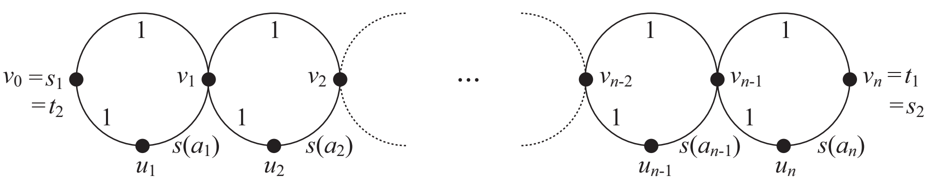

Fig. 3 Cactus with two st-pairs corresponding to instance A of PARTITION.

for any orientation $\mathbf{G}$ of G.

Suppose that G has an orientation $\mathbf{G}$ such that $h(\mathbf{G}) = n + \frac{1}{2}\sum_{a\in A} s(a)$. Note that by Eq. (1) we have $h^*(G) = h(\mathbf{G})$. Then, the following subset A' of A is clearly a desired subset for PARTITION:

$$A' = \{a_i \in A \mid \text{the } i\text{-th cycle } C_i \text{ of } G \text{ is directed as (i) above in } \mathbf{G}\}.$$

Conversely, suppose that there exists a desired subset A' of A. Then, $\sum_{a\in A'} s(a) = \sum_{a\in A\setminus A'} s(a) = \frac{1}{2}\sum_{a\in A} s(a)$. We define the corresponding orientation $\mathbf{G}$ of G, as follows: if $a_i \in A$ is in A', then the i-th cycle C_i of G is directed as (i) above; otherwise C_i is directed as (ii) above. Then

$$h(\mathbf{G}) = \omega(P_1) = \omega(P_2) = n + \frac{1}{2}\sum_{a\in A} s(a),$$

and hence by Eq. (1) this orientation $\mathbf{G}$ is optimal for the corresponding instance of MIN-MAX ORIENTATION. □

3 Polynomial-Time Algorithms

In this section, we first show that both MIN-SUM ORIENTATION and MIN-MAX ORIENTATION can be solved in polynomial time for cycles. We then show that MIN-SUM ORIENTATION is solvable in polynomial time for cacti by extending the algorithm for cycles.

3.1 MIN-SUM ORIENTATION and MIN-MAX ORIENTATION for cycles

The main result of this section is the following theorem.

Theorem 4 *Both* MIN-SUM ORIENTATION *and* MIN-MAX ORIENTATION *can be solved in time $O(n + q^2)$ for a cycle C, where n is the number of vertices in C.*

In the remainder of this subsection, we give a proof of Theorem 4. Suppose that we are given an edge-weighted cycle $C = (V, E)$ and q st-pairs (s_i, t_i), $1 \le i \le q$. Note that C has at least one orientation for any set of st-pairs: simply directing C in a clockwise direction. Therefore, $g^*(C) \ne +\infty$ and $h^*(C) \ne +\infty$.

For each st-pair (s_i, t_i), $1 \le i \le q$, let $\mathrm{cw}(i)$ be the set of all edges in the directed (s_i, t_i)-path when all edges in C are directed in a clockwise direction, and let $\mathrm{acw}(i)$ be the set of all edges in the directed (s_i, t_i)-path when all edges in C are directed in a counterclockwise (anticlockwise) direction. Clearly, for each i, $1 \le i \le q$, $\{\mathrm{cw}(i), \mathrm{acw}(i)\}$ is a partition of E, that is, $\mathrm{cw}(i) \cap \mathrm{acw}(i) = \emptyset$ and $\mathrm{cw}(i) \cup \mathrm{acw}(i) = E$. We introduce a $\{0,1\}$-variable x_i for each st-pair (s_i, t_i), $1 \le i \le q$: if $x_i = 0$, then the edges in $\mathrm{cw}(i)$ are directed in a clockwise direction; if $x_i = 1$, then the edges in $\mathrm{acw}(i)$ are directed in a counterclockwise direction. For two st-pairs (s_i, t_i) and (s_j, t_j), the two corresponding variables x_i and x_j have the following constraints (a)–(c):

(a) if $\mathrm{cw}(i) \cap \mathrm{acw}(j) \ne \emptyset$ and $\mathrm{acw}(i) \cap \mathrm{cw}(j) \ne \emptyset$, then $x_i = x_j$;
(b) if $\mathrm{cw}(i) \cap \mathrm{acw}(j) = \emptyset$ and $\mathrm{acw}(i) \cap \mathrm{cw}(j) \ne \emptyset$, then $x_i \le x_j$; and
(c) if $\mathrm{cw}(i) \cap \mathrm{acw}(j) \ne \emptyset$ and $\mathrm{acw}(i) \cap \mathrm{cw}(j) = \emptyset$, then $x_i \ge x_j$.

Since $\{\mathrm{cw}(k), \mathrm{acw}(k)\}$ is a partition of E for each k, $1 \le k \le q$, it is easy to see that no pair of st-pairs (s_i, t_i) and (s_j, t_j), $1 \le i, j \le q$, with $i \ne j$, satisfies $\mathrm{cw}(i) \cap \mathrm{acw}(j) = \emptyset$ and $\mathrm{acw}(i) \cap \mathrm{cw}(j) = \emptyset$, and hence any two variables x_i and x_j have exactly one of the constraints (a)–(c) above.

We now construct a *constraint graph* $\mathcal{C}$ in which each vertex v_i corresponds to an st-pair (s_i, t_i) and there is an edge between two vertices v_i and v_j if and only if the corresponding variables x_i and x_j have the constraint $x_i = x_j$, that is, $\mathrm{cw}(i) \cap \mathrm{acw}(j) \ne \emptyset$ and $\mathrm{acw}(i) \cap \mathrm{cw}(j) \ne \emptyset$. From an orientation of C, we can obtain an assignment of $\{0, 1\}$ to each variable x_k, $1 \le k \le q$; clearly, any two variables satisfy their constraint, and hence two variables x_i and x_j receive the same value if their corresponding vertices v_i and v_j are contained in the same connected component of $\mathcal{C}$.

Let $\mathcal{V} = \{V_1, V_2, \dots, V_m\}$ be the partition of the vertex set of $\mathcal{C}$ such that each V_i, $1 \le i \le m$, forms a connected component of $\mathcal{C}$. Then, we define a relation "$\le$" on $\mathcal{V}$, as follows: $V_i \le V_j$ if and only if there exist two vertices $v_i \in V_i$ and $v_j \in V_j$ such that their corresponding variables x_i and x_j have the constraint $x_i \le x_j$. We show that $\mathcal{V}$ is totally ordered under the relation $\le$, as in the following lemma.

Lemma 1 *$\mathcal{V}$ is totally ordered under the relation $\le$.*

Proof Consider any two subsets V_i and V_j in $\mathcal{V}$ such that $V_i \ne V_j$. We will show that exactly one of $V_i \le V_j$ and $V_i \ge V_j$ holds. It suffices to show that, for any two vertices v_{i_1} and v_{i_2} in V_i and a vertex v_j in V_j, their corresponding variables x_{i_1}, x_{i_2} and x_j have exactly one of the following constraints (i) and (ii): (i) $x_{i_1} \le x_j$ and $x_{i_2} \le x_j$; and (ii) $x_{i_1} \ge x_j$ and $x_{i_2} \ge x_j$.

Suppose for a contradiction that the variables have the constraints $x_{i_1} \le x_j$ and $x_{i_2} \ge x_j$; it is similar for the case $x_{i_1} \ge x_j$ and $x_{i_2} \le x_j$. Since $v_{i_1}, v_{i_2} \in V_i$, there is a path between v_{i_1} and v_{i_2} via only vertices in V_i. Then, since $x_{i_1} \le x_j$ and $x_{i_2} \ge x_j$, the path contains at least one edge joining v_{i_k} and $v_{i_{k'}}$ whose corresponding variables satisfy the two constraints $x_{i_k} \le x_j$ and $x_{i_{k'}} \ge x_j$. Since v_{i_k} and $v_{i_{k'}}$ are adjacent in $\mathcal{C}$, the constraint $x_{i_k} = x_{i_{k'}}$ holds. Therefore, we have $\mathrm{cw}(i_k) \cap \mathrm{acw}(i_{k'}) \ne \emptyset$ and $\mathrm{acw}(i_k) \cap \mathrm{cw}(i_{k'}) \ne \emptyset$. Let

$$e \in \mathrm{cw}(i_k) \cap \mathrm{acw}(i_{k'}). \tag{2}$$

Since the constraint $x_{i_k} \le x_j$ holds, we have

$$\mathrm{cw}(i_k) \cap \mathrm{acw}(j) = \emptyset \tag{3}$$

and $\mathrm{acw}(i_k) \cap \mathrm{cw}(j) \ne \emptyset$. Similarly, since the constraint $x_{i_{k'}} \ge x_j$ holds, we have

$$\mathrm{cw}(i_{k'}) \cap \mathrm{acw}(j) \ne \emptyset \tag{4}$$

and $\mathrm{acw}(i_{k'}) \cap \mathrm{cw}(j) = \emptyset$. Then by Eqs. (2) and (3) we have $e \notin \mathrm{acw}(j)$. Since $\{\mathrm{cw}(j), \mathrm{acw}(j)\}$ is a partition of E, we thus have $e \in \mathrm{cw}(j)$. Then, by Eq. (2) we have $e \in \mathrm{acw}(i_{k'}) \cap \mathrm{cw}(j) \ne \emptyset$. Together with Eq. (4), there is the constraint $x_{i_{k'}} = x_j$. Therefore, $\mathcal{C}$ has an edge between $v_{i_{k'}}$ and v_j, and hence $v_j \in V_i$. This contradicts the fact that $V_i \ne V_j$. □

Lemma 1 implies that, for some index k, $1 \le k \le m$, we have $x_i = 0$ for all variables x_i whose corresponding vertices are contained in V_j with $V_j \le V_k$; otherwise $x_i = 1$. Therefore, both MIN-SUM ORIENTATION and MIN-MAX ORIENTATION can be reduced simply to finding such an appropriate index k on $\mathcal{V} = \{V_1, V_2, \dots, V_m\}$. Then, both problems can be solved in time $O(n+q^2)$, as follows. We first label the vertices in a clockwise order starting from any vertex, say s_1. We can now easily determine, from the labels of vertices, which of the constraints (a)–(c) above holds in time $O(1)$ for each pair of st-pairs, and hence the constraint graph $\mathcal{C}$ can be constructed in time $O(n+q^2)$. As a preprocessing, we compute each of the total edge-weights of $\mathrm{cw}(i)$ and $\mathrm{acw}(i)$; this can be done in time $O(n+q)$ for all i, $1 \le i \le q$. Then, the appropriate index k on $\mathcal{V}$ can be found in time $O(q^2)$. Therefore, both problems can be solved in time $O(n+q^2)$ in total.

3.2 Min-sum orientation for cacti

By extending Theorem 4, min-sum orientation can be solved in polynomial time also for cacti, as in the following theorem.

Theorem 5 Min-sum orientation *can be solved in time $O(nq^2)$ for a cactus G, where n is the number of vertices in G.*

Proof It can be easily determined in time $O(nq)$ whether a given cactus $G=(V,E)$ has at least one (feasible) orientation for the given set of st-pairs; we simply check the st-pairs that pass through bridges in G; if there exists a pair of st-pairs that pass through the same bridge in different directions, then G has no orientation. Therefore, we assume without loss of generality that G has an orientation, and hence $g^*(G) \neq +\infty$.

Let B be the set of all bridges in G. Then, $E \setminus B$ induces the set of all elementary cycles in G; let C be the set of all elementary cycles in G. For each bridge $e \in B$, we denote by $b(e)$ the number of st-pairs that pass through the bridge e; the values $b(e)$ for all bridges $e \in B$ can be computed in time $O(nq)$. Consider any orientation $\mathbf{G}$ of G. Then, each directed (s_i,t_i)-path, $1 \le i \le q$, can be decomposed into bridges and subpaths in elementary cycles of G. We thus have

$$g(\mathbf{G}) = \sum_{e\in B} b(e)\cdot\omega(e) + \sum_{c\in C}\sum_{i=1}^{q}\omega(\mathbf{G},c,i), \tag{5}$$

where $\omega(\mathbf{G},c,i)$ is the sum of the weights of all edges that are contained in both a cycle $c \in C$ and the shortest directed (s_i,t_i)-path in $\mathbf{G}$. Equation (5) implies that computing $g^*(G)$ for a cactus G can be reduced to solving min-sum orientation for each cycle $c \in C$ independently. Using Theorem 4, min-sum orientation for a cycle c can be solved in time $O(|c|+q^2)$, where $|c|$ denotes the number of vertices in c. Therefore, min-sum orientation for a cactus G can be solved in time $O\left(nq + \sum_{c\in C}(|c|+q^2)\right) = O(nq^2)$. □

4 FPTAS for Min-Max Orientation on Cacti

In contrast to min-sum orientation, as we have shown in Theorem 3, min-max orientation remains NP-hard even for cacti with $q=2$. However, in this section, we give an FPTAS for min-max orientation on cacti if q is a fixed constant.

In Section 4.1 we first present a polynomial-time 2-approximation algorithm based on LP-relaxation, which gives us both lower and upper bounds on $h^*(G)$ for a given

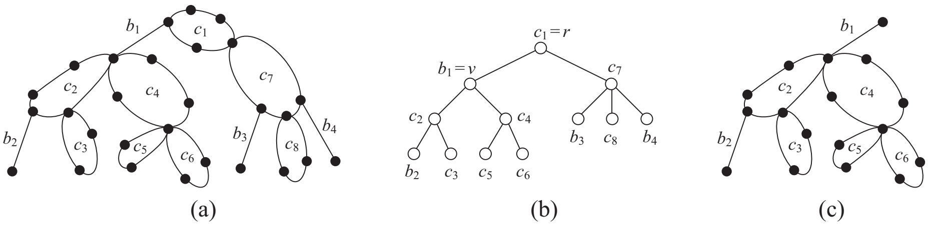

Fig. 4 (a) A cactus G, (b) an underlay tree T of G, and (c) the subgraph G_v of G.

cactus G. We then show in Section 4.2 that the problem can be solved in pseudo-polynomial time for cacti. In Section 4.3, we finally give our FPTAS based on the algorithm in Section 4.2 and using the lower and upper bounds on $h^*(G)$ obtained in Section 4.1. As in the proof of Theorem 5, we may assume without loss of generality that G has at least one orientation, and hence $h^*(G) \neq +\infty$.

[Cactus and its underlay tree]

A cactus G can be represented by an *underlay tree* T, which is a rooted tree and can be easily obtained from G in a straightforward way. (See Fig. 4(a) and (b) as an example). In the underlay tree T of G, each node represents either a bridge of G or an elementary cycle of G; and if there is an edge between nodes u and v of T, then bridges or cycles of G represented by u and v share exactly one vertex in G. (A similar idea can be found in [13, Theorem 11].) Each node v of T corresponds to a subgraph G_v of G induced by all bridges and cycles represented by the nodes that are descendants of v in T. Figure 4(c) depicts the subgraph G_v for the left child v of the root r of T in Fig. 4(b). Clearly, G_v is a cactus for each node v of T, and $G = G_r$ for the root r of T. It is easy to see that an underlay tree T of a given cactus G can be found in linear time, and hence we may assume that a cactus G and its underlay tree T are both given. In Section 4.2, we solve MIN-MAX ORIENTATION by a dynamic programming approach based on the underlay tree T of G.

4.1 2-approximation algorithm based on LP-relaxation

In this subsection, we give the following theorem. It should be noted that the number q of st-pairs is not required to be a fixed constant in the theorem.

Theorem 6 *There is a polynomial-time* 2*-approximation algorithm for* MIN-MAX ORIENTATION *on cacti.*

For each st-pair (s_i, t_i), $1 \leq i \leq q$, let C_i be the set of elementary cycles represented by the nodes which are on the path between nodes v_{s_i} and v_{t_i} in the underlay tree T of a given cactus G, where v_{s_i} and v_{t_i} are the nodes in T containing s_i and t_i, respectively. Let d_i be the sum of weights of all bridges represented by the nodes which are on the path from v_{s_i} to v_{t_i} in T. Clearly, both C_i and d_i can be computed in time $O(nq)$ for all st-pairs (s_i, t_i), $1 \leq i \leq q$.

Consider the following two orientations of G: the one, denoted by $\mathbf{G}^a$, directs all elementary cycles in G in a clockwise direction; the other, denoted by $\mathbf{G}^b$, directs all elementary cycles in G in a counterclockwise direction. Clearly, both $\mathbf{G}^a$ and $\mathbf{G}^b$ are (feasible) orientations of G. For each elementary cycle c in G, we call an ordered index-pair (i,j), $1 \le i,j \le q$, a *conflicting pair on* c if the directed (s_i,t_i)-path in $\mathbf{G}^a$ and the directed (s_j,t_j)-path in $\mathbf{G}^b$ share at least one edge of c. Then, for a conflicting pair (i,j) on c, any orientation $\mathbf{G}$ of G satisfies the followings:

(i) if $\mathbf{G}$ has a directed (s_i,t_i)-path which passes through c in a clockwise direction, then any directed (s_j,t_j)-path in $\mathbf{G}$ passes through c in a clockwise direction, too; and

(ii) if $\mathbf{G}$ has a directed (s_j,t_j)-path which passes through c in a counterclockwise direction, then any directed (s_i,t_i)-path in $\mathbf{G}$ passes through c in a counterclockwise direction, too.

For an st-pair (s_i,t_i), $1 \le i \le q$, and each elementary cycle $c \in C_i$, we denote by a_i^c and b_i^c the sums of weights of the edges which are contained in both c and the directed (s_i,t_i)-paths in $\mathbf{G}^a$ and $\mathbf{G}^b$, respectively.

For an st-pair (s_i,t_i), $1 \le i \le q$, and each elementary cycle $c \in C_i$, we introduce two kinds of $\{0,1\}$-variables x_i^c and y_i^c: if $x_i^c = 1$, then we direct edges of c so that there is a directed (s_i,t_i)-path which passes through c in a clockwise direction; if $y_i^c = 1$, then we direct edges of c so that there is a directed (s_i,t_i)-path which passes through c in a counterclockwise direction.

We are now ready to formulate MIN-MAX ORIENTATION for a cactus G.

$$\text{minimize} \quad z \tag{6}$$

$$\text{subject to} \quad x_i^c + y_i^c = 1 \quad \text{for all } c \in C_i,\ i = 1,\ldots,q, \tag{7}$$

$$x_i^c + y_j^c \le 1 \quad \text{for all conflicting pairs } (i,j) \text{ on each cycle } c \text{ in } G, \tag{8}$$

$$d_i + \sum_{c \in C_i} (a_i^c x_i^c + b_i^c y_i^c) \le z \quad \text{for each } i = 1,\ldots,q, \tag{9}$$

$$x_i^c,\ y_i^c \in \{0,1\} \quad \text{for all } c \in C_i,\ i = 1,\ldots,q. \tag{10}$$

Equations (7) and (8) ensure that there are directed (s_i,t_i)-paths for all st-pairs (s_i,t_i), $1 \le i \le q$. Therefore, according to the values of x_i^c and y_i^c, we can find an orientation $\mathbf{G}$ of G such that

$$h(\mathbf{G}) = \max\left\{ d_i + \sum_{c \in C_i} (a_i^c x_i^c + b_i^c y_i^c) \mid 1 \le i \le q \right\} = z.$$

Thus, minimizing z in Eq. (6) is equivalent to computing $h^*(G)$ for G. Since the size of the above integer programming formulation is polynomial in n, its linear relaxation problem can be solved in polynomial time.

[2-approximation algorithm]

We now propose a polynomial-time 2-approximation algorithm for cacti. We first solve the linear relaxation problem, and obtain a fractional solution $\bar{x}_i^c$ and $\bar{y}_i^c$, whose objective value is $\bar{z}$. Clearly, $h^*(G) \ge \bar{z}$, because $h^*(G)$ is the optimal value for the IP above. We then obtain an integer solution x_i^c and y_i^c by rounding the values of $\bar{x}_i^c$ and $\bar{y}_i^c$, as follows:

$$x_i^c = \begin{cases} 1 & \text{if } \bar{x}_i^c \ge 0.5; \\ 0 & \text{if } \bar{x}_i^c < 0.5, \end{cases}$$

and

$$y_i^c = \begin{cases} 1 & \text{if } \bar{y}_i^c > 0.5; \\ 0 & \text{if } \bar{y}_i^c \le 0.5. \end{cases}$$

Clearly, x_i^c and y_i^c satisfy Eqs. (7), (8) and (10), and hence x_i^c and y_i^c form a feasible solution for the IP above; we can thus obtain an orientation of G. Moreover, this algorithm clearly takes polynomial time. Therefore, it suffices to show that the approximation ratio of this algorithm is 2. Let z_A be the objective value for the solution x_i^c and y_i^c. Since $\bar{x}_i^c \geq \frac{1}{2}x_i^c$ and $\bar{y}_i^c \geq \frac{1}{2}y_i^c$, by Eq. (9) we have

$$\begin{aligned} h^*(G) &\geq \bar{z} \\ &= \max\left\{d_i + \sum_{c \in C_i}(a_i^c \bar{x}_i^c + b_i^c \bar{y}_i^c) \mid 1 \leq i \leq q\right\} \\ &\geq \frac{1}{2}\max\left\{d_i + \sum_{c \in C_i}(a_i^c x_i^c + b_i^c y_i^c) \mid 1 \leq i \leq q\right\} \\ &= \frac{1}{2}z_A. \end{aligned} \tag{11}$$

This completes the proof of Theorem 6. □

4.2 Pseudo-polynomial-time algorithm

The main result of this subsection is the following theorem.

Theorem 7 MIN-MAX ORIENTATION *can be solved in time* $O(q2^qU^{2q}n)$ *for a cactus* G*, where* U *is an arbitrary upper bound on* $h^*(G)$ *and* n *is the number of vertices in* G.

As the upper bound U on $h^*(G)$, we will employ the approximation value z_A obtained by the 2-approximation algorithm in Section 4.1; z_A can be computed in polynomial time.

[Main idea]

Let $G = (V, E)$ be a given cactus, let v be a node of an underlay tree T of G, and let G_v be the subgraph of G for the node v. Then, G_v and $G \setminus G_v$ share exactly one vertex u; in other words, u is the cut-vertex which separates G into $G_v \setminus \{u\}$ and $G \setminus G_v$. Consider an optimal orientation $\mathbf{G}$ of G. (Remember that G has at least one orientation for the given set of st-pairs.) Then, $\mathbf{G}$ naturally induces the "edge-direction" $\mathbf{G}_v$ of G_v, which is not always an orientation for the given set of st-pairs but satisfies the following four conditions: for each st-pair (s_i, t_i), $1 \leq i \leq q$,

(a) if both s_i and t_i are in G_v, then a shortest directed (s_i, t_i)-path in $\mathbf{G}$ is contained in $\mathbf{G}_v$ (remember that all edge-weights are non-negative);
(b) if s_i is in G_v but t_i is in $G \setminus G_v$, then there is a directed (s_i, u)-path in $\mathbf{G}_v$;
(c) conversely, if s_i is in $G \setminus G_v$ but t_i is in G_v, then there is a directed (u, t_i)-path in $\mathbf{G}_v$; and
(d) if neither s_i nor t_i are in G_v, then $\mathbf{G}$ has a shortest directed (s_i, t_i)-path which contains no edge of G_v.

For a q-tuple $(x_1, x_2, \ldots, x_q)$ of integers $0 \leq x_i \leq U$, $1 \leq i \leq q$, an edge-direction $\mathbf{G}_v$ of G_v is called an $(x_1, x_2, \ldots, x_q)$-*orientation* of G_v if the following three conditions (a)–(c) are satisfied: for each st-pair (s_i, t_i), $1 \leq i \leq q$,

(a) if both s_i and t_i are in G_v, then $\omega(\mathbf{G}_v, s_i, t_i) = x_i$;
(b) if s_i is in G_v but t_i is in $G \setminus G_v$, then $\omega(\mathbf{G}_v, s_i, u) = x_i$; and

(c) if s_i is in $G \setminus G_v$ but t_i is in G_v, then $\omega(\mathbf{G}_v, u, t_i) = x_i$.

Remember that $\omega(\mathbf{G}_v, x, y)$ denotes the total weight of a shortest directed (x, y)-path in $\mathbf{G}_v$ for two vertices x and y in G_v. We then define a set $F(G_v)$ of q-tuples, as follows:

$$F(G_v) = \{(x_1, x_2, \ldots, x_q) \mid G_v \text{ has an } (x_1, x_2, \ldots, x_q)\text{-orientation}\}.$$

Our algorithm computes $F(G_v)$ for each node v of T from the leaves to the root r of T by means of dynamic programming. Since $G = G_r$, we clearly have

$$h^*(G) = \min\left\{ \max_{1 \le i \le q} x_i \mid (x_1, x_2, \ldots, x_q) \in F(G_r) \right\}. \tag{12}$$

Note that $F(G_r) \neq \emptyset$ since we have assumed that G has at least one orientation for the given set of st-pairs. Therefore, we can always compute $h^*(G)$ by Eq. (12).

[Definitions]

Let v be a node of the underlay tree T for a cactus G, and let G_v be the subgraph of G for the node v. We simply call either a bridge or an elementary cycle of G represented by v the *component* of v. We say that an st-pair (s_i, t_i) *passes through* the component c of v if the node v is on the path between nodes v_{s_i} and v_{t_i} in T, where v_{s_i} and v_{t_i} are the nodes in T whose components contain s_i and t_i, respectively. Note that (s_i, t_i) passes through the components represented by v_{s_i} and v_{t_i} themselves. For each component c of v and each st-pair (s_i, t_i) passing through c, we can easily define the "dummy" st-pair (s_i^c, t_i^c), as follows: if s_i (or t_i) is in c, then $s_i^c = s_i$ (respectively, $t_i^c = t_i$); if s_i (or t_i) is not in c, then s_i^c (respectively, t_i^c) is the cut-vertex in c which separates c from s_i (respectively, t_i).

We have defined an $(x_1, x_2, \ldots, x_q)$-orientation of a subgraph G_v in order to know the distances of directed (s_i, t_i)-*sub*paths, $1 \le i \le q$, in G_v. Our dynamic programming algorithm needs more information when updating DP tables: we wish to fix the orientation of the component of v. For an elementary cycle c of G and a q-tuple $(j_1, j_2, \ldots, j_q)$ with $j_i \in \{0, 1\}$, $1 \le i \le q$, we define a $(j_1, j_2, \ldots, j_q)$-*orientation* $\mathbf{c}$ of c, as follows:

- if $j_i = 0$ and the st-pair (s_i, t_i) passes through c, then $\mathbf{c}$ must contain a directed (s_i^c, t_i^c)-path which is directed in a clockwise direction;
- if $j_i = 1$ and (s_i, t_i) passes through c, then $\mathbf{c}$ must contain a directed (s_i^c, t_i^c)-path which is directed in a counterclockwise direction.

Note that we do not care the st-pairs which do not pass through c. Clearly, we can determine in time $O(|c|q)$ whether c has a $(j_1, j_2, \ldots, j_q)$-orientation for a given q-tuple $(j_1, j_2, \ldots, j_q)$, where $|c|$ is the number of vertices in c. For the sake of convenience, we extend the notation of $(j_1, j_2, \ldots, j_q)$-orientations to a bridge $\{u, w\}$ of G: if $j_i = 0$ for all i, $1 \le i \le q$, then $\{u, w\}$ is directed from u to w; if $j_i = 1$ for all i, $1 \le i \le q$, then $\{u, w\}$ is directed from w to u; for the other q-tuples $(j_1, j_2, \ldots, j_q)$, the bridge $\{u, w\}$ has no feasible $(j_1, j_2, \ldots, j_q)$-orientation.

For a q-tuple $(j_1, j_2, \ldots, j_q)$, let k be the integer whose binary representation is $j_1 j_2 \ldots j_q$; and hence $0 \le k < 2^q$. For the graph G_v corresponding to a node v of T, we define a set F^k of q-tuples $(x_1, x_2, \ldots, x_q)$, as follows:

$$\begin{aligned} F^k(G_v) = \{(x_1, x_2, \ldots, x_q) \mid\ & G_v \text{ has an } (x_1, x_2, \ldots, x_q)\text{-orientation such that} \\ & \text{the component } c \text{ of } v \text{ is directed as the } (j_1, j_2, \ldots, j_q)\text{-orientation}\}. \end{aligned}$$

Clearly, we have

$$F(G_v) = \bigcup_{k=0}^{2^q-1} F^k(G_v). \tag{13}$$

Therefore, computing $F(G_v)$ is equivalent to computing $F^k(G_v)$ for all k, $0 \le k < 2^q$.

We now explain how to compute $F(G_v)$ for each node v of the underlay tree T of a cactus G. Let $v_1, v_2, \ldots, v_p$ be the children of v in T ordered arbitrarily. For each index l, $1 \le l \le p$, we denote by G_v^l the graph obtained by the union of the subgraphs $c, G_{v_1}, G_{v_2}, \ldots, G_{v_l}$, where c is the component of v. (See Fig. 5 in which the graph G_v^{l-1} is indicated by a thick dotted line.) Then, $G_v^p = G_v$. For the sake of convenience, the component c of v is sometimes denoted by G_v^0.

[Initialization]

We first compute $F^k(G_v^0)$ for each index k, $0 \le k < 2^q$. Since G_v^0 consists of a single component c of the node v, G_v^0 is either a single edge or a cycle. For the q-tuple $(j_1, j_2, \ldots, j_q)$ corresponding to k, if c has no $(j_1, j_2, \ldots, j_q)$-orientation, then let

$$F^k(G_v^0) = \emptyset; \tag{14}$$

and if c has a $(j_1, j_2, \ldots, j_q)$-orientation $\mathbf{c}$, then let $F^k(G_v^0) = \{(x_1, x_2, \ldots, x_q)\}$ where

$$x_i = \begin{cases} \omega(\mathbf{c}, s_i^c, t_i^c) & \text{if the } st\text{-pair } (s_i, t_i) \text{ passes through } c; \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

for each i, $1 \le i \le q$. By Eq. (13) we can thus compute the set $F(G_v^0)$ for each node v of T.

[Merge Operation]

We then compute $F^k(G_v)$ for each index k, $0 \le k < 2^q$. It should be noted that, since $G_v = G_v^0$ if v is a leaf of T, we have already computed the sets $F(G_v)$ for all leaves v of T. We may thus assume that v is an internal node of T, and that the sets $F(G_{v_l})$ have been computed for all children v_l, $1 \le l \le p$, of v in T.

Let c be the component of the node v. For the q-tuple $(j_1, j_2, \ldots, j_q)$ corresponding to the index k, if c has no $(j_1, j_2, \ldots, j_q)$-orientation, then let

$$F^k(G_v) = \emptyset.$$

Assume now that c has a $(j_1, j_2, \ldots, j_q)$-orientation $\mathbf{c}$. For each graph G_v^l, $1 \le l \le p$, we recursively compute the set $F^k(G_v^l)$ from the two sets $F^k(G_v^{l-1})$ and $F(G_{v_l})$; since $G_v^p = G_v$, we then have the set $F^k(G_v)$. Remember that by Eq. (15) we have already computed the set $F^k(G_v^0)$. From a pair of q-tuples $(y_1, y_2, \ldots, y_q) \in F^k(G_v^{l-1})$ and $(z_1, z_2, \ldots, z_q) \in F(G_{v_l})$, a q-tuple $(x_1, x_2, \ldots, x_q)$ in $F^k(G_v^l)$ can be obtained, as follows:

(i) $x_i = z_i$ for all st-pairs (s_i, t_i), $1 \le i \le q$, such that both s_i and t_i are contained in G_{v_l}, as illustrated in Fig. 5(i);

(ii) $x_i = y_i + z_i + \omega(\mathbf{c}, s_i^c, t_i^c)$ for all st-pairs (s_i, t_i), $1 \le i \le q$, such that either s_i or t_i is contained in G_{v_l} and the other is contained in G_v^{l-1} (in Fig. 5(ii), t_i is contained in G_{v_l} and s_i is contained in G_v^{l-1});

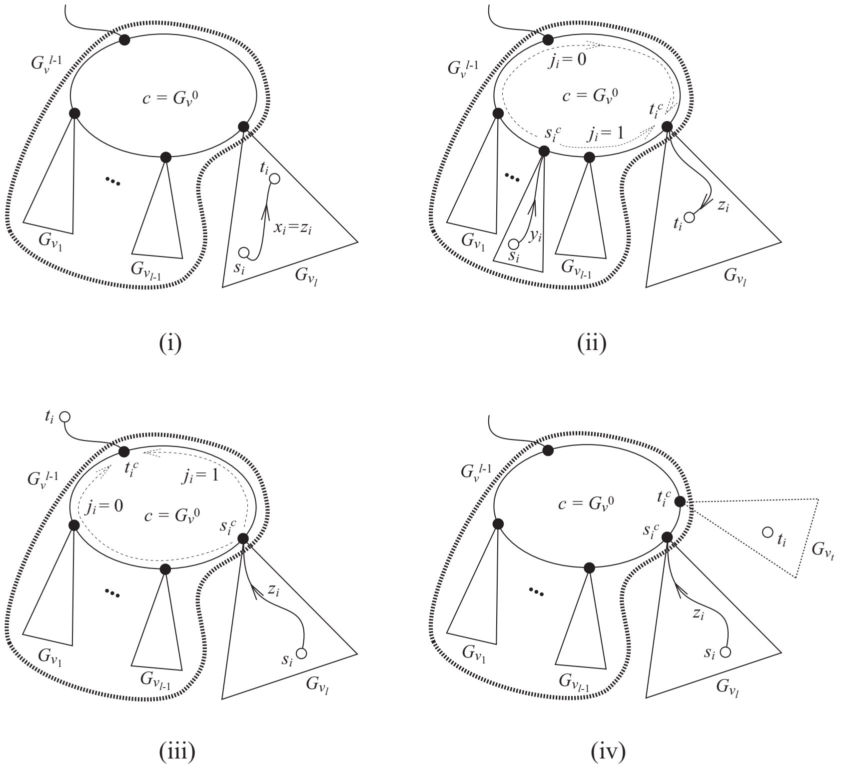

Fig. 5 The merge operation (i)–(iv).

(iii) $x_i = z_i + \omega(\mathbf{c}, s_i^c, t_i^c)$ for all st-pairs (s_i, t_i), $1 \le i \le q$, such that either s_i or t_i is contained in G_{v_l} and the other is contained in $G \setminus G_v$ (in Fig. 5(iii), s_i is contained in G_{v_l} and t_i is contained in $G \setminus G_v$);

(iv) $x_i = z_i$ for all st-pairs (s_i, t_i), $1 \le i \le q$, such that either s_i or t_i is contained in G_{v_l} and the other is contained in $G_v \setminus G_v^l$ (in Fig. 5(iv), s_i is contained in G_{v_l} and t_i is contained in G_{v_t} for some index t, $l < t \le p$); and

(v) $x_i = y_i$ for all the other elements x_i which are not defined yet by (i)–(iv) above.

If the q-tuple $(x_1, x_2, \ldots, x_q)$ obtained by (i)–(v) above contains an element x_i, $1 \le i \le q$, with $x_i > U$, then we delete the q-tuple from $F^k(G_v^l)$. It is obvious that the set $F^k(G_v^l)$ can be computed from all pairs of q-tuples $(y_1, y_2, \ldots, y_q) \in F^k(G_v^{l-1})$ and $(z_1, z_2, \ldots, z_q) \in F(G_{v_l})$ by (i)–(v) above.

[Proof of Theorem 7]

We finally show that our algorithm takes time $O(q2^qU^{2q}n)$.

The initialization can be done in time $O(q2^qn)$ for all nodes v of T and all indices k, $0 \le k < 2^q$, as follows:

(a) As a preprocessing, for the component c of each node v of T, we first determine (s_i^c, t_i^c), $1 \le i \le q$. This can be done in time $O(nq)$ for all i, $1 \le i \le q$, and all components c of v in T. We then compute the distances of directed (s_i^c, t_i^c)-paths for $j_i = 0, 1$ and for all st-pairs (s_i^c, t_i^c), $1 \leq i \leq q$. This can be done in time $O(|c|q)$ for each node v, and hence in time $O(nq)$ for all nodes v of T.

(b) Given a q-tuple $(j_1, j_2, \ldots, j_q)$, it can be determined in time $O(|c|q)$ whether the component c has a $(j_1, j_2, \ldots, j_q)$-orientation. If c does not have one, then by Eq. (14) we can compute $F^k(G_v^0)$ in time $O(1)$ for the index k. On the other hand, if c has a $(j_1, j_2, \ldots, j_q)$-orientation, then by Eq. (15) and using the preprocessing (a) above, we can compute $F^k(G_v^0)$ in time $O(q)$ for the index k. Therefore, $F^k(G_v^0)$ can be computed in time $O(|c|q)$ for an index k and a node v of T. Since k is taken over all $0 \leq k < 2^q$ and $|c|$ is taken over all nodes v of T, we can compute $F^k(G_v^0)$ in total time

$$\sum_{k=0}^{2^q-1} \sum_{v \in T} O(|c|q) = O(q2^q n).$$

We then estimate the running time of the merge operation. For a node v of T and an index k, $0 \leq k < 2^q$, clearly $|F^k(G_v)| \leq (U+1)^q = O(U^q)$. From a pair of q-tuples $(y_1, y_2, \ldots, y_q) \in F^k(G_v^{l-1})$ and $(z_1, z_2, \ldots, z_q) \in F(G_{v_l})$, we can compute a q-tuple $(x_1, x_2, \ldots, x_q)$ in $F^k(G_v^l)$ in time $O(q)$ by (i)–(v) above. Since $|F^k(G_v^{l-1})| = O(U^q)$ and $|F(G_{v_l})| = O(U^q)$, there are $O(U^{2q})$ pairs and hence we can compute the set $F^k(G_v^l)$ in time $O(qU^{2q})$. Therefore, $F^k(G_v) = F^k(G_v^p)$ can be computed in time $O(qU^{2q}p)$ for each index k, $0 \leq k < 2^q$. By Eq. (13) we can compute the set $F(G_v)$ in time $O(q2^qU^{2q}p)$ for a node v of T. Since p is the number of children of v, we can thus compute the set $F(G_r)$ for the root r of T in total time

$$\sum_{v \in T} O(q2^qU^{2q}p) = O(q2^qU^{2q}n).$$

Then, by Eq. (12) we can compute $h^*(G)$ in time $O(qU^q)$ from $F(G_r)$.

In this way, our algorithm solves MIN-MAX ORIENTATION for a cactus in time $O(q2^qU^{2q}n)$. □

4.3 FPTAS

From now on, we assume that the number q of st-pairs is a fixed constant. We finally give the main result of this section, as in the following theorem.

Theorem 8 MIN-MAX ORIENTATION *admits a fully polynomial-time approximation scheme for cacti if q is a fixed constant.*

As a proof of Theorem 8, we give an algorithm to find an orientation $\mathbf{G}$ of a cactus G with $h(\mathbf{G}) < (1+\varepsilon)h^*(G)$ in time polynomial in both n and $1/\varepsilon$ for any real number $\varepsilon > 0$, where n is the number of vertices in G. Thus, our approximation value $h_A(G)$ for G is $h(\mathbf{G})$, and hence the error is bounded by $\varepsilon h^*(G)$, that is,

$$h_A(G) - h^*(G) = h(\mathbf{G}) - h^*(G) < \varepsilon h^*(G). \tag{16}$$

We now give our algorithm. We extend the ordinary "scaling and rounding" technique [14, Chap. 8], and apply it to MIN-MAX ORIENTATION for a cactus $G = (V, E)$. For some scaling factor $\tau > 0$ (which will be defined later), let G_τ be the graph with the same vertex set V and edge set E as G, but the weight $\bar{\omega}(e)$ of each edge $e \in E$ is defined as follows: $\bar{\omega}(e) = \lceil \omega(e)/\tau \rceil$. Then, since both instances have the same set of st-pairs, any orientation of G_τ is an orientation of G. We optimally solve MIN-MAX ORIENTATION for G_τ by using the pseudo-polynomial-time algorithm in Section 4.2. We take the optimal orientation $\mathbf{G}_\tau$ for G_τ as our approximation solution for G.

We remark in passing that our polynomial-time 2-approximation algorithm in Section 4.1 will be employed to bound both the error and the running time of our FPTAS. Indeed, this constant-factor approximation helps us to obtain a faster FPTAS, compared with employing a non-constant, say $O(n)$, factor approximation.

[Error]

We first show that our approximation value $h_A(G)$ satisfies Eq. (16). Let $\mathbf{G}^*$ be any optimal orientation of G. For each st-pair (s_i, t_i), $1 \le i \le q$, we denote by O_i the set of edges in a shortest directed (s_i, t_i)-path in $\mathbf{G}^*$. Then, we have

$$h^*(G) = \max_{1\le i\le q} \omega(\mathbf{G}^*, s_i, t_i) = \max_{1\le i\le q} \sum_{e\in O_i} \omega(e). \tag{17}$$

Similarly, for each st-pair (s_i, t_i), $1 \le i \le q$, we denote by A_i the set of edges in a shortest directed (s_i, t_i)-path in $\mathbf{G}_\tau$. Since we take the orientation $\mathbf{G}_\tau$ as our approximation solution for G, we have

$$h_A(G) = \max_{1\le i\le q} \sum_{e\in A_i} \omega(e). \tag{18}$$

Since $\bar{\omega}(e) = \lceil \omega(e)/\tau \rceil$ for each edge $e \in E$, we have

$$\tau\bar{\omega}(e) \ge \omega(e) > \tau\Big(\bar{\omega}(e) - 1\Big). \tag{19}$$

Therefore, by Eq. (17) we have

$$h^*(G) > \max_{1\le i\le q} \sum_{e\in O_i} \tau\Big(\bar{\omega}(e) - 1\Big) = \max_{1\le i\le q} \Big\{ -\tau|O_i| + \sum_{e\in O_i} \tau\bar{\omega}(e) \Big\},$$

where $|O_i|$ denotes the number of edges in O_i. Since $|O_i| \le |E|$ for all i, $1 \le i \le q$, we have

$$h^*(G) > -\tau|E| + \max_{1\le i\le q} \sum_{e\in O_i} \tau\bar{\omega}(e). \tag{20}$$

Since $\mathbf{G}_\tau$ is an optimal orientation for G_τ (with respect to the weight $\bar{\omega}$), we clearly have

$$\max_{1\le i\le q} \sum_{e\in O_i} \bar{\omega}(e) \ge \max_{1\le i\le q} \sum_{e\in A_i} \bar{\omega}(e). \tag{21}$$

By Eqs. (19)–(21) we have

$$\begin{aligned} h^*(G) &> -\tau|E| + \max_{1\le i\le q} \sum_{e\in A_i} \tau\bar{\omega}(e) \\ &\ge -\tau|E| + \max_{1\le i\le q} \sum_{e\in A_i} \omega(e). \end{aligned} \tag{22}$$

Therefore, by Eqs. (18) and (22) we have

$$h^*(G) > -\tau|E| + h_A(G). \tag{23}$$

Let

$$\tau = \frac{\varepsilon z_A}{2|E|}. \tag{24}$$

Then, by Eqs. (11), (23) and (24) we have

$$h_A(G) - h^*(G) < \tau|E| = \frac{\varepsilon z_A}{2} \leq \varepsilon h^*(G).$$

We have thus verified Eq. (16).

[Computation time]

We then show that our algorithm finds the optimal orientation $\mathbf{G}_\tau$ for G_τ in time polynomial in both n and $1/\varepsilon$ for any real number $\varepsilon > 0$.

Since $\mathbf{G}_\tau$ is optimal for G_τ, by Eq. (21) we have

$$h^*(G_\tau) = h(\mathbf{G}_\tau) = \max_{1\leq i\leq q} \sum_{e\in A_i} \bar{\omega}(e) \leq \max_{1\leq i\leq q} \sum_{e\in O_i} \bar{\omega}(e). \tag{25}$$

We employ the approximation value z_A of Section 4.1 as the upper bound on $h^*(G)$. Then, by Eqs. (17), (19) and (25) we have

$$h^*(G_\tau) < \max_{1\leq i\leq q} \sum_{e\in O_i} \left(1 + \frac{\omega(e)}{\tau}\right) \leq |E| + \frac{h^*(G)}{\tau} \leq |E| + \frac{z_A}{\tau}.$$

By Eq. (24) we thus have $h^*(G_\tau) \leq (1 + 2/\varepsilon)|E|$, and hence we let $U = (1 + 2/\varepsilon)|E|$. Theorem 7 implies that we can find the optimal orientation $\mathbf{G}_\tau$ for G_τ in time $O(U^{2q}n)$ if q is a fixed constant. Therefore, $\mathbf{G}_\tau$ can be found in time

$$O\left(\left(|E| + \frac{2|E|}{\varepsilon}\right)^{2q} n\right) = O\left(\frac{n^{2q+1}}{\varepsilon^{2q}}\right).$$

Note that $|E| = O(n)$ since G is a cactus.

This completes the proof of Theorem 8. □

5 Conclusions

In this paper, we gave several results for MIN-SUM ORIENTATION and MIN-MAX ORIENTATION, mainly the following three results. We first showed that both problems are strongly NP-hard for planar graphs of maximum degree 4 even if all edge-weights are identical. We then showed that both problems can be solved in polynomial time for cycles. Finally, we gave an FPTAS for MIN-MAX ORIENTATION on cacti if q is a fixed constant.

As we have shown in Theorem 6, there is a polynomial-time 2-approximation algorithm for MIN-MAX ORIENTATION on cacti even if q is not a fixed constant. It remains open to obtain a polynomial-time constant-factor approximation algorithm for both problems (for a class of graphs larger than cacti) when q is *not* a fixed constant.

Acknowledgements

We thank the referees for their fruitful comments, one of which leads us to improvements of the hardness analyses for MIN-SUM ORIENTATION.

References

1. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. Society for Industrial and Applied Mathematics, Philadelphia (1999)
2. Chvátal, V., Thomassen, C.: Distances in orientations of graphs. J. Combinatorial Theory, Series B 24, 61–75 (1978)
3. Eggemann, N., Noble, S.D.: Minimizing the oriented diameter of a planar graph. Electronic Notes in Discrete Mathematics 34, 267–271 (2009)
4. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
5. Guibas, L.J., Hershberger, J.E., Mitchell, J.S.B., Snoeyink, J.S.: Approximating polygons and subdivisions with minimum-link paths. International Journal of Computational Geometry and Applications 3, 383–415 (1993)
6. Hakimi, S.L., Schmeichel, E.F., Young, N.E.: Orienting graphs to optimize reachability. Information Processing Letters 63, 229–235 (1997)
7. Ito, T., Miyamoto, Y., Ono, H., Tamaki, H., Uehara, R.: Route-enabling graph orientation problems. Proc. of the 20th Annual International Symposium on Algorithms and Computation (ISAAC 2009), LNCS 5878, 403–412 (2009)
8. Le-Anh, T., de Koster, M.B.M.: A review of design and control of automated guided vehicle systems. European Journal of Operational Research 171, 1–23 (2006)
9. Lee, C.-Y., Lei, L., Pinedo, M.: Current trends in deterministic scheduling. Annals of Operations Research 70, 1–41 (1997)
10. Medvedovsky, A., Bafna, V., Zwick, U., Sharan, R.: An algorithm for orienting graphs based on cause-effect pairs and its applications to orienting protein networks. Proc. of the 8th Workshop on Algorithms in Bioinformatics (WABI 2008), LNBI 5251, 222–232 (2008)
11. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)
12. Robbins, H.E.: A theorem on graphs with an application to a problem of traffic control. American Mathematical Monthly 46, 281–283 (1939)
13. Uehara, R., Uno, Y.: On computing longest paths in small graph classes. International Journal of Foundations of Computer Science 18, 911–930 (2007)
14. Vazirani, V.V.: Approximation Algorithms. Springer, Heidelberg (2001)