Covering partially directed graphs with directed paths

Romeo Rizzi\textsuperscript{a}, Marco Rospocher\textsuperscript{b}

\textsuperscript{a}Dipartimento di Matematica ed Informatica (DIMI), Università di Udine, Via delle Scienze 208, I-33100 Udine, Italy
\textsuperscript{b}Dipartimento di Informatica e Telecomunicazioni (DIT), Università di Trento, Via Sommarive 14, I-38050 Povo, Trento, Italy

Received 13 November 2003; received in revised form 27 December 2005; accepted 11 January 2006
Available online 2 May 2006

Abstract

We consider graphs which contain both directed and undirected edges (partially directed graphs). We show that the problem of covering the edges of such graphs with a minimum number of edge-disjoint directed paths respecting the orientations of the directed edges is polynomially solvable. We exhibit a good characterization for this problem in the form of a min–max theorem. We introduce a more general problem including weights on possible orientations of the undirected edges. We show that this more general weighted formulation is equivalent to the weighted bipartite $b$-factor problem. This implies the existence of a strongly polynomial algorithm for this weighted generalization of Euler’s problem to partially directed graphs (compare this with the negative results for the mixed Chinese postman problem). We also provide a compact linear programming formulation for the weighted generalization that we propose.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Partially directed graphs; Good characterization; $b$-factor; Mixed Chinese postman problem

1. Introduction

Let $G$ be a partially directed graph, i.e. a graph which contains both directed and undirected edges. We consider the problem of covering the edges of $G$ with the minimum number of directed paths, in such a way that:

1. directed edges are covered accordingly to their orientation;
2. a path may cover any edge only once (paths are edge-simple);
3. two or more paths cannot cover the same edge (paths are edge-disjoint).

Intuitively, the problem that we consider is the one of tracing with a pen the edges of a graph, in which some edges can be traced in only one direction and some in both. Every edge must be traced exactly once, but it is allowed to jump with the pen from a node to another. The goal is to minimize the number of jumps of the pen.

This problem has been introduced for the first time in [1]. In that paper, the authors characterized those graphs which contain an Eulerian directed path or cycle; furthermore, they provided a necessary and sufficient condition for covering the edges of a graph by $n$ edge-disjoint edge-simple directed paths when every node has even degree. In this paper,
we pick up the challenge posed by the authors of [1] to provide a necessary and sufficient condition which holds for arbitrary node degrees.

We next consider a weighted version of the problem previously introduced, in which there are two weights for each undirected edge of the graph, which state the cost of traversing the edge in one or the other direction, and there is a weight associated to each node of the graph, which states the cost of starting a new path from that node. The goal is to cover the edges of the graph with a minimum weight collection of edge-disjoint edge-simple directed paths, where the weight of a collection of paths is given by the sum over the undirected edges of the weights corresponding to the direction in which they are traversed, plus the sum, over all nodes, of the number of paths starting at that node times the corresponding weight. We show that this weighted generalization is equivalent to the weighted bipartite $b$-factor problem, which is known to be strongly polynomially solvable [4,16]. Exploiting the insight of this equivalence, we also provide a compact linear programming formulation for the problem considered.

We remark that this weighted generalization of Euler’s problem to partially directed graphs is different from the mixed Chinese postman problem (mixed CPP), which in fact is NP-hard [14]: in the former each edge is covered by exactly one edge-simple path in a feasible solution, while in the latter an edge can be traversed more than once by the closed path. That is, a feasible solution for this weighted generalization of Euler’s problem selects one of the two possible orientations of each undirected edge, while in a postman tour an undirected edge can be traversed in both directions.

1.1. Historical perspective

Covering graphs with paths is a classical and fundamental problem in graph theory. The first known written reference on such problem (and also on graph theory) is the milestone work of Euler [6], in which he gave a characterization, although he proved only the necessary condition, of those undirected graphs that can be covered with just one edge-simple path or circuit. The sufficiency of Euler’s condition has been proved by Hierholzer [9]. In 1847, Listing [12] stated a necessary and sufficient condition for covering the edges of an undirected graph with the minimum number of edge-disjoint edge-simple paths, although the first proof was due to Lucas [13].

A characterization of those directed graphs that can be covered with just one edge-simple path or circuit has been provided by König [11].

In the last century many generalizations followed. The most famous and important one is the Chinese postman problem (CPP), introduced for the first time in 1962 by Mei Gu Guan [8]: each edge of the graph has a positive real weight, the goal is to find a directed closed path in $G$ of minimum weight that traverses each edge at least once. Edmonds and Johnson [5], and Christofides [3] showed that there exists a polynomial time algorithm to solve the CPP for undirected graphs. The algorithm proposed in [5] works also for directed graphs. In 1976, Papadimitriou [14] showed that the problem is NP-hard when the graph contains both directed and undirected edges (mixed CPP).

It could have seemed at that point that the generalizations of Euler’s problem to partially directed graphs, although quite natural, were at a dead end. Despite of this, in 1999 Barnette and Gillett [1] got back to original Euler’s formulation of the problem, studying it for graphs with both directed and undirected edges: they provided a characterization of those partially directed graphs that can be covered with just one Eulerian directed path or circuit. In our paper, we lead to completion the research line opened by Barnette and Gillett, solving the problem of covering with the minimum number of edge-disjoint edge-simple directed paths the edges of a partially directed graph. Furthermore, introducing weights on possible orientations of undirected edges, we propose a weighted generalization of Euler’s problem to partially directed graphs. This weighted formulation, differently from the mixed CPP, is polynomially solvable.

1.2. Paper’s outline

In Section 2, you can find the basic notions used throughout the paper and the statement of the optimal edge orientation problem (OEOP), an equivalent formulation to the unweighted problem previously introduced. In Section 3, a local necessary and sufficient condition that characterizes completely optimal orientations for the OEOP is given. Exploiting this condition, we describe a polynomial time algorithm for solving the problem. In Sections 4 and 5, we provide an alternative characterization for the OEOP in the form of a min–max formula. In Section 6, we introduce the minimum weight edge orientation problem (MWEOP), a weighted generalization of the OEOP. We prove that the MWEOP is polynomially solvable by means of a reduction to the weighted bipartite $b$-factor problem. Indeed, we show
that the MWEOP and the weighted bipartite \(b\)-factor problem are equivalent. In Section 7, we give a compact linear programming formulation of the MWEOP and we prove that the MWEOP can be solved in polynomial time by means of linear programming, without reducing it to the weighted \(b\)-factor problem.

2. Optimal edge orientation problem

A partially directed graph (p.d.g.) is a graph \(G = (V; A, E)\) where \(A\) is a set of directed edges and \(E\) is a set of undirected edges.

A path \(P\) in a p.d.g. \(G\) is a sequence \(v_0, e_1, v_1, e_2, \ldots, e_k, v_k\) such that:

- \(v_0, \ldots, v_k \in V;\)
- \(e_1, \ldots, e_k \in A \cup E;\)
- going from \(v_0\) to \(v_k\), all the directed edges in the path are traversed from the tail to the head.

A path such that \(v_0 = v_k\) is called cycle. A path \(P\) is edge-simple if there is no repetition of edges. From now on, all paths are assumed to be edge-simple, unless explicitly reported.

Two or more paths are said to be edge-disjoint if their edge-sets are disjoint.

Consider the following problem.

**Problem 2.1** (edge-disjoint edge-simple directed path covering). Let \(G\) be a p.d.g. Cover the edges of \(G\) with the minimum number of edge-disjoint edge-simple directed paths.

In order to solve this problem, we introduce some definitions and basic results.

**Definition 2.1** (Orientation). Let \(G = (V; A, E)\) be a p.d.g.. An orientation \(G_\ast = (V; A \cup E_\ast, \emptyset)\) (or, simply, \(G_\ast = (V; A \cup E_\ast)\)) of \(G\) is a directed graph arising from \(G\) by assigning an orientation to each undirected edge in \(G\). We call oriented edges the directed edges in \(E_\ast\).

Consider a p.d.g. \(G = (V; A, E)\). Let \(V' \subseteq V, A' \subseteq A\) and \(E' \subseteq E\). We define:

- \(\delta^+_{G,A'}(V') := \{u \in A' \mid u \in V', v \notin V'\}\) and \(d^+_G(A'(V')) := |\delta^+_{G,A'}(V')|\);
- \(\delta^-_{G,A'}(V') := \{u \in A' \mid u \notin V', v \in V'\}\) and \(d^-_{G,A'}(V') := |\delta^-_{G,A'}(V')|\);
- \(\delta^+_{G,A'}(V') := \delta^+_{G,A'}(V') + \delta^-_{G,A'}(V')\) and \(d^-_{G,A'}(V') := d^+_G(A'(V')) - d^-_{G,A'}(V')\);
- \(\delta^u_{G,E'}(V') := \{uv \in E' \mid u \in V', v \notin V'\}\) and \(d^u_{G,E'}(V') := |\delta^u_{G,E'}(V')|\);
- \(\delta_G(V') := \delta^+_{G,A'}(V') + \delta^-_{G,E'}(V')\) and \(d_G(V') := d^+_G(A'(V')) + d^u_{G,E'}(V')\).

We call \(d_G(V')\) the degree of a subset \(V'\) in a p.d.g. \(G\). In the case \(V' = \{v\}\), we will use the shorthands \(\delta_G(v), d_G(v)\) and so on.

Given a node \(v\) in a directed graph \(G\), we define the surplus \(s_G(v)\) of \(v\) in the following way:

\[
s_G(v) := \begin{cases} 
  d_G(v) & \text{if } d_G(v) > 0, \\
  0 & \text{otherwise}.
\end{cases}
\]

Given a directed graph \(G\), we define its cost as \(c(G) := \sum_{v \in V} s_G(v)\).

**Problem 2.2** (optimal edge orientation (OEOP)). Let \(G\) be a p.d.g.. Find an orientation \(G_{\text{opt}}\) of \(G\) such that the cost of \(G_{\text{opt}}\) is minimum. We will call \(G_{\text{opt}}\) an optimal orientation of \(G\).

Problem 2.2 is strictly related to Problem 2.1, due to the following theorem.

**Theorem 2.1.** If \(G\) is a directed graph, then \(G\) can be covered by at most \(n\) edge-disjoint edge-simple paths if and only if \(c(G) \leq n\).
For this reason, from now on, we will focus our attention on solving Problem 2.2. See [2] for a proof of Theorem 2.1.

A conservation of flow argument (Kirchhoff’s Second Law) relates the degree of a subset of nodes \( U \) of a digraph \( G \), to the degree of the nodes in \( U \): we have that, 
\[
d_{G}(U) = \sum_{v \in U} d_{G}(v).
\]
Therefore, 
\[
\sum_{v \in U} s_{G}(v) \geq d_{G}(U).
\]

3. Local optimality: a necessary and sufficient condition for optimality

**Property 3.1** (local optimality). Given a p.d.g. \( G = (V; A, E) \), we say that an orientation \( G_{\text{loc}} = (V; A \cup E_{\text{loc}}) \) of \( G \) is locally optimal if:

1. \( E_{\text{loc}} \) does not contain directed paths from nodes \( u \), with \( d_{G_{\text{loc}}}(u) > 1 \), to nodes \( v \), with \( d_{G_{\text{loc}}}(v) < 0 \);
2. \( E_{\text{loc}} \) does not contain directed paths from nodes \( u \), with \( d_{G_{\text{loc}}}(u) < 0 \), to nodes \( v \), with \( d_{G_{\text{loc}}}(v) < -1 \);

**Observation 3.1.** An optimal orientation is also locally optimal.

**Proof.** Let \( G = (V; A, E) \) be a p.d.g.. Let \( G_{\text{opt}} = (V; A \cup E_{\text{opt}}) \) be an optimal orientation for \( G \). Suppose for a contradiction that there exists a directed path \( P_{x,y}^{G_{\text{opt}}} \) in \( G_{\text{opt}} \) which connects a node \( x \), with \( d_{G_{\text{opt}}}(x) > 1 \), to a node \( y \), with \( d_{G_{\text{opt}}}(y) < 0 \). Clearly, flipping (i.e. reversing the orientation of) all the edges of \( P_{x,y}^{G_{\text{opt}}} \) in \( G_{\text{loc}} \), we obtain a new graph \( G_{\text{opt}} \) with the following property:

1. \( d_{G_{\text{opt}}'}(z) = d_{G_{\text{opt}}}(z) \) for each node \( z \neq x, y \);
2. \( d_{G_{\text{opt}}'}(x) = d_{G_{\text{opt}}}(x) - 2 \) and \( d_{G_{\text{opt}}'}(y) = d_{G_{\text{opt}}}(y) + 2 \).

Therefore, \( c(G_{\text{opt}}') < c(G_{\text{opt}}) \). Absurd. \( \square \)

The following theorem states that the converse is also true.

**Theorem 3.1.** A locally optimal orientation is optimal.

**Proof.** Let \( G = (V; A, E) \) be a p.d.g.. Let \( G_{\text{opt}} = (V; A \cup E_{\text{opt}}) \) be an optimal orientation for \( G \). Let \( G_{\text{loc}} = (V; A \cup E_{\text{loc}}) \) be a locally optimal orientation for \( G \). Assume that \( (G, G_{\text{loc}}) \) is a minimal counterexample for our theorem, that is, \( c(G_{\text{loc}}) > c(G_{\text{opt}}) \) and \( |E(G)| \) is as small as possible.

**Claim 1.** \( E_{\text{opt}} \cap E_{\text{loc}} = \emptyset \). Assume that there exists an edge \( e \in E \) oriented in the same way in \( G_{\text{loc}} \) and \( G_{\text{opt}} \). Then, replacing \( e \) in \( G \) with its oriented version in \( E_{\text{opt}} \cap E_{\text{loc}} \), we obtain a new graph \( G' \) such that \( (G', G_{\text{loc}}) \) is a counterexample to our theorem, and \( |E(G')| < |E(G)| \). Absurd.

**Claim 2.** \( E_{\text{loc}} \) cannot contain directed cycles. In fact, assume that \( E_{\text{loc}} \) contains a directed cycle \( C \). If we flip all the edges of \( C \) in \( G_{\text{loc}} \), we obtain a new graph \( G_{\text{flip}} \) such that the degree of each node in \( G_{\text{flip}} \) is equal to the degree of the corresponding node in \( G_{\text{loc}} \), and hence \( c(G_{\text{flip}}) = c(G_{\text{loc}}) > c(G_{\text{opt}}) \). Furthermore, \( G_{\text{flip}} \) is locally optimal, since \( G_{\text{loc}} \) is locally optimal, \( C \) is a directed cycle and reversing the orientation of a directed cycle does not affect the reachability relation among nodes. Finally, \( G_{\text{flip}} \cap G_{\text{opt}} \) is exactly the edge-set of \( C \). Absurd, since this contradicts Claim 1.

**Claim 3.** \( E_{\text{loc}} \) cannot contain a directed path from a node with degree equal to 1 to a node with degree equal to \(-1\). Suppose for a contradiction that there exists a path \( P_{u,v}^{G_{\text{loc}}} \) from \( u \) to \( v \) in \( E_{\text{loc}} \), such that \( d_{G_{\text{loc}}}(u) = +1 \) and \( d_{G_{\text{loc}}}(v) = -1 \). Flipping all the edges of \( P_{u,v}^{G_{\text{loc}}} \) in \( G_{\text{loc}} \), we obtain a new graph \( G_{\text{flip}} \) with the following property:

1. \( d_{G_{\text{flip}}}(w) = d_{G_{\text{loc}}}(w) \) for each node \( w \neq u, v \);
2. \( d_{G_{\text{flip}}}(u) = -1 \) and \( d_{G_{\text{flip}}}(v) = +1 \).

Hence, \( c(G_{\text{flip}}) = c(G_{\text{loc}}) > c(G_{\text{opt}}) \).
Furthermore, $G_{\text{flip}}$ is locally optimal. In fact, assume that there exists a node $y$ in $G_{\text{flip}}$ with $d_{G_{\text{flip}}}(y) < 0$, reachable from a node $x$, with $d_{G_{\text{flip}}}(x) > 1$. Let $P'_{x,y}$ be a directed path from $x$ to $y$ in $G_{\text{flip}}$. Clearly, $P'_{x,y}$ must contain some edges in $P_{v,u}$ (which is the path in $G_{\text{flip}}$ corresponding to $P_{v,u}^G$ after flipping its edges), otherwise $G_{\text{loc}}$ would not be locally optimal. Let $z$ be the first node, going from $x$ to $y$ in $P'_{x,y}$, that also belongs to $P_{v,u}$. By this assumption, $z$ is reachable from $x$ not only in $G_{\text{flip}}$, but also in $G_{\text{loc}}$. Furthermore, since $z$ is a node on $P_{v,u}$, $v$ is reachable from $z$ in $G_{\text{loc}}$. But then, $v$ is reachable from $x$ in $G_{\text{loc}}$. Absurd, since $G_{\text{loc}}$ is locally optimal.

This proves the first statement of Property 3.1: the second one can be proved in a very similar fashion.

Now, as for Claim 2, observe that $G_{\text{flip}} \cap G_{\text{opt}}$ is exactly the edge-set of $P_{v,u}$, which leads to a contradiction with Claim 1.

So we can assume that in $G_{\text{loc}}$ no negative degree node is reachable through only oriented edges from a node with positive degree.

Consider the subset $S \subseteq V$ composed by the nodes with positive degree in $G_{\text{loc}}$ together with the nodes reachable from them through only oriented edges. Let $G_*$ be an arbitrary orientation of $G$. Clearly the following inequalities hold:

$$c(G_*) = \sum_{v \in V} s_{G_*}(v) \geq \sum_{v \in S} s_{G_*}(v) \geq \sum_{v \in S} d_{G_*}(v) = d_{G,A}(S) + d_{G_*,E_*}(S) \geq d_{G,A}(S) - d_{G,E}(S) = d_{G_{\text{loc}}}(S) = \sum_{v \in S} d_{G_{\text{loc}}}(v) = \sum_{v \in S} s_{G_{\text{loc}}}(v) = c(G_{\text{loc}}).$$

Absurd, since we assumed that $G_{\text{loc}}$ is not an optimal orientation. □

**Example.** Consider the p.d.g. $G$ represented in Fig. 1(a). $G$ is the graph chosen in [1] to show that the necessary and sufficient condition proposed in that work does not hold for graphs with more than two odd nodes. Note that three paths are necessary to cover $G$. Let $G_*$ be the orientation proposed in Fig. 1(b). Note that, $G_*$ is (locally) optimal and $c(G_*) = d_{G_*}(a) + d_{G_*}(b) = 3$.

Observation 3.1 and Theorem 3.1 prove that the problem of finding an optimal orientation and the one of finding a locally optimal orientation are equivalent.

Algorithm 1 returns a locally optimal orientation of a given partially directed graph in polynomial time.

---

**Algorithm 1. L.O.O. Algorithm**

**Input:** A partially directed graph $G = (V; A, E)$.

**Output:** A locally optimal orientation $G_{*} = (V; A \cup E_{*})$ of $G$.

Choose an arbitrary orientation $G_{*}$ of $G$;

while $\exists u, v \in V, d(u) > 0, d(u) \cdot d(v) < 1, v$ reachable from $u$ using only oriented edges do

consider a path $P_{u,v} \subseteq E_{*}$ and flip all its edges.
Note. Algorithm 1 can be implemented to compute a locally optimal orientation in \(O(|A| + |E|(|V| + |E|))\) time. Although more efficient algorithms than Algorithm 1 may exist, our purpose here is just to provide a good characterization for the OEP. Indeed, Algorithm 1 here above makes clear that local optimality (and, hence, optimality) can be checked in polynomial time. These considerations already show that Problem 2.2 is somehow well-characterized. In the next two sections, we will provide a more classical good characterization in terms of a min–max formula.

4. A lower bound for the optimal edge orientation problem

Consider a p.d.g. \(G = (V; A, E)\). We say that a node \(v \in V(G)\) is odd (resp. even) if \(d_{G,A}^+(v) + d_{G,A}^-(v) + d_{G,E}^a(v)\) is odd (resp. even).

Given a subset \(V_* \subseteq V\), we indicate with Odd\((V_*)\) (resp. Even\((V_*)\)) the subset of odd (resp. even) nodes in \(V_*\).

**Lemma 4.1.** Let \(G = (V; A)\) be a digraph. Let \(U\) be a subset of \(V\). Then,

\[
\sum_{v \in U} s_G(v) \geq \frac{d_G(U) + |\text{Odd}(U)|}{2}.
\]

**Proof.** Consider the subset Odd\((U)\). Note that, \(d_G(v) \neq 0\) for each node \(v \in \text{Odd}(U)\). Hence, we can partition Odd\((U)\) into two disjoint subsets, Odd\(_{>0}(U) := \{v \in \text{Odd}(U) \mid d(v) > 0\}\) and Odd\(_{<0}(U) := \{v \in \text{Odd}(U) \mid d(v) < 0\}\).

The following inequalities are clearly satisfied:

\[
2 \sum_{v \in U} s_G(v) \geq \sum_{v \in U \setminus \text{Odd}_{<0}(U)} s_G(v) + \sum_{v \in \text{Odd}_{<0}(U)} s_G(v) \\
\geq d_G(U \setminus \text{Odd}_{<0}(U)) + |\text{Odd}_{>0}(U)| \\
= d_G(U \setminus \text{Odd}_{<0}(U)) - |\text{Odd}_{<0}(U)| + |\text{Odd}(U)| \\
\geq d_G(U) + |\text{Odd}(U)|.
\]

Note that, since \(d_G(U)\) and \(|\text{Odd}(U)|\) have the same parity, \((d_G(U) + |\text{Odd}(U)|)/2\) is always an integer. □

Consider a p.d.g. \(G = (V; A, E)\). Given two subsets \(U_1, U_2 \subseteq V\) with \(U_1 \cap U_2 = \emptyset\), we define \(E_{(U_1, U_2)} := \{uv \in E(G) \mid u \in U_1, v \in U_2\}\).

**Definition 4.1.** Given a p.d.g. \(G = (V; A, E)\), we say that \(T(G) = (V_1, V_3, V_2)\) is a tripartition of \(G\) if

1. \(V_1 \cup V_3 \cup V_2 = V\);
2. \(V_1 \cap V_3 = V_1 \cap V_2 = V_3 \cap V_2 = \emptyset\).

To every tripartition \(T(G) = (V_1, V_3, V_2)\) we associate a value \(\text{val}(T(G))\) defined in the following way:

\[
\text{val}(T(G)) := d_{G,A}(V_1)) - d_{G,E}(V_1) + \frac{\text{cap}_G(V_3) + |\text{Odd}(V_3)|}{2},
\]

where \(\text{cap}_G(V_3) := d_{G,A}(V_3) + |E_{V_3,V_1}| - |E_{V_3,V_2}|\). Again, note that \(\text{val}(T(G))\) is an integer.

**Example.** Consider the p.d.g. \(G\) represented in Fig. 1(a). An example of tripartition \(T(G)\) of \(G\) is the following:

\(V_1 = \{a\}, V_3 = \{d, e\}, V_2 = \{b, c\}\). The value of \(T(G)\) is \(\text{val}(T(G)) = 2 + \frac{2+2}{2} = 2\).

**Lemma 4.2 (lower bound).** The value of a tripartition is a lower bound for the cost of an optimal solution for the OEP.

**Proof.** Let \(G = (V; A, E)\) be a p.d.g. Let \(T(G) = (V_1, V_3, V_2)\) be a tripartition of \(G\). Let \(G_\circ\) be an arbitrary orientation of \(G\).
The following inequalities hold:
\[ c(G_+) \geq \sum_{v \in V_1} s_{G_+}(v) + \sum_{v \in V_3} s_{G_+}(v) \]
\[ \geq d_{G_+,A}(V_1) + d_{G_+,E}(V_1) + \sum_{v \in V_3} s_{G_+}(v) \]
\[ = d_{G_+,A}(V_1) - d_{G,E}^-(V_1) + 2d_{G,E}^+(V_1) + \sum_{v \in V_3} s_{G_+}(v) \]
\[ \geq d_{G_+,A}(V_1) - d_{G,E}^-(V_1) + 2d_{G,E}^+(V_1) + \frac{d_{G_+}(V_3) + |\text{Odd}(V_3)|}{2} \]
\[ \geq \frac{d_{G_+}(V_3) + |\text{Odd}(V_3)|}{2} \]
\[ \geq \text{val}(T(G)). \quad \square \]

5. Good characterization for the optimal edge orientation problem

In the previous section we have proved that the value of a tripartition is a lower bound for the cost of an optimal solution for the OEO. In this section, we will show that to each optimal orientation \( G_{\text{opt}} \) of a p.d.g. \( G \) corresponds a tripartition \( T_{\text{opt}}(G) \) such that its value is equal to the cost of orientation \( G_{\text{opt}} \). As a consequence, we will obtain a good characterization in the form of a min–max formula for Problem 2.2.

**Definition 5.1.** We define the tripartition \( T_{\text{opt}}(G) = (V_1, V_3, V_2) \) induced by a (locally) optimal orientation \( G_{\text{opt}} = (V; A \cup E_{\text{opt}}) \) of a p.d.g. \( G = (V; A, E) \) in the following way:

- \( V_1 \) is the set of nodes with degree greater than 1, together with the nodes reachable from them using only oriented edges;
- \( V_2 \) is the set of nodes with degree less than −1, together with the nodes from which they are reachable using only oriented edges;
- \( V_3 \) is the set of the remaining nodes.

**Observation 5.1.** Let \( G_{\text{opt}} = (V; A \cup E_{\text{opt}}) \) be a (locally) optimal orientation of a p.d.g. \( G = (V; A, E) \). Let \( T_{\text{opt}}(G) = (V_1, V_3, V_2) \) be the tripartition induced by \( G_{\text{opt}} \). The following statements hold:

1. \( V_1 \) contains only nodes with non-negative degree;
2. \( V_2 \) contains only nodes with non-positive degree;
3. \( V_3 \) contains only nodes with degree equal to −1, 0 and +1;
4. the edges in \( E_{\text{opt}} \) corresponding to edges in \( E_{(V_1, V_3)} \) are oriented from \( V_3 \) to \( V_1 \);
5. the edges in \( E_{\text{opt}} \) corresponding to edges in \( E_{(V_1, V_2)} \) are oriented from \( V_2 \) to \( V_1 \);
6. the edges in \( E_{\text{opt}} \) corresponding to edges in \( E_{(V_3, V_2)} \) are oriented from \( V_2 \) to \( V_3 \).

**Proof.** Follows from the local optimality of orientation \( G_{\text{opt}} \) and the definition of \( T_{\text{opt}}(G) \). \( \square \)

**Observation 5.2.** Let \( G_{\text{opt}} = (V; A \cup E_{\text{opt}}) \) be a (locally) optimal orientation of a p.d.g. \( G = (V; A, E) \). Let \( T_{\text{opt}}(G) = (V_1, V_3, V_2) \) be the tripartition induced by \( G_{\text{opt}} \). Then, the following statements hold:

1. \( c(G_{\text{opt}}) = \sum_{v \in V_1} s_{G_{\text{opt}}}(v) + \sum_{v \in V_3} s_{G_{\text{opt}}}(v) \);
Theorem 5.1. Let $\sum_{v \in V_1} s_{G_{opt}}(v) = d_{G_{opt}}(V_1) = d_{G,A}(V_1) - d_{G,E}(V_1)$;
3. $\text{cap}_G(V_3) = d_{G_{opt}}(V_3)$.

Proof. Follows directly by Observation 5.1. □

Let $\#_V(+) - 1$ (resp. $\#_V(-1)$) be the number of nodes $v \in V_3$ with $d_{G_{opt}}(v) = +1$ (resp. $d_{G_{opt}}(v) = -1$).

Observation 5.3. Let $G_{opt} = (V; A \cup E_{opt})$ be a (locally) optimal orientation of a p.d.g. $G = (V; A, E)$. Let $T_{opt}(G) = (V_1, V_3, V_2)$ be the tripartition induced by $G_{opt}$. Then,

$$\sum_{v \in V_1} s_{G_{opt}}(v) = \#_V(+) - 1 - \#_V(-1) = \frac{\text{cap}_G(V_3) + |\text{Odd}(V_3)|}{2}. \tag{2.3}$$

Proof. The first equality follows directly by Observation 5.1. By Observations 5.1 and 5.2, we have that $\text{cap}_G(V_3) = d_{G_{opt}}(V_3) - \#_V(+) - 1 - \#_V(-1)$. Then,

$$\frac{\text{cap}_G(V_3) + |\text{Odd}(V_3)|}{2} = \#_V(+) - 1 - \#_V(-1) = \frac{\#_V(+) - 1 + \#_V(-1)}{2}. \tag{2.3}$$

Lemma 5.1. Let $G_{opt} = (V; A \cup E_{opt})$ be a (locally) optimal orientation of a p.d.g. $G = (V; A, E)$. Then, $G_{opt}$ induces a tripartition $T_{opt}(G) = (V_1, V_3, V_2)$ on $G$ such that

$$\text{val}(T_{opt}(G)) = c(G_{opt}). \tag{2.5}$$

Proof. By Observations 5.2 and 5.3 we have that,

$$c(G_{opt}) = \sum_{v \in V_1} s_{G_{opt}}(v) + \sum_{v \in V_3} s_{G_{opt}}(v)$$
$$= d_{G,A}(V_1) - d_{G,E}(V_1) + \sum_{v \in V_3} s_{G_{opt}}(v)$$
$$= d_{G,A}(V_1) - d_{G,E}(V_1) + \frac{\text{cap}_G(V_3) + |\text{Odd}(V_3)|}{2}$$
$$= \text{val}(T_{opt}(G)). \tag{2.5}$$

Example. Consider the p.d.g. $G$ represented in Fig. 1(a). As observed before, the orientation $G_*$ represented in Fig. 1(b) is optimal. Let $T_*(G)$ be the tripartition induced by $G_*$. By definition, $V_1 = \{a\}, V_3 = \{b, c, d, e\}, V_2 = \emptyset$. The value of $T_*(G)$ is $\text{val}(T(G)) = 2 + (-2 + 4)/2 = 3$.

Lemmas 4.2 and 5.1 imply that Problem 2.2 has a good characterization in the form of the following min–max theorem.

Theorem 5.1. Let $G = (V; A, E)$ be a p.d.g. Let $\Sigma(G)$ be the set of all the possible orientations of $G$. Let $\Xi(G)$ be the set of all the possible tripartitions of $G$. Then,

$$\min\{c(G_*) : G_* \in \Sigma(G)\} = \max\{\text{val}(T(G)) : T(G) \in \Xi(G)\}. \tag{5.1}$$

Hence, if we consider the decision version of Problem 2.2, i.e. “Is there an orientation $G_*$ of $G$ with $c(G_*) \leq k$?”, there always exists a polynomial length certificate for correctness of the answer, whatever the answer is. In details, if the answer is yes, as a certificate we may provide an orientation $G_*$ of $G$ with $c(G_*) \leq k$, while if the answer is no, as a certificate we may provide a tripartition $T(G)$ of $G$ with $\text{val}(T(G)) > k$. 

6. Minimum weight edge orientation

We now consider a generalization of the OEOP.

Assume that each undirected edge \( uv \) in a p.d.g. \( G = (V; A, E) \) has two weights, \( w^{or}(\overrightarrow{uv}) \) and \( w^{or}(\overrightarrow{vu}) \) which indicate, respectively, the cost of orienting edge \( uv \) from \( u \) to \( v \) and from \( v \) to \( u \) in any orientation.

Furthermore, assume that each node \( v \) in \( G \) has a weight \( w^p(v) \), which states the cost of starting a new path from node \( v \).

If \( G \) satisfies the two previous requirements, we say that it is a weighted p.d.g., and we indicate it with \( G = (V; A, E; w^{or}, w^p) \).

Given an orientation \( G_s = (V; A \cup E_s) \) of \( G \), the weight \( w(G_s) \) of \( G_s \) is defined as

\[
w(G_s) = \sum_{\overrightarrow{uv} \in E_s} w^{or}(\overrightarrow{uv}) + \sum_{v \in V} w^p(v) s_{G_s}(v).
\]

**Problem 6.1** (minimum weight edge orientation (MWEOP)). Let \( G = (V; A, E; w^{or}, w^p) \) be a weighted p.d.g.. Find an orientation of minimum weight.

Note that, Problem 6.1 includes as special subcases the following problems:

- *Optimal edge orientation problem*: just set \( w^{or}(\overrightarrow{uv}) = w^{or}(\overrightarrow{vu}) = 0 \) for each edge \( uv \) and \( w^p(v) = 1 \) for each node \( v \);
- *Minimum weight optimal edge orientation problem (MWOEOP)*: this problem consists in finding, among all orientations covering the graph with the minimum number of paths, one of minimum weight. To obtain this problem as a particular case of Problem 6.1, just set \( w^p(x) := 1 + \sum_{e \in E}(w^{or}(\overrightarrow{uv}) + w^{or}(\overrightarrow{vu})) \) for each node \( x \in V \).

In this section, we will show that Problem 6.1 is polynomially solvable by means of a reduction to the following degree constrained problem.

**Problem 6.2** (weighted \( b \)-factor). Let \( G = (U \cup V; E) \) be a bipartite undirected graph with color classes \( U, V \). Let \( w \in \mathbb{N}^E \) be a weight vector and let \( b \in \mathbb{N}^{U \cup V} \). A \( b \)-factor of \( G \) is a vector \( x \in \{0, 1\}^E \) satisfying

\[
x(\delta(v)) = b_v \quad \text{for each} \quad v \in U \cup V.
\]

Find a \( b \)-factor \( x \) of \( G \) such that \( \sum_{e \in E} w_e x_e \) is minimum.

Problem 6.2 is a particular case of the general matching problem; we recall that for the general matching problem a strongly polynomial-time algorithm is known ([4,16]; see also [7,15]).

Now we describe the reduction from Problem 6.1 to Problem 6.2.

Let \( G = (V; A, E; w^{or}, w^p) \) be a weighted p.d.g.. We build a new weighted bipartite undirected graph \( G_{\text{fac}} = (U_{\text{fac}} \cup V_{\text{fac}}; E_{\text{fac}}; w) \) as follow. Let \( k_v := \max\{d_{G,E}^a(v), [d_G(v)/2]\} \). We define:

\[
U_{\text{fac}} := V \quad \text{and} \quad V_{\text{fac}} := V \cup \{v_g\},
\]

where \( V_e := \{v_e : e \in E\} \), and

\[
E_{\text{fac}} := E_e \cup E_e = U_{\text{fac}} \cup V_{\text{fac}},
\]

where \( E_e = \{uv_v, vu_e : e \in E, e = uv\} \) and \( E_e = \{e_i = vv_i : v \in V, i = 1, \ldots, k_v\} \).

Now we define the weight vector \( w \) on \( G_{\text{fac}} \). Given an edge \( uv \) in \( G \), we set \( w_{uv} := w^{or}(\overrightarrow{uv}) \) and \( w_{vu} := w^{or}(\overrightarrow{vu}) \).

The weight of the edges in \( E_{\text{fac}} \) is assigned in the following way:

\[
w_{e_i} := \begin{cases} 2w^p(v) & \text{for } i = 1, \ldots, [d_G(v)/2] ; \\ w^p & \text{if } d_G(v) \text{ is odd} ; \\ 0 & \text{for the remaining } vv_{i} \text{ edges.} \end{cases}
\]

Note that,

\[
\sum_{i=j}^{k_v} w_{e_i} = w^p(v) [d_G(v) - 2(j - 1)], \quad (1)
\]
Now we define \( b_v \) for each node \( v \) in \( G_{\text{fac}} \). We set:

- \( b_v := k_v \) for each node \( v \in V \);
- \( b_v := 1 \) for each node \( v \in V_e \);
- \( b_{v^g} := \sum_{v \in V} b_v - |E| \).

**Lemma 6.1.** Let \( G = (V; A, E; w^w, w^p) \) be a weighted p.d.g., let \( G_* = (V; A \cup \overline{E}_*) \) be an orientation of \( G \), and let \( \langle G_{\text{fac}} = (U_{\text{fac}} \cup V_{\text{fac}}; E_{\text{fac}}; w) \rangle b \) be the instance of Problem 6.2 built accordingly to the above schema. Then, there exists a \( b \)-factor \( x \) of \( G_{\text{fac}} \) such that \( \sum_{v \in E_{\text{fac}}} w_ex_e = w(G_*) \).

**Proof.** We define the \( b \)-factor \( x \) of \( G_{\text{fac}} \) in the following way.

For each undirected edge \( e = uv \) in \( G \), we set:

- \( x_{uv;e} = 1 \) and \( x_{vu;e} = 0 \) if \( e \) has been oriented from \( u \) to \( v \) in \( G_* \);
- \( x_{uv;e} = 0 \) and \( x_{vu;e} = 1 \) otherwise.

For each node \( v \in V \), we set \( x_{e^v} = 0 \) for \( i = 1, \ldots, d^-_{G_*,E_*}(v) \), and \( x_{e^v} = 1 \) for \( i = d^-_{G_*,E_*}(v) + 1, \ldots, k_v \).

Notice that \( \sum_{e \in E_{\text{fac}}} x_{e} = w(G_*) \). In fact,

\[
\begin{align*}
    w(G_*) &= \sum_{\overline{e} \in \overline{E}_*} w^w(\overline{e}b) + \sum_{v \in V} w^p(v)s_{G_*}(v) \\
    &= \sum_{e \in E_e} w_{e}x_{e} + \sum_{v \in V} w^p(v)s_{G_*}(v).
\end{align*}
\]

Hence, it remains to prove that \( \sum_{e \in E_{\text{fac}}} w_ex_e = \sum_{v \in V} w^p(v)s_{G_*}(v) \). This equation is implied by the following stronger result, which holds for each \( v \in V \):

\[
\begin{align*}
    \sum_{i = d^-_{G_*,E_*}(v) + 1}^{k_v} w_{e^v} &= \max\{(d_G(v) - 2(d^-_{G_*,E_*}(v) + 1) - 1), 0\} \\
    &= \max\{d_G(v) - 1, 0\} \\
    &= \max\{d_G(v), 0\},
\end{align*}
\]

where the first equality is obtained applying Eq. (1). \( \square \)

**Lemma 6.2.** Let \( G = (V; A, E; w^w, w^p) \) be a weighted p.d.g., let \( \langle G_{\text{fac}} = (U_{\text{fac}} \cup V_{\text{fac}}; E_{\text{fac}}; w) \rangle b \) be the instance of Problem 6.2 built accordingly to the above schema, and let \( x \) be a \( b \)-factor of \( G_{\text{fac}} \). Then, there exists an orientation \( G_* = (V; A \cup \overline{E}_*) \) of \( G \) such that \( w(G_*) \leq \sum_{e \in E_{\text{fac}}} w_{e}x_{e} \).

**Proof.** Starting from \( x \), we first build a \( b \)-factor \( x' \) in canonical form. We say that a \( b \)-factor \( x' \) is in **canonical form** if \( x'_{e^v} = 1 \) implies that \( x'_{e_j} = 1 \), for each edge \( e^v \) with \( w_{e^v} < w_{e_j} \).

Let \( v_{out} := x(\delta(v) \cap E_{\text{fac}}) \), for each node \( v \) in \( G_{\text{fac}} \). We build \( x' \) in the following way:

- \( x'_{e^v} := x_{e^v} \) for each edge \( e \in E_{\text{fac}} \);
- \( x'_{e^v} := 1 \) if \( i = 1, \ldots, v_{out} \) and \( x'_{e^v} := 0 \) for the remaining \( vv^g \) edges.

Clearly, \( \sum_{e \in E_{\text{fac}}} w_{e}x'_{e} \leq \sum_{e \in E_{\text{fac}}} w_{e}x_{e} \).

Let \( G_* \) be the orientation of \( G \) obtained orienting each edge \( e = uv \) of \( G \) in \( G_* \) accordingly to the following rule:

- if \( x'_{v_{out}} = 1 \), then orient \( e \) in \( G_* \) from \( v \) to \( u \);
- otherwise, orient \( e \) in \( G_* \) from \( u \) to \( v \).

Clearly, \( \sum_{e \in E_{\text{fac}}} w_{e}x'_{e} = \sum_{e \in E_{\text{fac}}} w_{e}x_{e} = \sum_{\overline{e} \in \overline{E}_*} w^w(\overline{e}b) \). But, we also have that \( \sum_{e \in E_{\text{fac}}} w_{e}x'_{e} = \sum_{e \in V} w^p(v)s_{G_*}(v) \), where last equality follows from the application of Eq. (1). \( \square \)
Hence, we have proved that Problem 6.1 can be reduced to the problem of finding a b-factor of minimum weight in a bipartite graph. Indeed, the converse is also true. Here below, we describe this reduction.

The generic instance of Problem 6.2 is a couple \((G, b)\), where \(G = (U \cup V; E; w)\) is a weighted bipartite graph with color classes \(U, V\), and \(b\) is the degree-constraint vector on \(G\). We build a new weighted p.d.g. \(G_{or} = (V_{or}; A_{or}, E_{or}; w_{or}, w^{p})\) as follow. Let \(k_v := |2b_v - d_G(v)|\) for each node \(v\) of \(G\). We define

\[
V_{or} := U \cup V \cup \{z_v^i : v \in U \cup V, i = 1, \ldots, k_v\}, \quad E_{or} := E, \quad A_{or} := A_U \cup A_V,
\]

where

\[
A_U := \{\vec{z}_u^i : u \in U, d_G(u) > 2b_u, i = 1, \ldots, k_u\} \cup \{u\vec{z}_u^i : u \in U, d_G(u) < 2b_u, i = 1, \ldots, k_u\},
\]

and

\[
A_V := \{\vec{z}_v^i : v \in V, d_G(v) < 2b_v, i = 1, \ldots, k_v\} \cup \{v\vec{z}_v^i : v \in V, d_G(v) > 2b_v, i = 1, \ldots, k_v\}.
\]

Furthermore, we set the weight functions in the following way. For each undirected edge \(e = uv\) of \(G\), where \(u \in U\) and \(v \in V\), we define

\[
w_{or}(\vec{u}v) := 0 \quad \text{and} \quad w_{or}(v\vec{u}) := w_{uv},
\]

while we set

\[
w^p(v) := \begin{cases} 
1 + \sum_{e \in E} w_e & \text{for } v \in U \cup V, \\
0 & \text{otherwise}.
\end{cases}
\]

**Lemma 6.3.** Let \((G = (U \cup V; E; w), b)\) be an instance of Problem 6.2. Let \(G_{or} = (V_{or}; A_{or}, E_{or}; w_{or}, w^p)\) be the weighted p.d.g. built accordingly to the above schema. If there exists a b-factor \(x\) of \(G\), then there exists an orientation \(G_s = (V_{or}; A_{or} \cup E_s)\) of \(G_{or}\) such that \(d_{G_s}(v) = 0\) for each node \(v\) in \(U \cup V\) and \(w(G_s) = \sum_{e \in E} w_e x_e\).

**Proof.** We define \(E_s\) in the following way:

\[E_s := \{\vec{u}v : uv \in E, x_{uv} = 1\} \cup \{v\vec{u} : uv \in E, x_{uv} = 0\}.
\]

First, we prove that \(d_{G_s}(v) = 0\) for each node \(v\) in \(U \cup V\).

Consider a node \(u \in U\). If \(d_G(u) > 2b_u\), by the definitions of \(E_s\) and \(A_{or}\), we have that \(d_{G_s}^-(u) = b_u + (d_G(u) - 2b_u)\) and \(d_{G_s}^+(u) = d_G(u) - b_u\), while if \(d_G(u) < 2b_u\), we have that \(d_{G_s}^-(u) = b_u\) and \(d_{G_s}^+(u) = d_G(u) - b_u + (2b_u - d_G(u))\): in both cases, \(d_{G_s}^+(u) = d_{G_s}^-(u)\). With a very similar argumentation, we also have that \(d_{G_s}(v) = 0\) for each node \(v\) in \(V\).

Now, we focus our attention on the second claim of the lemma’s statement, the one regarding the cost of the orientation. By definition,

\[
w(G_s) = \sum_{\vec{u}v \in E_s} w_{or}(\vec{u}v) + \sum_{v \in V_{or}} w^p(v) s_{G_s}(v)
\]

\[= \sum_{\vec{u}v \in E_s, v \in V, u \in U} w_{or}(\vec{u}v) + 0
\]

\[= \sum_{e \in E} w_e x_e. \quad \square
\]

**Lemma 6.4.** Let \((G = (U \cup V; E; w), b)\) be an instance of Problem 6.2. Let \(G_{or} = (V_{or}; A_{or}, E_{or}; w_{or}, w^p)\) be the weighted p.d.g. built accordingly to the above schema. If there exists an orientation \(G_s = (V_{or}; A_{or} \cup E_s)\) of \(G_{or}\) such that \(d_{G_s}(v) = 0\) for each node \(v\) in \(U \cup V\), then there exists a b-factor \(x\) of \(G\) such that \(\sum_{e \in E} w_e x_e = w(G_s)\).

**Proof.** We define the b-factor \(x\) in the following way. For each edge \(uv \in E\), we set

\[
x_e := \begin{cases} 
1 & \text{if } uv \text{ has been oriented from } v \text{ to } u \text{ in } G_s; \\
0 & \text{otherwise}.
\end{cases}
\]
Now, we focus our attention on the weight of $b$-factor $x$.

$$\sum_{e \in E} w_e x_e = \sum_{\overrightarrow{uv} \in E, v \in V, u \in U} w^\text{or}(\overrightarrow{uv})$$

$$= \sum_{\overrightarrow{uv} \in E} w^\text{or}(\overrightarrow{uv})$$

$$= w(G_s),$$

where the second equality follows from $\sum_{\overrightarrow{uv} \in E} w^\text{or}(\overrightarrow{uv}) = 0$, and the third equality follows from $\sum_{v \in V} w^p(v) s_{G_s}(v) = 0$. □

Hence, in order to solve Problem 6.2, we can compute the MWEOP on the instance built accordingly to the above schema: if the result is an orientation such that $d(v) = 0$ for each node $v \in U \cup V$, then there exists a solution for the instance of Problem 6.2 considered, while if we obtain an orientation such that $d(v) \neq 0$ for some $v \in U \cup V$, by Lemmas 6.3 we have that the instance of Problem 6.2 considered does not admit a $b$-factor.

Lemmas 6.1–6.4 prove the following theorem.

**Theorem 6.1.** Problems 6.2 and 6.1 are equivalent.

### 7. Minimum weight edge orientation polytope

We conclude our investigations on Problem 6.1 with a structural result: we provide a compact linear programming formulation for the MWEOP.

Let $G = (V; A, E; w^\text{or}, w^p)$ be an instance of Problem 6.1. Only for the sake of establishing a consistent convention on the feasible orientations of the undirected edges, we assume that each node $v \in V$ is labeled with a unique integer $i(v) \in \{1, 2, \ldots, |V|\}$. Given a node $v \in V$, we indicate with $B(v)$ (resp. $F(v)$) the set of nodes $w$ such that $vw$ is an undirected edge of $G$ and $i(w) > i(v)$ (resp. $i(w) < i(v)$) (as a mnemonic aid, you can think that $B$ stays for backward, $F$ stays for forward).

For each undirected edge $e \in E$ we introduce a variable $x_e \in [0, 1]$, such that:

$$x_e = \begin{cases} 0 & \text{if we choose to orient } e \text{ from the node with minimum label to the other;} \\ 1 & \text{if we choose to orient } e \text{ in the other direction.} \end{cases}$$

Notice that the $\{0, 1\}$-cube defined by variables $x_e$ describes all the feasible orientations of a given p.d.g. $G$. Hence, finding an orientation of minimum cost is equivalent to minimize the objective function

$$\sum_{e \in E} w_e x_e + \sum_{v \in V} w^p(v) s_{G_s}(v) \tag{2}$$

over the $\{0, 1\}$-cube, where $G_s$ is the orientation of $G$ obtained orienting the undirected edges accordingly to the value of variables $x_e$. In detail,

$$s_{G_s}(v) = \max \left\{ d_{G,A}(v) + d_{G,E}^B(v) - 2 \left( \sum_{u \in E, u \in B(v)} (1 - x_{uv}) + \sum_{u \in E, u \in F(v)} x_{uv} \right), 0 \right\}. $$

Unfortunately, this objective function is not linear in variables $x_e$, and hence the compactness of the $\{0, 1\}$-cube defined by variables $x_e$ does not help to obtain a compact linear programming formulation of Problem 6.1.

In order to obtain a linear programming formulation of Problem 6.1, we introduce, for each node $v \in V$, $k_v := \max\{d_{G,E}^B(v), [d_G(v)/2]\}$ variables $z^v_i$, such that $z^v_i \in [0, 1]$ for each $i = 1, \ldots, k_v$.

For each node $v \in V$, let us define the quantity $\phi_v(x, z)$ as follows:

$$\phi_v(x, z) := \sum_{u \in E, u \in B(v)} x_{uv} + \sum_{u \in E, u \in F(v)} (1 - x_{uv}) + \sum_{i=1}^{k_v} z_i.$$

Now, we focus our attention on the weight of $b$-factor $x$. 

$$\sum_{e \in E} w_e x_e = \sum_{\overrightarrow{uv} \in E, v \in V, u \in U} w^\text{or}(\overrightarrow{uv})$$

$$= \sum_{\overrightarrow{uv} \in E} w^\text{or}(\overrightarrow{uv})$$

$$= w(G_s),$$

where the second equality follows from $\sum_{\overrightarrow{uv} \in E} w^\text{or}(\overrightarrow{uv}) = 0$, and the third equality follows from $\sum_{v \in V} w^p(v) s_{G_s}(v) = 0$. □
Finally, the linear system that we consider is the following:

\[
\begin{align*}
0 \leq x_e & \leq 1 \quad \forall e \in E; \\
0 \leq z_i^v & \leq 1 \quad \forall v \in V, i = 1, \ldots, k_v; \\
\phi_i(x, z) &= k_v \quad \forall v \in V.
\end{align*}
\]

\[ (3) \]

**Theorem 7.1.** Let \( G = (V, A, E; w^{or}, w^p) \) be an instance of Problem 6.1. Each orientation \( G_\ast \) of \( G \) corresponds to a non-empty family of integral solutions of (3). Conversely, to each integral solution of (3) there corresponds a unique orientation \( G_\ast \) of \( G \).

**Proof.** Let \( G_\ast = (V, A, E_\ast) \) be an orientation of \( G \). We build an integral solution \((x, z)\) of (3) in the following way. For each edge \( uv \in E \), with \( \overrightarrow{uv} \in E_\ast \), we set

\[ x_{uv} := \begin{cases} 1 & \text{if } i(u) > i(v); \\ 0 & \text{otherwise}. \end{cases} \]

To complete this partial assignment to a solution \((x, z)\) of (3), we set, for each node \( v \in V \),

\[ z_i^v := \begin{cases} 1 & \text{for } i = 1, \ldots, k_v - \left( \sum_{uv \in E, \ uv \in B(v)} x_e + \sum_{uv \in E, \ uv \in F(v)} (1 - x_e) \right); \\ 0 & \text{for the remaining indices}. \end{cases} \]

Clearly, \((x, z)\) is an integral solution of (3).

Conversely, let \((x, z)\) be an integral solution of (3). We define an orientation \( G_\ast = (V, A \cup E_\ast) \) of \( G \) as follow. For each undirected edge \( e = uv \) of \( G \), assuming that \( i(u) < i(v) \), we consider the following two cases:

- if \( x_e = 1 \), then we orient \( e \) in \( G_\ast \) from \( v \) to \( u \);
- if \( x_e = 0 \), then we orient \( e \) in \( G_\ast \) from \( u \) to \( v \).

Note that, the same orientation \( G_\ast \) is associated to all the solutions of (3) different from \((x, z)\) only for the value of some \( z_i^v \).

**Theorem 7.2.** Let \( G = (V; A, E; w^{or}, w^p) \) be an instance of Problem 6.1. Linear system (3) associated to \( G \) describes an integral polytope.

**Proof.** Let \( P \) be the polyhedron defined by linear system (3). By Theorem 7.1, \( P \) is not empty. Moreover, since the value of any variable cannot be negative or exceed 1, \( P \) is bounded: this implies that \( P \) is a polytope with at least one vertex. We will show that all vertices of \( P \) are integral.

Suppose for a contradiction that \((\tilde{x}, \tilde{z})\) is a non-integral vertex of \( P \). We first consider the case \( \tilde{x}_e \in \{0, 1\} \) for each \( e \in E \). Let us define \( V_{\text{frac}} := \{ v \in V : \exists \tilde{z}^v \text{ with } 0 < \tilde{z}^v < 1 \} \) and \( Z_{\text{frac}}(v) := \{ \tilde{z}^v : 0 < \tilde{z}^v < 1 \} \). Since \( \phi_i(\tilde{x}, \tilde{z}) = k_v \), we have that \( |Z_{\text{frac}}(v)| \geq 2 \) for each node \( v \in V_{\text{frac}} \). Without loss of generality, we can assume that if \( v \in V_{\text{frac}} \), then \( \tilde{z}^v_1, \tilde{z}^v_2 \in Z_{\text{frac}}(v) \).

We define two new vectors \((\tilde{x}, \tilde{z})\) and \((\hat{x}, \hat{z})\) as follow: they are the exact copy of \((\tilde{x}, \tilde{z})\), except that, for each node \( v \in V_{\text{frac}} \), we set:

- \( \tilde{z}^v_1 := \tilde{z}^v_1 + \varepsilon \) and \( \tilde{z}^v_2 := \tilde{z}^v_2 - \varepsilon; \)
- \( \hat{z}^v_1 := \tilde{z}^v_1 - \varepsilon \) and \( \hat{z}^v_2 := \tilde{z}^v_2 + \varepsilon; \)

where \( \varepsilon \in \mathbb{R}^+_0 \). Note that, for \( \varepsilon \) sufficiently small, both \((\tilde{x}, \tilde{z})\) and \((\hat{x}, \hat{z})\) are feasible solutions of (3). Furthermore, \((\tilde{x}, \tilde{z}) = \frac{1}{2}(\hat{x}, \hat{z}) + \frac{1}{2}(\tilde{x}, \tilde{z}) \). Absurd, since a vertex of the polytope cannot be a convex combination of two of its vectors.

Hence we can assume that there exists at least one edge \( e \in E \) such that \( x_e \notin \{0, 1\} \). Let \( G_{\text{frac}} \) be the subgraph of \( G \) induced by the edges \( e \in E \) such that \( 0 < x_e < 1 \).

We first claim that \( G_{\text{frac}} \) is acyclic. Indeed, suppose for a contradiction that \( G_{\text{frac}} \) contains a cycle \( C \) with edge-set \( \{u_0u_1, u_1u_2, \ldots, u_{c-1}u_0\} \). We define two new vectors \((\hat{x}, \hat{z})\) and \((\tilde{x}, \tilde{z})\) as follow: they are the exact copy of vector
\((\tilde{x}, \tilde{z})\) except for the variables associated to the edges in \(C\). Let \(\varepsilon \in \mathbb{R}_0^+\). For each edge \(e = u_i u_{i+1}\) in \(C\) (where, \(|i + 1| \equiv i + 1 \mod c\)), we set:

- \(\tilde{x}_e := \tilde{x}_e + \varepsilon\) and \(\hat{x}_e := \tilde{x}_e - \varepsilon\) if \(i(u_i) > i(u_{i+1})\);
- \(\tilde{x}_e := \tilde{x}_e - \varepsilon\) and \(\hat{x}_e := \tilde{x}_e + \varepsilon\) otherwise.

Note that, for \(\varepsilon\) sufficiently small, both \((\tilde{x}, \tilde{z})\) and \((\hat{x}, \hat{z})\) are feasible solutions of (3). In fact, consider one of the nodes, say \(v\), in the cycle \(C\). Let \(e', e''\) be the two edges in \(C\) with \(v\) as endpoint. If \(\tilde{x}_e' - \tilde{x}_e'' = \tilde{x}_e'' - \tilde{x}_e', \) by definition, one edge connects \(v\) to a node in \(B(v)\) and the other to a node in \(F(v)\). Hence, \(\phi_v(\tilde{x}, \tilde{z}) = \phi_v(\hat{x}, \hat{z}) + \varepsilon = k_v\). Otherwise, if \(\tilde{x}_e' - \tilde{x}_e'' \neq \tilde{x}_e'' - \tilde{x}_e'\), \(e', e''\) connect \(v\) to nodes both in \(B(v)\) or both in \(F(v)\). However, in both cases it follows that \(\phi_v(\tilde{x}, \tilde{z}) = k_v\). Analogous considerations hold for \((\hat{x}, \hat{z})\) too. Furthermore, observe that \((\tilde{x}, \tilde{z}) = \frac{1}{2}(\hat{x}, \hat{z}) + \frac{1}{2}(\tilde{x}, \tilde{z})\). Absurd.

So \(G_{\text{frac}}\) is a forest. Consider two arbitrary leaves \(v^*, w^*\) both belonging to a connected component of \(G_{\text{frac}}\). Let \(P_{v^*, w^*}\) be an arbitrary path in \(G_{\text{frac}}\) connecting \(v^*\) and \(w^*\). Going from \(v^*\) to \(w^*\), label with \(u_0 = v^*, u_1, \ldots, u_p = w^*\) the nodes in \(P_{v^*, w^*}\). Since \(v^*\) and \(w^*\) are leaves of \(G_{\text{frac}}\), and since \(\phi_{v^*}(\tilde{x}, \tilde{z}) = k_{v^*}\) and \(\phi_{w^*}(\hat{x}, \hat{z}) = k_{w^*}\), there must exist at least one variable for each node, say \(\tilde{z}_{v^*}^1\) and \(\tilde{z}_{w^*}^1\), such that \(0 < \tilde{z}_{v^*}^1 < 1\) and \(0 < \tilde{z}_{w^*}^1 < 1\). We define two new vectors \((\tilde{x}, \tilde{z})\) and \((\hat{x}, \hat{z})\) as follow: they are the exact copy of vector \((\tilde{x}, \tilde{z})\) except for the variables associated to the edges in \(P_{v^*, w^*}\), and variables \(\tilde{z}_{v^*}^2, \tilde{z}_{v^*}^3, \tilde{z}_{w^*}^2, \tilde{z}_{w^*}^3\). Let \(\varepsilon \in \mathbb{R}_0^+\). For each edge \(e = u_i u_{i+1}\) in \(P_{v^*, w^*}\), we set:

- \(\tilde{x}_e := \tilde{x}_e + \varepsilon\) and \(\hat{x}_e := \tilde{x}_e - \varepsilon\) if \(i(u_i) > i(u_{i+1})\);
- \(\tilde{x}_e := \tilde{x}_e - \varepsilon\) and \(\hat{x}_e := \tilde{x}_e + \varepsilon\) otherwise.

Furthermore, we set:

\[
\begin{align*}
\tilde{z}_v^{w^*} &:= \tilde{z}_v - \varepsilon & \text{if } i(v^*) > i(u_1); \\
\hat{z}_v^{w^*} &:= \hat{z}_v + \varepsilon & \text{otherwise};
\end{align*}
\]

and

\[
\begin{align*}
\tilde{z}_v^{v^*} &:= \tilde{z}_v + \varepsilon & \text{if } i(v^*) > i(u_{p-1}); \\
\hat{z}_v^{v^*} &:= \hat{z}_v - \varepsilon & \text{otherwise}.
\end{align*}
\]

With analogous considerations like those applied proving that \(G_{\text{frac}}\) is acyclic, it can be shown that, for \(\varepsilon\) sufficiently small, both \((\tilde{x}, \tilde{z})\) and \((\hat{x}, \hat{z})\) are feasible solutions of (3). Furthermore, observe that \((\tilde{x}, \tilde{z}) = \frac{1}{2}(\hat{x}, \hat{z}) + \frac{1}{2}(\tilde{x}, \tilde{z})\). Absurd.

Hence, no vertex of the polytope \(P\) can be fractional. In other words, \(P\) is integral. \(\square\)

The main consequence of Theorem 7.2 is that Problem 6.1 can also be solved directly using linear programming, without reducing it to the weighted \(b\)-factor problem. Indeed, a compact linear programming formulation of Problem 6.1 is the following:

\[
\begin{align*}
\min \left\{ \sum_{e \in E} w_e x_e + \sum_{v \in V} \sum_{i=1}^{k_v} w_{ij} z_i^v : (x, z) \in P \right\},
\end{align*}
\]

where \(w_{ij}^v\) is defined exactly as \(w_{ij}^v\) in Section 6, and \(P\) is the polytope defined by (3).

8. Conclusions

In this paper, we first considered the problem of covering the edges of partially directed graphs (i.e. graphs which contain both directed and undirected edges) with a minimum number of edge-disjoint directed paths, respecting the orientations of the directed edges. We showed that this problem is polynomially solvable presenting an \(O(|A| + |E|(|V| + |E|))\) worst-case time complexity algorithm, where \(V\) is the set of nodes, \(A\) is the set of directed edges and \(E\) is the set of undirected edges of the given graph. We next provided a good-characterization of this problem through a min–max theorem.
Then we considered a more general weighted formulation of this problem including weights on the two possible orientations of the undirected edges (MWEOP). We showed that the MWEOP is equivalent to the weighted bipartite $b$-factor problem. In details, we proposed a linear time reduction from the MWEOP to the weighted bipartite $b$-factor problem. Furthermore, once a solution for the latter problem is obtained, a solution for the former problem can be computed in linear time. We recall that the minimum weight $b$-factor problem in bipartite graph can be solved in strongly polynomial-time as a minimum cost flow problem [16] or as a transportation problem [10]; we redirect the reader to [15, pp. 355–356] for an up-to-date detailed complexity survey on weighted bipartite $b$-matchings.

To conclude, we also provided a compact linear programming formulation for the MWEOP: hence a solution for the MWEOP can be obtained directly using an LP solver, without applying the $b$-factor reduction.

Acknowledgments

Thanks are due to an anonymous referee for some advice on the general organization of the paper.

References