ON THE INTEGRALITY OF SOME FACILITY LOCATION POLYTOPES

MOURAD BAÏOU† AND FRANCISCO BARAHONA‡

Abstract. We study a system of linear inequalities associated with some facility location problems. We show that this system defines a polytope with integer extreme points if and only if the graph does not contain a certain type of odd cycles. We also derive odd cycle inequalities and give a separation algorithm.

Key words. facility location, odd cycle inequalities

AMS subject classifications. 05C85, 90C27

DOI. 10.1137/070706070

1. Introduction. Let \( G = (V, A) \) be a directed graph, not necessarily connected, where each arc and each node has weight associated with it. We study a “prize collecting” version of a location problem (LP) as follows. A set of nodes is selected, usually called centers, and then each nonselected node can be assigned to a center. The weight of a node is the revenue obtained by opening a facility at that location, minus the cost of building the facility. The weight of an arc \((i, j)\) is the revenue obtained by assigning the location \(i\) to the location \(j\), minus the cost originated by this assignment. The goal is to maximize the sum of the weights of the selected nodes plus the sum of the weights yielded by the assignment. The linear system below defines a linear programming relaxation:

\[
\begin{align*}
\max & \sum_{(u,v) \in A} w(u,v)x(u,v) + \sum_v w(v)y(v) \\
\text{s.t.} & \quad \sum_{(u,v) \in A} x(u,v) + y(u) \leq 1 \quad \forall u \in V, \\
& \quad x(u,v) \leq y(v) \quad \forall (u,v) \in A, \\
& \quad 0 \leq y(v) \leq 1 \quad \forall v \in V, \\
& \quad x(u,v) \geq 0 \quad \forall (u,v) \in A.
\end{align*}
\]

For each node \(u\), the variable \(y(u)\) takes the value 1 if the node \(u\) is selected and 0 otherwise. For each arc \((u,v)\) the variable \(x(u,v)\) takes the value 1 if \(u\) is assigned to \(v\) and 0 otherwise. Inequalities (1) express the fact that either node \(u\) can be selected or it can be assigned to another node. Inequalities (2) indicate that if a node \(u\) is assigned to a node \(v\), then this last node should be selected. The set of integer vectors that satisfy (1)–(4) corresponds to a transitive packing as defined in [15].

Let \(P(G)\) be the polytope defined by (1)–(4), and let \(LP(G)\) be the convex hull of \(P(G) \cap \{0,1\}^{V+|A|}\). Clearly

\[LP(G) \subseteq P(G).\]
In this paper we characterize the graphs $G$ for which $LP(G) = P(G)$. More precisely, we show that $LP(G) = P(G)$ if and only if $G$ does not contain certain types of “odd” cycles. We also give a polynomial algorithm to recognize the graphs in this class.

The uncapacitated facility location problem (UFLP) is a variation where $V$ is partitioned into $V_1$ and $V_2$. The set $V_1$ corresponds to the customers, and the set $V_2$ corresponds to the potential facilities. Each customer in $V_1$ should be assigned to an opened facility in $V_2$. This is obtained by considering $A \subseteq V_1 \times V_2$, fixing to zero the variables $y$ for the nodes in $V_1$, and setting into equations all the inequalities (1) for the nodes in $V_1$. More precisely, the linear programming relaxation for this case is

$$\min \sum c(u,v)x(u,v) + \sum d(v)y(v)$$

$$\sum_{(u,v) \in A} x(u,v) = 1 \ \forall u \in V_1,$$  

(5)

$$x(u,v) \leq y(v) \ \forall (u,v) \in A,$$  

(6)

$$0 \leq y(v) \leq 1 \ \forall v \in V_2,$$  

(7)

$$x(u,v) \geq 0 \ \forall (u,v) \in A.$$  

(8)

Here we also characterize the cases for which (5)–(8) define an integral polytope.

The facets of the uncapacitated facility location polytope have been studied in [13], [11], [5], [6], [12], and others. Other references on this problem are [10] and [14]. The relationship between location polytopes and the stable set polytope has been studied in [16], [7], [17], [2], [18], and others.

For a directed graph $G = (V,A)$ and a set $W \subset V$, we denote by $\delta^+(W)$ the set of arcs $(u,v) \in A$, with $u \in W$ and $v \in V \setminus W$. Also, we denote by $\delta^-(W)$ the set of arcs $(u,v)$, with $v \in W$ and $u \in V \setminus W$. We write $\delta^+(v)$ and $\delta^-(v)$ instead of $\delta^+({v})$ and $\delta^-(\{v\})$, respectively. If there is a risk of confusion, we use $\delta_G^+$ and $\delta_G^-$. A node $u$ with $\delta^+(u) = 0$ is called a pendent node.

A simple cycle $C$ is an ordered sequence

$$v_0, a_0, v_1, a_1, \ldots, a_{p-1}, v_p,$$

where

- $v_i$, $0 \leq i \leq p - 1$, are distinct nodes,
- $a_i$, $0 \leq i \leq p - 1$, are distinct arcs,
- either $v_i$ is the tail of $a_i$ and $v_{i+1}$ is the head of $a_i$, or $v_i$ is the head of $a_i$ and $v_{i+1}$ is the tail of $a_i$ for $0 \leq i \leq p - 1$, and
- $v_0 = v_p$.

By setting $a_p = a_0$, we associate with $C$ three more sets as below.

- We denote by $\tilde{C}$ the set of nodes $v_i$, such that $v_i$ is the head of $a_{i-1}$ and also the head of $a_i$, $1 \leq i \leq p$.
- We denote by $\tilde{C}$ the set of nodes $v_i$, such that $v_i$ is the tail of $a_{i-1}$ and also the tail of $a_i$, $1 \leq i \leq p$.
- We denote by $\tilde{C}$ the set of nodes $v_i$, such that either $v_i$ is the head of $a_{i-1}$ and also the tail of $a_i$, or $v_i$ is the tail of $a_{i-1}$ and also the head of $a_i$, $1 \leq i \leq p$.

Notice that $|\tilde{C}| = |\tilde{C}|$. A cycle will be called odd if $p + |\tilde{C}|$ (or $|\tilde{C}| + |\tilde{C}|$) is odd; otherwise it will be called even. A cycle $C$ with $\tilde{C} = 0$ is a directed cycle. The set of arcs in $C$ is denoted by $A(C)$. We plan to prove that $LP(G) = P(G)$ if and only if $G$ has no odd cycle.
If we do not require \( v_0 = v_p \), we have a path \( P \). In a similar way we define \( \hat{P} \) and \( \check{P} \), excluding \( v_0 \) and \( v_p \). We say that \( P \) is odd if \( p + |\hat{P}| \) is odd; otherwise it is even. For the path \( P' \), the nodes \( v_1, \ldots, v_{p-1} \) are called internal.

If \( G \) is a connected graph and there is a node \( u \) such that its removal disconnects \( G \), we say that \( u \) is an articulation point. A graph is said to be two-connected if at least two nodes should be removed to disconnect it. For simplicity, sometimes we use \( z \) to denote the vector \((x, y)\), i.e., \( z(u) = y(u) \) and \( z(u, v) = x(u, v) \). Also for \( S \subseteq V \cup A \) we use \( z(S) \) to denote \( z(S) = \sum_{a \in S} z(a) \).

A polyhedron \( P \) is defined by a set of linear inequalities, i.e., \( P = \{x \mid Ax \leq b\} \). A face of \( P \) is obtained by setting into equations some of these inequalities. An extreme point of \( P \) is given by a face that contains a unique element. In other words, some inequalities are set to equations so that this system has a unique solution. A polyhedron whose extreme points are integer is called an integral polyhedron.

This paper is organized as follows. In section 2 we give a decomposition theorem that shows that one has to concentrate on two-connected graphs. In section 3 we describe some transformations of the graph that are needed in the following section. Section 4 is devoted to two-connected graphs. In section 5 we study graphs with odd cycles. The separation problem for the so-called odd cycle inequalities is studied in section 6. In section 7 we show how to test the existence of an odd cycle. Section 8 is devoted to the bipartite case.

2. Decomposition. In this section we consider a graph \( G = (V, A) \) that decomposes into two graphs \( G_1 = (V_1, A_1) \) and \( G_2 = (V_2, A_2) \), with \( V = V_1 \cup V_2 \), \( V_1 \cap V_2 = \{u\} \), \( A = A_1 \cup A_2 \), \( A_1 \cap A_2 = \emptyset \). We define \( G'_1 \), which is obtained from \( G_1 \) after replacing \( u \) by \( u' \). We also define \( G'_2 \), which is obtained from \( G_2 \) after replacing \( u \) by \( u'' \). See Figure 1. The theorem below shows that we have to concentrate on two-connected graphs.

![Fig. 1.](image)

**Theorem 1.** Suppose that the system

\[
(9) \quad Az' \leq b,
\]

\[
(10) \quad z'(\delta_{G_1}'(u')) + z'(u') \leq 1
\]
describes $LP(G'_1)$. Suppose that (9) contains the inequalities (1)–(4) except for (10). Similarly, suppose that

\[(11) \quad Cz'' \leq d,\]

\[(12) \quad z''(\delta_{G'_2}^+(u'')) + z''(u'') \leq 1\]

describes $LP(G'_2)$. Also (11) contains the inequalities (1)–(4) except for (12). Then the system below describes an integral polyhedron:

\[(13) \quad A\vec{z}' \leq b,\]

\[(14) \quad Cz'' \leq d,\]

\[(15) \quad z'(\delta_{G'_1}^+(u')) + z''(\delta_{G'_2}^+(u'')) + z'(u') \leq 1,\]

\[(16) \quad z'(u') = z''(u'').\]

**Proof.** Let $(\vec{z}', \vec{z}'')$ be an extreme point of the polytope defined by the above system. We study two cases.

**Case 1.** $\vec{z}'(u') = 0$. We have that $\vec{z}' \in LP(G'_1)$ and $\vec{z}'' \in LP(G'_2)$. If $\vec{z}'$ is an extreme point of $LP(G'_1)$, we have to consider two subcases:

- $\vec{z}'(\delta_{G'_1}^+(u')) = 0$. If $\vec{z}''$ is not an extreme point of $LP(G'_2)$, $\vec{z}'' = 1/2\lambda_1 + 1/2\lambda_2$, with $\lambda_1, \lambda_2$ in $LP(G'_2)$, $\lambda_1 \neq \lambda_2$. Since $\lambda_1(\delta_{G'_2}^+(u'')) \leq 1$, $\lambda_2(\delta_{G'_2}^+(u'')) \leq 1$, we have that $(\vec{z}', \vec{z}'') = 1/2(\vec{z}', \lambda_1) + 1/2(\vec{z}', \lambda_2)$, with $(\vec{z}', \lambda_1)$ and $(\vec{z}', \lambda_2)$ satisfying (13)–(16), which is a contradiction. Thus $\vec{z}''$ is an extreme point and $(\vec{z}', \vec{z}'')$ is an integral vector.

- $\vec{z}'(\delta_{G'_1}^+(u')) = 1$. This implies that $\vec{z}''(\delta_{G'_2}^+(u'')) = 0$. If $\vec{z}''$ is not an extreme point, then $\vec{z}'' = 1/2\lambda_1 + 1/2\lambda_2$, with $\lambda_1, \lambda_2$ in $LP(G'_2)$, $\lambda_1 \neq \lambda_2$. Since $\lambda_1(\delta_{G'_2}^+(u'')) = 0 = \lambda_2(\delta_{G'_2}^+(u''))$, we have that $(\vec{z}', \vec{z}'') = 1/2(\vec{z}', \lambda_1) + 1/2(\vec{z}', \lambda_2)$, with $(\vec{z}', \lambda_1)$ and $(\vec{z}', \lambda_2)$ satisfying (13)–(16), which is a contradiction. Thus $\vec{z}''$ is an extreme point and $(\vec{z}', \vec{z}'')$ is an integral vector.

Now we should study the situation in which $\vec{z}'$ and $\vec{z}''$ are not extreme points.

We should have $\vec{z}' = 1/2\omega_1 + 1/2\omega_2$, with $\omega_1, \omega_2$ in $LP(G'_1)$, $\omega_1 \neq \omega_2$. If $\omega_1(\delta_{G'_1}^+(u')) = \omega_2(\delta_{G'_1}^+(u')) = \vec{z}'(\delta_{G'_1}^+(u'))$, we have $(\vec{z}', \vec{z}'') = 1/2(\omega_1, \vec{z}'') + 1/2(\omega_2, \vec{z}'')$, with $(\omega_1, \vec{z}'')$ and $(\omega_2, \vec{z}'')$ satisfying (13)–(16), which is a contradiction.

Now we assume that

\[
\omega_1\left(\delta_{G'_1}^+(u')\right) = \vec{z}'\left(\delta_{G'_1}^+(u')\right) - \epsilon,
\]

\[
\omega_2\left(\delta_{G'_1}^+(u')\right) = \vec{z}'\left(\delta_{G'_1}^+(u')\right) + \epsilon,
\]

with $\epsilon > 0$.

We also have $\vec{z}'' = 1/2\lambda_1 + 1/2\lambda_2$, with $\lambda_1, \lambda_2$ in $LP(G'_2)$, $\lambda_1 \neq \lambda_2$. If $\lambda_1(\delta_{G'_2}^+(u'')) = \lambda_2(\delta_{G'_2}^+(u'')) = \vec{z}''(\delta_{G'_2}^+(u''))$, we obtain a contradiction as above. Thus we suppose that

\[
\lambda_1\left(\delta_{G'_2}^+(u'')\right) = \vec{z}''\left(\delta_{G'_2}^+(u'')\right) + \rho,
\]

\[
\lambda_2\left(\delta_{G'_2}^+(u'')\right) = \vec{z}''\left(\delta_{G'_2}^+(u'')\right) - \rho,
\]

with $\rho > 0$. 

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
We can assume that \( \epsilon = \rho \); otherwise we can change \( \lambda_1 \) and \( \lambda_2 \). Thus we have \((z', z'') = 1/2(\omega_1, \lambda_1) + 1/2(\omega_2, \lambda_2)\), with \((\omega_1, \lambda_1) \) and \((\omega_2, \lambda_2)\) satisfying (13)–(16), which is a contradiction.

Case 2. \( 0 < \bar{z}'(u') \). We have that \( \bar{z}' \in LP(G'1) \) and \( \bar{z}'' \in LP(G'2) \). Thus \( \bar{z}' \) is a convex combination of extreme points \( \mu_i \) of \( LP(G'1) \) that satisfy with equality every constraint that is satisfied with equality by \( \bar{z}' \). Also \( \bar{z}'' \) is a convex combination of extreme points \( \phi_j \) of \( LP(G'2) \) that satisfy with equality every constraint satisfied with equality by \( \bar{z}'' \).

We can assume that \( \mu_1(u') = 1 = \phi_1(u'') \). After putting together these two vectors we obtain a 0-1 vector that satisfies with equality every constraint that is satisfied with equality by the original vector \((\bar{z}', \bar{z}'')\), which is a contradiction.

We have the following corollary.

**Corollary 2.** The polytope \( LP(G) \) is defined by the system (13)–(16) after identifying the variables \( z'(u') \) and \( z''(u'') \).

This last corollary shows that if \( LP(G'1) \) and \( LP(G'2) \) are defined by (1)–(4), then \( LP(G) \) is also defined by (1)–(4). Thus we have to concentrate on graphs that are two-connected. A result analogous to Theorem 1, for the stable set polytope, has been given in [8].

### 3. Graph transformations

First we plan to prove that if \( G \) has no odd cycle, then \( LP(G) = P(G) \). The proof consists of assuming that \( \bar{z} \) is a fractional extreme point of \( P(G) \) and arriving at a contradiction. Below we give several assumptions that can be made about \( \bar{z} \) and \( G \); they will be used in the next section.

**Lemma 1.** We can assume that \( \bar{z}(u, v) > 0 \) for all \((u, v) \in A \).

**Proof.** Let \( G' \) be the graph obtained after removing all arcs \((u, v) \) with \( \bar{z}(u, v) = 0 \), and let \( z' \) be the vector obtained after removing all components \( \bar{z}(u, v) = 0 \). Then \( z' \) is a fractional extreme point of \( P(G') \).

**Lemma 2.** If \( 0 < \bar{z}(u, v) < \bar{z}(v) \), we can assume that \( v \) is a pendent node with \( |\delta^-(v)| = 1 \) and \( \bar{z}(v) = 1 \).

**Proof.** If \( v \) is not pendent or \( |\delta^-(v)| > 1 \), we can remove \((u, v) \) and add a new node \( v' \) and the arc \((u, v') \). Then we can define \( z'(u, v') = \bar{z}(u, v) \), \( z'(v') = 1 \), and \( z'(s, t) = \bar{z}(s, t) \), \( z'(r) = z(r) \) for all other nodes and arcs. Let \( G' \) be this new graph. We have that the constraints that are tight for \( \bar{z} \) are also tight for \( z' \), so \( z' \) is an extreme point of \( P(G') \).

**Lemma 3.** We can assume that \( G \) consists of only one connected component.

**Proof.** Let \( G_1 \) be a connected component of \( G \). Let \( z_1 \) be the projection of \( \bar{z} \) onto the space associated with \( G_1 \). Then \( z_1 \) is an extreme point of \( P(G_1) \).

**Lemma 4.** We can assume that \( 0 < \bar{z}(u, v) < 1 \) for all \((u, v) \in A \).

**Proof.** If \( \bar{z}(u, v) = 1 \), it follows from Lemma 1 that \( \delta^-(u) = \emptyset \) and \( \delta^+(u) = \{(u, v)\} \). Since \( \bar{z}(v) = 1 \), Lemma 1 implies that \( v \) is pendent. It follows from Lemma 2 that \( \bar{z}(r, v) = 1 \) for all \((r, v) \in \delta^-(v) \). Therefore, the graph induced by \( \delta^-(v) \) is a connected component of \( G \). All variables associated with this connected component take integer values.

**Lemma 5.** We can assume that either \( G \) is two-connected or it consists of a single arc.

**Proof.** If \( G \) has an articulation point, we can apply Theorem 1 to decompose \( G \) into \( G_1 \) and \( G_2 \). If inequalities (1)–(4) define \( LP(G_1) \) and \( LP(G_2) \), then a similar system should define \( LP(G) \). One can keep decomposing as long as the graph has an articulation point.
If the graph $G$ consists of a single arc, it is fairly easy to see that $LP(G) = P(G)$, so now we have to deal with the two-connected components. This is treated in the next section.

4. **Treating two-connected graphs.** In this section we assume that the graph $G$ is two-connected and it has no odd cycle. Let $\bar{z}$ be a fractional extreme point of $P(G)$; we are going to assign labels $l$ to the nodes and arcs and define $z'(u,v) = \bar{z}(u,v) + l(u,v)\epsilon$, $z'(u) = \bar{z}(u) + l(u)\epsilon$, $\epsilon > 0$, for each arc $(u,v)$ and each node $u$. We shall see that every constraint that is satisfied with equality by $\bar{z}$ is also satisfied with equality by $z'$. This is the required contradiction.

Given a path $P = v_0, a_0, \ldots, a_p, v_p$, assume that the label of $a_0$, $l(a_0)$, has the value $1$ or $-1$. We define the labeling procedure as follows.

For $i = 1$ to $p - 1$ do the following:

- If $v_i$ is the head of $a_{i-1}$ and it is the tail of $a_i$, then $l(v_i) = l(a_{i-1})$, $l(a_i) = -l(v_i)$.
- If $v_i$ is the head of $a_{i-1}$ and it is the tail of $a_i$, then $l(v_i) = l(a_{i-1})$, $l(a_i) = l(v_i)$.
- If $v_i$ is the tail of $a_{i-1}$ and it is the head of $a_i$, then $l(v_i) = -l(a_{i-1})$, $l(a_i) = l(v_i)$.
- If $v_i$ is the tail of $a_{i-1}$ and it is the tail of $a_i$, then $l(v_i) = 0$, $l(a_i) = -l(a_{i-1})$.

Notice that the labels of $v_0$ and $v_p$ were not defined.

This procedure will be used in four different cases as below.

**Case 1.** $G$ contains a directed cycle $C = v_0, a_0, \ldots, a_p, v_p$. Assume that the head of $a_0$ is $v_1$, set $l(v_0) = -1$ and $l(a_0) = 1$, and extend the labels as above.

**Case 2.** $G$ contains a cycle $C = v_0, a_0, \ldots, a_p, v_p$ and $C \neq \emptyset$. Assume $v_0 \in C$.

Set $l(v_0) = 0$ and $l(a_0) = 1$, and extend the labels.

The lemma below is needed to show that for $v_0$, the constraints that were satisfied with equality by $\bar{z}$ remain satisfied with equality.

**Lemma 6.** After labeling as in Cases 1 and 2, we have $l(a_{p-1}) = -l(a_0)$.

**Proof.** Case 1 should be clear, so we have to study Case 2. Let $v_{j(0)}, v_{j(1)}, \ldots, v_{j(k)}$ be the ordered sequence of nodes in $C$, with $v_{j(0)} = v_{j(k)}$. A path in $C$

$$v_{j(i)}, a_{j(i)}, \ldots, a_{j(i+1)-1}, v_{j(i+1)}$$

from $v_{j(i)}$ to $v_{j(i+1)}$ will be called a *segment* and denoted by $S_i$. A segment is odd (resp., even) if it contains an odd (resp., even) number of arcs. Let $n_e$ be the number of even segments and $n_o$ the number of odd segments. We have that $n_e + n_o = |C|$. We also have that the parity of $p$ is equal to the parity of $n_o$. Therefore, $n_o + |C|$ should be even.

The labeling has the following properties:

(a) If the segment is odd, then $l(a_{j(i)}) = -l(a_{j(i+1)-1})$.

(b) If the segment is even, then $l(a_{j(i)}) = l(a_{j(i+1)-1})$.

Now we build an undirected cycle as follows. For every node $v_{j(i)}$ we have two nodes $u_{j}^{1}$ and $u_{j}^{2}$; we add an edge between them marked “blue.” For every segment from $v_{j(i)}$ to $v_{j(i+1)}$ we have an edge from $u_{j}^{2}$ to $u_{j+1}^{1}$. If the segment is odd, we mark the edge “blue”; otherwise we mark it “green.” Start by giving the label $l(u_{0}^{2}) = 1$ to $u_{0}^{2}$. Continue labeling so that if $st$ is a blue edge, then $l(t) = -l(s)$, and if the edge is green, then $l(t) = l(s)$. The label of $u_{j}^{2}$ corresponds to the label of $a_{j(i)}$, and the label of $u_{j+1}^{1}$ corresponds to the label of $a_{j(i+1)-1}$. There is an even number of blue edges in the cycle; therefore, $l(u_{0}^{1}) = -l(u_{0}^{2})$. Thus

$$l(a_{p-1}) = -l(a_0).$$
Notice that after the first cycle has been labeled as in Cases 1 or 2, the properties below hold. We shall see that these properties hold throughout the entire labeling procedure.

Property 1. If a node has a nonzero label, then it is the tail of at most one labeled arc.

Property 2. If a node has a zero label, then it is the tail of exactly two arcs with opposite labels, and it is not the head of any labeled arc.

The lemma below shows a property of the labeling procedure that will be used in the analysis of the next case.

**Lemma 7.** Let $P = v_0, a_0, v_1, a_1, \ldots, a_p, v_p$ be a path. Suppose that we set $l(a_0)$ and we extend the labels; then the label of $a_{p-1}$ is determined by

- the orientation of $a_0$,
- the orientation of $a_{p-1}$, and
- the parity of $P$.

**Proof.** Add a node $t$ and the arcs $\tilde{a} = (t, v_0)$ and $\tilde{a} = (t, v_p)$ to create a cycle. If the cycle is odd, subdivide $\tilde{a}$ to make it even. Set $l(t) = 0$ and $l(\tilde{a}) = 1$, and extend the labels as in Case 2. It follows from Lemma 6 that the label of the arc before $\tilde{a}$ is $-l(\tilde{a})$; this determines the label of the previous arc, and so on.

Once a cycle $C$ has been labeled as in Cases 1 or 2, we have to extend the labeling as follows.

**Case 3.** Suppose that $l(v_0) \neq 0$ for $v_0 \in C$ ($v_0$ is the head of a labeled arc) and there is a path $P = v_0, a_0, v_1, a_1, \ldots, a_p, v_p$ in $G$ such that

- $v_0$ is the head of $a_0$,
- $v_p \in C$, and
- $\{v_1, \ldots, v_{p-1}\}$ is disjoint from $C$.

We set $l(a_0) = l(v_0)$ and extend the labels. Case 3 is needed so that any inequality (2) associated with $v_0$ that is satisfied with equality remains satisfied with equality.

We have to see that the label $l(a_{p-1})$ is such that constraints associated with $v_p$, that were satisfied with equality, remain satisfied with equality. This is discussed in the next lemma.

**Lemma 8.** If $v_p$ is the head of $a_{p-1}$, then $l(a_{p-1}) = l(v_p)$. If $v_p$ is the tail of $a_{p-1}$, then $l(a_{p-1}) = -l(v_p)$.

**Proof.** Notice that $v_0 \notin C$. In Figure 2 we represent the possible configurations for the paths in $C$ between $v_0$ and $v_p$. In this figure we show whether $v_0$ and $v_p$ are the head or the tail of the arcs in $C$ incident to them. These two paths are denoted by $P_1$ and $P_2$. Lemma 7 shows that we have to pay attention to their parity and to the orientation of the first and last arcs.

![Figure 2](image-url)

**Fig. 2.** Possible paths in $C$ between $v_0$ and $v_p$. It is shown whether $v_0$ and $v_p$ are the head or the tail of the arcs in $C$ incident to them.
Consider configuration (1): these two paths should have different parity. When adding the path $P$, an odd cycle is created with either $P_1$ or $P_2$. So configuration (1) will not occur. The same happens with configuration (2).

Now we discuss configuration (3). These two paths should have the same parity. If $v_p$ is the tail of $a_{p-1}$, then $P$ creates an odd cycle with either $P_1$ or $P_2$. If $v_p$ is the head of $a_{p-1}$, then $P$ should have the same parity as $P_1$ and $P_2$. Then $l(a_{p-1}) = l(v_p)$.

The study of configuration (4) is similar. The two paths should have the same parity. If $v_p$ is the tail of $a_{p-1}$, then $P$ creates an odd cycle with either $P_1$ or $P_2$. If $v_p$ is the head of $a_{p-1}$, then $P$ should have the same parity as $P_1$ and $P_2$, and $l(a_{p-1}) = l(v_p)$.

For configuration (5), again the two paths should have the same parity. If $v_p$ is the head of $a_{p-1}$, then $P$ should have the same parity as $P_1$ and $P_2$, and $l(a_{p-1}) = l(v_p)$. If $v_p$ is the tail of $a_{p-1}$, then $P$ should have the same parity as $P_1$ and $P_2$, and $l(a_{p-1}) = -l(v_p)$.

Also, in configuration (6) the paths $P_1$ and $P_2$ should have the same parity. If $v_p$ is the tail of $a_{p-1}$, then $P$ should have the same parity as $P_1$ and $P_2$, and $l(a_{p-1}) = l(v_p)$. If $v_p$ is the head of $a_{p-1}$, then $P$ should have the same parity as $P_1$ and $P_2$, and $l(a_{p-1}) = -l(v_p)$.

Based on this the labels are extended recursively. Denote by $G_l$ the subgraph defined by the labeled arcs. This is a two-connected graph, so for any two nodes $v_0$ and $v_p$, it contains a cycle going through these two nodes. Thus we can check if Case 3 applies and extend the labels adding a path to the graph $G_l$ each time. The two lemmas below show that Properties 1 and 2 remain satisfied.

**Lemma 9.** Suppose that $v_p$ has a label different from 0. If $v_p$ is the tail of an arc in $G_l$, then in Case 3 it cannot be the tail of $a_{p-1}$. Thus Property 1 remains satisfied.

**Proof.** There is a cycle $C$ in $G_l$ containing $v_0$ and $v_p$. Property 1 implies that $v_0$ is the head of at least one arc in $C$. We can assume that $v_p$ is the tail of an arc in $C$. Suppose not; let $a$ be an arc in $G_l$ whose tail is $v_p$. Let $u$ be the head of $a$. Since $G_l$ is two-connected, there is a path $Q$ from $u$ to a node $v$ in $C$ with $v \neq v_p$. The path $Q$ intersects $C$ only at the node $v$. We can add $a$ to $Q$ to $C$ and remove the path in $C$ from $v_p$ to $v$ that does not contain $v_0$ as an internal node.

The cycle $C$ can contain configurations (3), (4), and (6) of Figure 2. In these three cases, the head of $a_{p-1}$ is $v_p$.

**Lemma 10.** Let $w$ be a node in $G_l$ with $l(w) = 0$; then in Case 3 we have that $v_p \neq w$. Therefore, Property 2 remains satisfied.

**Proof.** Let $a_1, a_2$ be the two arcs in $G_l$ having $w$ as their tail. If $v_p = w$, the cycle $C$ in Case 3 must contain both arcs $a_1$ and $a_2$. But configurations (1) and (2) cannot occur.

Once Case 3 has been exhausted we might have some nodes in $G_l$ that are only the heads of labeled arcs. For such nodes we have to ensure that inequalities (1) that were satisfied as equalities remain satisfied as equalities. This is treated as follows.

**Case 4.** Suppose that $v_0$ is only the head of labeled arcs, and $v_0$ is not pendent. Then there is a cycle $C$ in $G_l$ and there is a path $P = v_0, a_0, v_1, a_1, \ldots, a_{p-1}, v_p$ in $G$ such that

- $v_0 \in C$ is the tail of $a_0$,
- $v_p \in C$, and
- $\{v_1, \ldots, v_{p-1}\}$ is disjoint from $G_l$. 

---

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
We set \( l(a_0) = -l(v_0) \) and extend the labels. We have to see that the label \( l(a_{p-1}) \)
is such that constraints associated with \( v_p \), that were satisfied with equality, remain satisfied with equality. This is discussed below.

**Lemma 11.** In Case 4 we have that \( v_p \) is the tail of \( a_{p-1} \) and \( l(a_{p-1}) = -l(v_p) \). Also Properties 1 and 2 continue to hold.

**Proof.** The cycle \( C \) can correspond to configurations (1), (3), or (5) of Figure 2.

For configuration (1), the paths \( P_1 \) and \( P_2 \) have different parities, therefore adding the path \( P \) would create an odd cycle.

Consider now configuration (3). The paths \( P_1 \) and \( P_2 \) have the same parity. If \( v_p \) is the tail of \( a_{p-1} \), then adding \( P \) to \( C \) would create an odd cycle. If \( v_p \) is the head of \( a_{p-1} \), we would have a situation treated in Case 3 and configuration (7).

Finally consider configuration (5). If \( v_p \) is the head of \( a_{p-1} \), we have a situation treated in Case 3 and configuration (5). If \( v_p \) is the tail of \( a_{p-1} \), then \( P \) should have the same parity as \( P_1 \) and \( P_2 \); thus \( l(a_{p-1}) = -l(v_p) \). If \( v_p \) were the tail of an arc in \( G_t \), we would have a cycle like in configuration (3). Adding \( P \) to this cycle would create an odd cycle. Therefore, \( v_p \) was not the tail of an arc in \( G_t \) and Properties 1 and 2 continue to hold.

To summarize, the labeling algorithm consists of the following steps.

- Step 1. Identify a cycle \( C \) in \( G \) and treat it as in Cases 1 or 2. Set \( G_1 = C \).
- Step 2. For as long as needed, label as in Case 3. Each time add to \( G_t \) the new set of labeled nodes and arcs.
- Step 3. If needed, label as in Case 4. Each time add to \( G_t \) the new set of labeled nodes and arcs. If some new labels have been assigned in this step, go to Step 2; otherwise stop.

At this point we can discuss the properties of the labeling procedure. The labels are such that any inequality (2) that was satisfied with equality by \( \bar{z} \) is also satisfied with equality by \( z' \). To see that inequalities (1) that were tight remain tight, we use Properties 1 and 2:

- Any node that has a nonzero label is the tail of exactly one labeled arc having the opposite label.
- If \( u \) is a node with \( l(u) = 0 \), then there are exactly two labeled arcs having opposite labels and whose tails are \( u \).

Finally, we give the label “0” to all nodes and arcs that are unlabeled; this completes the definition of \( z' \). Lemma 4 shows that inequalities (4) will not be violated. Since nodes \( v \) with \( \bar{z}(v) = 0 \) receive a zero label, and there are no nodes \( v \) with \( \bar{z}(v) = 1 \), we have that inequalities (3) cannot be violated. Any constraint that is satisfied with equality by \( \bar{z} \) is also satisfied with equality by \( z' \). This contradicts the assumption that \( \bar{z} \) is an extreme point. We can now state the main result of this section.

**Theorem 3.** If the graph \( G \) is two-connected and has no odd cycle, then \( LP(G) = P(G) \).

This implies the following.

**Theorem 4.** If \( G \) is a graph with no odd cycle, then \( LP(G) = P(G) \).

**Theorem 5.** For graphs with no odd cycle, the UFLP is polynomially solvable.

In some cases one might want to fix to zero the variables \( y \) for some set of nodes and also set to equations some of the inequalities (1). This defines a face \( Q(G) \) of \( P(G) \). We have the following corollary that will be used in section 8.

**Corollary 6.** If \( G \) is a graph with no odd cycle, then \( Q(G) \) is an integral polytope.
5. Odd cycles. In this section we study the effect of odd cycles in $P(G)$. Let $C$ be an odd cycle. We can define a fractional vector $(\bar{x}, \bar{y}) \in P(G)$ as follows:

\begin{align}
\bar{y}(u) &= 0 \quad \forall \text{ nodes } u \in \hat{C}, \\
\bar{y}(u) &= 1/2 \quad \forall \text{ nodes } u \in C \setminus \hat{C}, \\
\bar{x}(a) &= 1/2 \quad \text{for } a \in A(C), \\
\bar{y}(v) &= 0 \quad \forall \text{ other nodes } v \notin C, \\
\bar{x}(a) &= 0 \quad \forall \text{ other arcs}.
\end{align}

In Figure 3 we show two examples. The numbers close to the nodes correspond to the $y$ variables, and the numbers close to the arcs correspond to the $x$ variables.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig3.png}
\caption{Fractional vectors associated with odd cycles.}
\end{figure}

Below we show a family of inequalities that separate the vectors defined above from $LP(G)$. We call them odd cycle inequalities.

**Lemma 12.** The following inequalities are valid for $LP(G)$:

\begin{equation}
2 \sum_{a \in A(C)} x(a) - 2 \sum_{v \in \hat{C}} y(v) + \sum_{v \in \hat{C}} x(\delta^+(v) \setminus A(C)) + \sum_{v \in \hat{C}} x(\delta^+(v) \setminus A(C)) \leq |A(C)| - 2|\hat{C}| + |\hat{C}|,
\end{equation}

for every odd cycle $C$.

**Proof.** From inequalities (1)–(4) we obtain
\begin{align}
x(u, v) + x(\delta^+(v)) &\leq 1 \quad \text{for every arc } (u, v) \in C, v \notin \hat{C}, \\
x(u, v) - y(v) &\leq 0 \quad \text{for every arc } (u, v) \in C, v \in \hat{C}, \\
x(\delta^+(v)) &\leq 1 \quad \text{for } v \in \hat{C}.
\end{align}

Their sum gives
\begin{equation}
2 \sum_{a \in A(C)} x(a) - 2 \sum_{v \in \hat{C}} y(v) + \sum_{v \in \hat{C}} x(\delta^+(v) \setminus A(C)) + \sum_{v \in \hat{C}} x(\delta^+(v) \setminus A(C)) \leq |A(C)| - 2|\hat{C}| + |\hat{C}|,
\end{equation}

which implies
\begin{equation}
2 \sum_{a \in A(C)} x(a) - 2 \sum_{v \in \hat{C}} y(v) \leq |\hat{C}| + |\hat{C}|.
\end{equation}

Dividing by 2 and rounding down the right-hand side, we obtain
\begin{equation}
\sum_{a \in A(C)} x(a) - \sum_{v \in \hat{C}} y(v) \leq \frac{|\hat{C}| + |\hat{C}| - 1}{2}.
\end{equation}
Now we can present our main result.

**Theorem 7.** Let \( G \) be a directed graph; then \( LP(G) = P(G) \) if and only if \( G \) does not contain an odd cycle.

**Proof.** If \( G \) contains an odd cycle \( C \), then we can define a vector \( (\bar{x}, \bar{y}) \in P(G) \) as in (17)–(21). We have

\[
\sum_{a \in A(C)} \bar{x}(a) - \sum_{v \in \hat{C}} \bar{y}(v) = |\hat{C}| + |\hat{C}| \over 2.
\]

Lemma 12 shows that \( \bar{z} \notin LP(G) \).

Then the theorem follows from Theorem 4.

**6. Separation of odd cycle inequalities.** Now we study the separation problem: Given a vector \( (\bar{x}, \bar{y}) \in P(G) \), find an odd cycle inequality (22), if there is any, that separates \( (\bar{x}, \bar{y}) \) from \( LP(G) \). These inequalities are \( \{0, 1/2\} \)-Chvátal–Gomory cuts, using the terminology of [4]. A separation algorithm can be obtained from the results of [4]. Here we give an alternative algorithm.

To solve the separation problem we write the inequalities as

\[
2 \sum_{a \in A(C)} x(a) + \sum_{v \in C} (1 - 2y(v)) \leq |A(C)| - 1
\]

or

\[
\sum_{a \in A(C)} (1 - 2x(a)) + \sum_{v \in C} (2y(v) - 1) \geq 1.
\]

In order to reduce this to a shortest path problem, several graph transformations are required.

**6.1. First transformation.** We build an auxiliary undirected graph \( H = (N, F) \). For every arc \( a = (u, v) \in A \) we create the nodes \( (u, a) \) and \( (v, a) \) in \( H \). The first node is called a tail node, and the second is called a head node. The tail node is associated with \( u \), and the head node is associated with \( v \). We also create an edge between these two nodes with the weight \( (1 - 2\bar{x}(u, v)) \) and label this edge blue. See Figure 4.

![Fig. 4. Edge associated with the arc \((u, v)\). It has the label blue and is called old.](image)

Now for every node \( v \in V \) and every pair of nodes in \( H \) associated with \( v \) we create an edge in \( H \) as follows. This type of edge will be called new. Let \( n_1 \) and \( n_2 \) be two nodes in \( H \) associated with \( v \); we distinguish two cases:

- At least one of them is a tail node. In this case we add an edge between them with weight zero and label it black.
- Both \( n_1 \) and \( n_2 \) are head nodes. In this case we add an edge between them with weight \( 2\bar{y}(v) - 1 \) and label this edge blue. See Figure 5.

A cycle in \( H \) consisting of an alternating sequence of old and new edges is called an alternating cycle. The separation problem reduces to finding an alternating cycle in \( H \) with an odd number of blue edges and total weight less than one.
6.2. Second transformation. To find an alternating cycle in \( H \) with an odd number of blue edges, we create a new graph \( H' = (N', F') \) as follows. For every node \( n \in H \) we make two copies \( n' \) and \( n'' \). Let \( n_1n_2 \) be an edge in \( H \); we have two cases:

- If \( n_1n_2 \) is blue, we create the edges \( n'_1n'_2 \) and \( n''_1n''_2 \) with the same weight as \( n_1n_2 \) and the same name (old or new).
- If \( n_1n_2 \) is black, we create the edges \( n'_1n'_2 \) and \( n''_1n''_2 \) with the same weight as \( n_1n_2 \) and the same name (new).

Then for every node \( n \in H \) we find a shortest alternating path \( P \) from \( n' \) to \( n'' \) in \( H' \). The first edge in the path should be new, and the last edge should be old. Suppose that the weight of \( P \) is less than one; then for each node \( p \in H \) such that \( p' \) and \( p'' \) are in \( P \) we identify them. This gives a (nonnecessarily simple) cycle that is alternating, has an odd number of blue edges, and has weight less than one. Notice that the derivation of inequalities (22) does not depend upon the cycle being simple.

Since the edge-weights could be negative, to find a shortest alternating path we have to modify the Bellman–Ford algorithm for shortest paths as follows. Let \( s \) be a source node. Let \( f^k_n(v) \) be the length of a shortest alternating path from \( s \) to \( v \) having at most \( k \) arcs, whose first arc is new and whose last arc is old. Let \( f^0_n(v) \) be the length of a shortest alternating path from \( s \) to \( v \) having at most \( k \) arcs, whose first arc is new and whose last arc is new. These values are computed with the following formulas:

\[
\begin{align*}
  f^k_n(v) &= \min \{ f^{k-1}_n(v), \min \{ f^{k-1}_n(u) + d_{uv} \mid uv \text{ is old} \} \}, \\
  f^0_n(v) &= \min \{ f^{k-1}_n(v), \min \{ f^{k-1}_n(u) + d_{uv} \mid uv \text{ is new} \} \}, \\
  f^0_s(s) &= 0, \quad f^0_n(s) = \infty, \\
  f^0_n(v) &= f^0_n(v) = \infty \quad \text{for } v \neq s.
\end{align*}
\]

This algorithm requires that the graph has no alternating cycle of negative weight; this is shown below.

**Lemma 13.** The edge weights cannot create a cycle of negative weight.

**Proof.** Suppose that

\[
\sum_{a \in A(C)} (1 - 2x(a)) + \sum_{v \in C} (2y(v) - 1) < 0
\]

for some cycle \( C \). This implies

\[
2 \sum_{a \in A(C)} x(a) - 2 \sum_{v \in C} y(v) > |C| - |\hat{C}|,
\]

but when deriving inequalities (22) we had

\[
2 \sum_{a \in A(C)} x(a) - 2 \sum_{v \in C} y(v) \leq |C| - |\hat{C}|. \quad \square
\]
The complexity of this method is as follows.

**Theorem 8.** The separation problem for inequalities (22) can be solved in \(O(|V|^2|A|^2)\) time.

**Proof.** For the graph \(H = (N, F)\), we have \(|N| = 2|A|\) and \(|F| \leq |A| + |A||V|\). For \(H' = (N', F')\), we have \(|N'| = 4|A|\) and \(|F'| \leq 2|A| + 2|A||V|\). For a particular value \(k\), computing the values \(f\) takes \(O(|F'|)\) operations. Since \(k \leq |V|\), applying this algorithm for a particular source \(s\) takes \(O(|V|^2|A|)\) operations. Since every node of \(H\) should be tried as a source, the entire procedure takes \(O(|V|^2|A|^2)\) time. \(\square\)

7. **Detecting odd cycles.** Now we study how to recognize the graphs \(G\) for which \(LP(G) = P(G)\). We start with a graph \(G\), and a new undirected graph \(H = (N, E)\) is built as follows. For every node \(u \in G\) we have the nodes \(u'\) and \(u''\) in \(N\) and the edge \(u'u'' \in E\). For every arc \((u, v) \in G\) we have an edge \(u'v'' \in E\). See Figure 6.

![Fig. 6. Basic transformation to create the graph H.](image)

Considering a cycle \(C\) in \(G\), we build a cycle \(C_H\) in \(H\) as follows:
- If \((u, v)\) and \((u, w)\) are in \(C\), then the edges \(u'v''\) and \(u''w'\) are taken.
- If \((u, v)\) and \((w, v)\) are in \(C\), then the edges \(u'v''\) and \(v''w'\) are taken.
- If \((u, v)\) and \((v, w)\) are in \(C\), then the edges \(u'v'', v''v',\) and \(v'w''\) are taken.

On the other hand, a cycle in \(H\) corresponds to a cycle in \(G\). Thus there is a one to one correspondence among cycles of \(G\) and cycles of \(H\). Moreover, if the cycle in \(H\) has cardinality \(2q\), then \(q = |C| + |\bar{C}|\), where \(C\) is the corresponding cycle in \(G\). Therefore, an odd cycle in \(G\) corresponds to a cycle in \(H\) of cardinality \(2(2p + 1)\) for some positive integer \(p\). See Figure 7.

![Fig. 7. An odd cycle in G and the corresponding cycle in H. The nodes of H close to a node u ∈ G correspond to u’ or u’’.](image)

In other words, finding an odd cycle in \(G\) reduces to finding a cycle of cardinality \(2(2p + 1)\) for some positive integer \(p\) in the bipartite graph \(H\).

For this question, a linear time algorithm was given in [19]. A simple \(O(|V||A|^2)\) has been given in [9]; we describe it below.
First we should find a cycle basis of $H$ and test if the cardinality of every cycle in this basis is $0 \ mod \ 4$. If there is one whose cardinality is $2 \ mod \ 4$, we are done. Otherwise consider the symmetric difference of two cycles whose cardinality is $0 \ mod \ 4$. If the cardinality of their intersection is even, then the cardinality of their symmetric difference is $0 \ mod \ 4$; otherwise it is $2 \ mod \ 4$. Since any cycle $C$ can be obtained as the symmetric difference of a set of cycles in the basis, if the cardinality of $C$ is $2 \ mod \ 4$, then there are at least two cycles in the basis whose symmetric difference has cardinality $2 \ mod \ 4$. Therefore, one just has to test all elements of a cycle basis and the symmetric difference of all pairs.

8. Uncapacitated facility location. Now we assume that $V$ is partitioned into $V_1$ and $V_2$, $A \subseteq V_1 \times V_2$, and we deal with the system

$$\sum_{(u,v) \in A} x(u,v) = 1 \ \forall u \in V_1,$$
$$x(u,v) \leq y(v) \ \forall (u,v) \in A,$$
$$0 \leq y(v) \leq 1 \ \forall v \in V_2,$$
$$x(u,v) \geq 0 \ \forall (u,v) \in A.$$

We denote by $\Pi(G)$ the polytope defined by (23)–(26). Notice that $\Pi(G)$ is a face of $P(G)$. Let $\bar{V}_1$ be the set of nodes $u \in V_1$ with $|\delta^+(u)| = 1$. Let $\bar{V}_2$ be the set of nodes in $V_2$ that are adjacent to a node in $\bar{V}_1$. It is clear that the variables associated with nodes in $\bar{V}_2$ should be fixed, i.e., $y(v) = 1$ for all $v \in \bar{V}_2$. Let us denote by $\bar{G}$ the subgraph induced by $V \setminus \bar{V}_2$. In this section we prove that $\Pi(G)$ is an integral polytope if and only if $\bar{G}$ has no odd cycle.

Let us first assume that $\bar{G}$ has no odd cycle. As before, we suppose that $\bar{z}$ is a fractional extreme point of $\Pi(G)$. The analogues of Lemmas 1–4 apply here. Thus we can assume that we deal with a connected component $G'$. Lemma 2 implies that any node in $\bar{V}_2$ is not in a cycle of $G'$. Therefore, $G'$ has no odd cycle and $P(G')$ is an integral polytope. Since $\Pi(G')$ is a face of $P(G')$, we have a contradiction.

Now let $C$ be an odd cycle of $\bar{G}$. We can define a fractional vector as follows:

$$\bar{y}(v) = 1/2 \ \forall \text{ nodes } v \in V_2 \cap V(C),$$
$$\bar{x}(a) = 1/2 \ \text{ for } a \in A(C),$$
$$\bar{y}(v) = 1 \ \forall \text{ nodes } v \in V_2 \setminus V(C).$$

For every node $u \in V_1 \setminus V(C)$, we look for an arc $(u,v) \in \delta^+(u)$. If $\bar{y}(v) = 1$, we set $\bar{x}(u,v) = 1$. If $\bar{y}(v) = 1/2$, then there is another arc $(u,w) \in \delta^+(u)$ such that $\bar{y}(w) = 1/2$ or $\bar{y}(w) = 1$. We set $\bar{x}(u,v) = \bar{x}(u,w) = 1/2$. Finally, we set $\bar{x}(a) = 0$ for each remaining arc $a$. This vector satisfies (23)–(26), but it violates the inequality (22) associated with $C$. So in this case (23)–(26) does not define an integral polytope. Thus we can state the following.

**Theorem 9.** The system (23)–(26) defines an integral polytope if and only if $\bar{G}$ has no odd cycle.

**Theorem 10.** The UFLP is polynomially solvable for graphs $G$ such that $\bar{G}$ has no odd cycle.

This class of bipartite graphs can be recognized in polynomial time as described in section 7.

**Acknowledgments.** We are grateful to Gérard Cornuéjols for pointing out references [19] and [9]. We also thank the referees for their helpful comments.
REFERENCES