Revista de Matemática: Teoría y Aplicaciones 2012 **19**(2) : 201–210
cimpa – ucr issn: 1409-2433

VARIANTS OF THE MIXED POSTMAN PROBLEM SOLVABLE USING LINEAR PROGRAMMING

VARIANTES DEL PROBLEMA DEL CARTERO MIXTO QUE SE PUEDEN RESOLVER USANDO PROGRAMACIÓN LINEAL

Francisco Javier Zaragoza Martínez*
Rafael López Bracho†

Received: 27 Nov 2009; Revised: 27 Aug 2011; Accepted: 10 May 2012

---

*Universidad Autónoma Metropolitana Unidad Azcapotzalco, Departamento de Sistemas. Av. San Pablo 180, México, D.F., México 02200. E-mail franz@correo.azc.uam.mx

†Universidad Autónoma Metropolitana Unidad Azcapotzalco, Departamento de Sistemas. Av. San Pablo 180, México, D.F., México 02200. E-mail rlb@correo.azc.uam.mx

Abstract

Given a connected mixed graph with costs on its edges and arcs, the mixed postman problem consists of finding a minimum cost closed tour of the mixed graph traversing all of its edges and arcs. It is well-known that this problem is NP-hard. However, under certain conditions, the problem can be solved in polynomial time using linear programming, in other words, the corresponding polyhedra are integral. Some of these conditions are: the mixed graph is series-parallel or the mixed graph is even. Also, we show that if we add the constraint that each edge is traversed exactly once then the problem can be solved in polynomial time if the set of arcs forms a forest.
Keywords: Mixed graph, postman problem, linear programming.

Resumen

Dada una gráfica mixta y conexa con costos en sus aristas y arcos, el problema del cartero mixto consiste en encontrar un circuito cerrado de la gráfica mixta que recorra sus aristas y arcos a costo mínimo. Se sabe que este problema es NP-duro. Sin embargo, bajo ciertas condiciones adicionales, el problema se puede resolver en tiempo polinomial usando programación lineal, en otras palabras, los poliedros correspondientes son enteros. Algunas de estas condiciones son: la gráfica mixta es serie paralelo o la gráfica mixta tiene grado total par en todos sus vértices. Además, mostramos que si agregamos la restricción adicional de que cada arista se recorra exactamente una vez entonces el problema se puede resolver en tiempo polinomial si el conjunto de arcos forma un bosque.
Palabras clave: Gráfica mixta, problema de cartero, programación lineal.

Mathematics Subject Classification: 05C45, 90C35.

1 Introduction

As the name indicates, *postman* problems [1] are those faced by a postman who needs to deliver mail to all streets in a city, starting and ending his labour at the city's post office, and minimizing the length of his walk. In graph theoretical terms, a postman problem consists of finding a minimum cost tour of a graph traversing all its arcs (one-way streets) and edges (two-way streets) at least once. Hence, we can see postman problems as generalizations of *Eulerian* problems.

The postman problem when all streets are one-way, known as the *directed* postman problem, can be solved in polynomial time by a network flow algorithm, and the postman problem when all streets are two-way, known as the *undirected* postman problem, can be solved in polynomial time using Edmonds' matching algorithm, as shown by Edmonds and Johnson [2]. However, Papadimitriou showed that the postman problem becomes NP-hard when both kinds of streets exist [3]. This problem, known as the *mixed postman problem*, is the central topic of this paper.

Under certain conditions, the mixed postman problem can be solved in polynomial time using linear programming, in other words, the corresponding polyhedra are integral. A well-known such condition is when the mixed graph is even, that is, when the total number of edges and arcs incident to each vertex is even. We show that another such condition is when the mixed graph is series-parallel. Also, we show that if we add the constraint that each edge is traversed exactly once then the problem can be solved using linear programming if the set of arcs forms a forest.

2 Preliminaries

A *mixed graph* M is an ordered triple $(V(M), E(M), A(M))$ of three mutually disjoint sets $V(M)$ of *vertices*, $E(M)$ of *edges*, and $A(M)$ of *arcs*. When it is clear from the context, we simply write $M = (V, E, A)$. Each edge $e \in E$ has two *ends* $u, v \in V$, and each arc $a \in A$ has a *head* $u \in V$ and a *tail* $v \in V$. Each edge can be *traversed* from one of its ends to the other, while each arc can be *traversed* from its tail to its head. The *associated directed graph* $\vec{M} = (V, A \cup E^+ \cup E^-)$ of M is the directed graph obtained from M by replacing each edge $e \in E$ with two oppositely oriented arcs $e^+ \in E^+$ and $e^- \in E^-$ between the same vertices. The *underlying undirected graph* G of M is the undirected graph obtained from M by replacing each arc $a \in A$ with an edge between the same vertices.

Let $S \subseteq V$. The *undirected cut* $\delta_E(S)$ determined by S is the set of edges with one end in S and the other end in $\bar{S} = V \setminus S$. The *directed cut* $\delta_A(S)$ determined by S is the set of arcs with tails in S and heads in $\bar{S}$. The *total cut* $\delta_M(S)$ determined by S is the set $\delta_E(S) \cup \delta_A(S) \cup \delta_A(\bar{S})$. For single vertices $v \in V(M)$ we write $\delta_E(v)$, $\delta_A(v)$, $\delta_M(v)$ instead of $\delta_E(\{v\})$, $\delta_A(\{v\})$, $\delta_M(\{v\})$, respectively. We also define the *degree of* S as $d_E(S) = |\delta_E(S)|$, and the *total degree of* S as $d_M(S) = |\delta_M(S)|$.

A *walk* from v_0 to v_n is an ordered tuple $W = (v_0, e_1, v_1, \ldots, e_n, v_n)$ on $V \cup E \cup A$ such that, for all $1 \leq i \leq n$, e_i can be traversed from v_{i-1} to v_i. If $v_0 = v_n$, W is said to be a *closed* walk. If, for any two vertices u and v, there is a walk from u to v, we say that M is *strongly connected*. If W is closed and uses all vertices of M, we call it a *tour*, and if it traverses each edge and arc exactly once, we call it *Eulerian*. If $e_1, \ldots, e_n$ are pairwise distinct, W is called a *trail*. If W is a closed trail, and $v_1, \ldots, v_n$ are pairwise distinct, we call it a *cycle*.

A connected, undirected graph is *series-parallel* if it cannot be transformed into K_4 by a sequence of edge deletions or contractions. Equivalently, if it does not contain a subdivision of K_4 as a subgraph [4]. Several other equivalent characterizations exist.

Given a matrix $A \in \mathbb{Q}^{n\times m}$ and a vector $b \in \mathbb{Q}^n$, the *polyhedron* determined by A and b is the set $P = \{x \in \mathbb{R}^m : Ax \leq b\}$. A vector $x \in P$ is called an *extreme point* of P if x is not a convex combination of vectors in $P\setminus\{x\}$. For our purposes, P is *integral* if all its extreme points have integer coordinates, and it is *half-integral* if all its extreme points have coordinates which are integer multiples of $\frac{1}{2}$.

Let S be a set, and let $T \subseteq S$. If $x \in \mathbb{R}^S$, we define $x(T) = \sum_{t\in T} x_t$. The *characteristic* vector χ^T of T with respect to S is defined by the entries $\chi^T(t) = 1$ if $t \in T$, and $\chi^T(t) = 0$ otherwise. If $T = S$ we write $\mathbf{1}_S$ or $\mathbf{1}$ instead of χ^S, if T consists of only one element t we write $\mathbf{1}_t$ instead of $\chi^{\{t\}}$, and if T is empty we write $\mathbf{0}_S$ or $\mathbf{0}$ instead of $\chi^{\emptyset}$. If $x \in \mathbb{R}^n$, the *positive support* of x is the vector $y \in \mathbb{R}^n$ such that $y_i = 1$ if $x_i > 0$, and $y_i = 0$ otherwise, and it is denoted by $\mathrm{supp}_+(x)$. The *negative support* $\mathrm{supp}_-(x)$ is defined similarly.

3 Integer programming and linear relaxations

Let $M = (V, E, A)$ be a strongly connected mixed graph, and let $c \in \mathbb{Q}_+^{E\cup A}$. A *postman tour* of M is a tour that traverses all edges and arcs of M at least once. The *cost* of a postman tour is the sum of the costs of all edges and arcs traversed, counting repetitions. The *mixed postman problem* is to find the minimum cost of a postman tour. We present an integer programming formulation due to Kappauf and Koehler [5], and Christofides et al. [6]. Similar formulations were given by other authors [2, 7, 8]. All these formulations are based on the following characterization of mixed Eulerian graphs.

Theorem 3.1 (Veblen [9]) *A connected, mixed graph M is Eulerian if and only if M is the disjoint union of some cycles.*

Let $\vec{M} = (V, A \cup E^+ \cup E^-)$ be the associated directed graph of M. For every $e \in E$, let $c_{e^+} = c_{e^-} = c_e$. A nonnegative integer *circulation* x of $\vec{M}$ (a vector on $A \cup E^+ \cup E^-$ such that $x(\vec{\delta}(\bar{v})) = x(\vec{\delta}(v))$ for every $v \in V$, for more on the theory of *flows* see [10]) is the incidence vector of a postman tour of M if and only if $x_e \geq 1$ for all $e \in A$, and $x_{e^+} + x_{e^-} \geq 1$ for all $e \in E$. Therefore, we obtain the integer program:

$$\begin{aligned}
\text{MMPT1}(M,c) &= \min c_A^\top x_A + c_E^\top x_E^+ + c_E^\top x_E^- && (1)\\
\text{subject to} & \\
x(\vec{\delta}(\bar{v})) - x(\vec{\delta}(v)) &= 0 \text{ for all } v \in V && (2)\\
x_a &\geq 1 \text{ for all } a \in A && (3)\\
x_{e^+} + x_{e^-} &\geq 1 \text{ for all } e \in E && (4)\\
x_a &\geq 0 \text{ and integer for all } a \in A \cup E^+ \cup E^-. && (5)
\end{aligned}$$

Let $\mathcal{P}^1_{MPT}(M)$ be the convex hull of the feasible solutions to the integer program above, and let $\mathcal{Q}^1_{MPT}(M)$ be the set of feasible solutions to its linear programming relaxation:

$$\begin{aligned}
\text{LMMPT1}(M,c) &= \min c_A^\top x_A + c_E^\top x_E^+ + c_E^\top x_E^- && (6)\\
\text{subject to} & \\
x(\vec{\delta}(\bar{v})) - x(\vec{\delta}(v)) &= 0 \text{ for all } v \in V && (7)\\
x_a &\geq 1 \text{ for all } a \in A && (8)\\
x_{e^+} + x_{e^-} &\geq 1 \text{ for all } e \in E && (9)\\
x_a &\geq 0 \text{ for all } a \in A \cup E^+ \cup E^-. && (10)
\end{aligned}$$

We present here a first integrality result. A mixed graph $M = (V, E, A)$ is *even* if the total degree $d_{E\cup A}(v)$ is even for every $v \in V$.

Theorem 3.2 (Edmonds and Johnson [2]) *If M is even, then the polyhedron $\mathcal{Q}^1_{MPT}(M)$ is integral. Therefore the mixed postman problem can be solved in polynomial time for the class of even mixed graphs.*

A simple proof of this theorem can be found in [11]. A related result due independently to Kappauf and Koehler [5], Ralphs [8], and Win [12] says that, in general, $\mathcal{Q}^1_{MPT}(M)$ is half-integral. We say that $e \in E$ is *tight* if $x_{e^+} + x_{e^-} = 1$.

Theorem 3.3 (Kappauf and Koehler, Ralphs, Win) *Every extreme point x of the polyhedron $\mathcal{Q}^1_{MPT}(M)$ has components whose values are either $\frac{1}{2}$ or a nonnegative integer. Moreover, fractional components occur only on tight edges.*

3.1 Odd-cut constraints

Let $S \subset V$ be such that $d_M(S)$ is odd. Then, in any postman tour of M, at least one element of $\delta_M(S)$ must be duplicated. Therefore, the inequality

$$x(\vec{\delta}(S)) + x(\vec{\delta}(\bar{S})) \geq \vec{d}(S) + \vec{d}(\bar{S}) + 1 \qquad (11)$$

is valid for $\mathcal{P}^1_{MPT}(M)$. Note that this inequality can be rewritten as:

$$x(\delta_M(S)) \geq d_M(S) + 1, \tag{12}$$

for all S such that $d_M(S)$ is odd. Let $\mathcal{O}^1_{MPT}(M)$ be the subset of $\mathcal{Q}^1_{MPT}(M)$ that satisfies all the odd-cut constraints (11).

Theorem 3.4 (Grötschel and Win [7, 12]) *There exists a polynomial-time algorithm that, given a mixed graph $M = (V, E, A)$ and a vector $c \in \mathbb{Q}_+^{A\cup E}$, finds a vector $x \in \mathbb{Q}_+^{A\cup E^+\cup E^-}$ minimizing $c^\top x$ over $\mathcal{O}^1_{MPT}(M)$. Hence, the mixed postman problem can be solved in polynomial time for the class of mixed graphs M with $\mathcal{O}^1_{MPT}(M)$ integral.*

This is a consequence of the equivalence of optimization and separation [13].

Win conjectured that this class of graphs included the series-parallel graphs [12]. This was proved later in [14].

Theorem 3.5 *Let G be a series-parallel undirected graph. Then $\mathcal{O}^1_{MPT}(G)$ is integral.*

Observing that if the mixed graph M has an underlying undirected series-parallel graph G then the polyhedron $\mathcal{O}^1_{MPT}(M)$ is a projection of the polyhedron $\mathcal{O}^1_{MPT}(G)$ and, therefore, it is also integral we obtain:

Corollary 3.6 *Let M be a series-parallel mixed graph. Then $\mathcal{O}^1_{MPT}(M)$ is integral.*

Corollary 3.7 *The mixed postman problem can be solved in polynomial time for the class of series-parallel mixed graphs.*

A similar result was obtained in [15] using dynamic programming.

4 The bounded mixed postman problem

We can generalize the mixed postman problem by providing, for each edge and arc e, two integers $u_e \geq l_e \geq 0$, and requiring that e is used at least l_e and at most u_e times. We allow $u_e = \infty$ (that is, no upper bound), but l_e must be finite. We say that a family of circuits $\mathcal{C}$ is an (l, u)-*postman tour* (or simply, a *bounded postman tour*) if, for every edge and arc e, the total number of times that e is used by the elements of $\mathcal{C}$ is at least l_e and at most u_e. Note that a bounded postman tour is not necessarily a tour simply because the edges and arcs of the elements of $\mathcal{C}$ may induce a disconnected subgraph of M. However, if the spanning subgraph of M induced by the support of l is connected, then a bounded postman tour is a tour.

The study of these problems was originally suggested by Edmonds and Johnson [2]. Since the decision version of the mixed postman problem is NP-complete, it follows that the decision version of the bounded postman problem is also NP-complete. Tohyama and Adachi proved that the same is true even if all upper bounds are finite (even if $u_e = 2$ and $l_e = 1$ for all $e \in E \cup A$) [16]. It is easy to see that the following is an integer programming formulation for the bounded mixed postman problem:

$$
\begin{aligned}
\text{MBMPT1}(M,l,u,c) \quad &= \quad \min c_A^\top x_A + c_E^\top x_E^+ + c_E^\top x_E^- && (13)\\
\text{subject to} \quad & && \\
x(\vec{\delta}(\bar{v})) - x(\vec{\delta}(v)) \quad &= \quad 0 \text{ for all } v \in V && (14)\\
u_a \;\geq\; x_a \quad &\geq \quad l_a \text{ for all } a \in A && (15)\\
u_e \;\geq\; x_{e^+} + x_{e^-} \quad &\geq \quad l_e \text{ for all } e \in E && (16)\\
x_e \quad & \quad\ \text{integer for all } e \in A \cup E^+ \cup E^-. && (17)
\end{aligned}
$$

Let $\mathcal{P}^1_{BMPT}(M,l,u)$ be the convex hull of the feasible solutions to the integer program above, let $\mathcal{Q}^1_{BMPT}(M,l,u)$ be the set of feasible solutions to its linear programming relaxation. Similarly to Theorem 3.3, we can prove that $\mathcal{Q}^1_{BMPT}(M,l,u)$ is half-integral. We say that $e \in E$ is *bound tight* if $x_{e^+} + x_{e^-} = u_e$ or $x_{e^+} + x_{e^-} = l_e$.

Theorem 4.1 *Every extreme point x of the polyhedron $\mathcal{Q}^1_{BMPT}(M,l,u)$ is half-integral. Moreover, fractional components occur only on bound tight edges.*

We present now a special case of the bounded mixed postman problem.

5 The arcs postman problem

We say that a postman tour of M is an *arcs postman tour* if it uses each edge of M exactly once. The *arcs postman problem* consists in finding a minimum cost arcs postman tour of a graph. As above, this problem can be solved in polynomial time if M is even or series-parallel. However, even to decide whether M has an arcs postman tour is known to be NP-complete even when M is restricted to be planar [17]. Our main contribution is to show that the optimization problem can be solved in polynomial time using linear programming if the set of arcs forms a forest.

5.1 Directed forests

If we disallow cycles in the underlying undirected graph of the directed graph $D = (V, A)$, that is, if we restrict D to be a forest, we can solve in polynomial time not only the feasibility version, but also the optimization version of the arcs postman problem. Let U be a subset of the vertices of M. We say that U is *outgoing* if no arc enters U, and that it is *undirected* if no arc crosses the cut induced by U. Let us observe first that if M contains a vertex of odd degree with no incident arcs then M does not have an arcs postman tour. Similarly, if M contains an undirected set of vertices U such that $d_M(U)$ is odd (that is, U is *odd*) then M does not have an arcs postman tour. We call this the *even-cut condition*.

We use the next lemma in our algorithm, which guarantees that certain arcs must be duplicated in any arcs postman tour.

Lemma 5.1 *Assume that $S \subseteq V$ is an odd outgoing set of M such that $\delta_A(S)$ contains exactly one arc a. Let M' be the mixed graph obtained from M by adding an arc a' parallel to a. Then M has an arcs postman tour if and only if M' does.*

Proof. Since S is odd, any arcs postman tour of M must use at least twice some element of $\delta_M(S)$. Since a is the only arc in $\delta_M(S)$, any arcs postman tour of M must use a at least twice. □

Let $G = (V, E)$ be an undirected graph. We say that $J \subseteq E$ is a *T-join* of (E, T) if, for every $v \in V$, $d_J(v)$ is odd if and only if $v \in T$.

Theorem 5.2 *There exists a polynomial-time algorithm for the arcs postman problem, with input $M = (V, E, A)$ restricted so that $D = (V, A)$ is a forest.*

Proof. Let $T \subseteq V$ be the set of vertices v with $d_M(v)$ odd. If there exists $v \in T$ such that v is not incident to any arc, then M fails the even-cut condition at v, so we can assume that each vertex in T is incident to some arc. Let $F = (V_F, A_F)$ be a maximal tree in D. If $T_F = V_F \cap T$ is odd, then M fails the even-cut condition at V_F, so we can assume that it is even. For each $a \in A_F$ there exists a unique outgoing subset $V_a \subseteq V_F$ such that a is the unique element of $\delta_A(V_a)$. By Lemma 5.1, if V_a is odd then a must be used at least twice by any arcs postman tour of M. Let A_2 be the set of all such arcs, and let $A_1 = A \setminus A_2$. Note that A_2 is the unique minimal T-join of $(\bar{D}, T)$, and hence can be found in polynomial time. Let $l_e = u_e = 1$ for all $e \in E$, $l_a = 1$ and $u_a = +\infty$ for all $a \in A_1$, and $l_a = 2$ and $u_a = +\infty$ for all $a \in A_2$. By Theorem 4.1, since $l(\delta_M(v))$ is even for all $v \in V$, the polyhedron $\mathcal{Q}^1_{BMPT}(M, l, u)$ is integral, and hence the linear program

$$\begin{aligned}
\text{LMAPTF}(M, c) &= \min c^\top x_A && (18)\\
\text{subject to} & && \\
x(\vec{\delta}(\bar{v})) - x(\vec{\delta}(v)) &= 0 \text{ for all } v \in V && (19)\\
x_a &\geq 1 \text{ for all } a \in A_1 && (20)\\
x_a &\geq 2 \text{ for all } a \in A_2 && (21)\\
x_{e^+} + x_{e^-} &= 1 \text{ for all } e \in E && (22)\\
x_e &\geq 0 \text{ for all } e \in A \cup E^+ \cup E^-, && (23)
\end{aligned}$$

solves the given instance of the arcs postman problem. □

6 Conclusions and further work

It was previously known that two special cases of the mixed postman problem can be solved in polynomial time (one using linear programming, another using dynamic programming). We present a variant of the mixed postman problem that has the same property, that is, that a linear programming formulation induces a polyhedron with integral vertices and therefore linear programming also solves this problem in polynomial time. We believe that these statements are true in a more general setting, and this is one aspect that we would like to investigate further. However, all the results given depend on the equivalence theorem between separation and optimization [13] and in the ellipsoid method for linear programming [18]. Therefore, we would like to further investigate whether our problems can be solved by a combinatorial algorithm.

Acknowledgments: One of the authors would like to thank Bill Cunningham for his insightful comments and for his encouragement during his stay at the University of Waterloo. This work was supported by Universidad Autónoma Metropolitana Azcapotzalco grant 2270314 and by CONACyT Sistema Nacional de Investigadores grant 33694.

References

[1] Guan, M.G. (1960) "Graphic programming using odd or even points", *Chinese Math.* **1**: 273–277.

[2] Edmonds, J.; Johnson, E.L. (1973) "Matching, Euler tours and the Chinese postman", *Math. Programming* **5**: 88–124.

[3] Papadimitriou, C.H. (1976) "On the complexity of edge traversing", *J. ACM* **23**: 544–554.

[4] Duffin, R.J. (1965) "Topology of series-parallel networks", *Journal of Mathematical Analysis and Applications* **10**: 303–318.

[5] Kappauf, C.H.; Koehler, G.J. (1979) "The mixed postman problem", *Discrete Appl. Math.* **1**: 89–103.

[6] Christofides, N.; Benavent, E.; Campos, V.; Corberán, Á.; Mota, E. (1984) "An optimal method for the mixed postman problem", in: P. Thoft-Christensen (Ed.) *Lecture Notes in Control and Inform. Sci.* **59** Springer, Berlin: 641–649.

[7] Grötschel, M.; Win, Z. (1992) "A cutting plane algorithm for the windy postman problem", *Math. Programming* **55**: 339–358.

[8] Ralphs, T.K. (1993) "On the mixed Chinese postman problem", *Oper. Res. Lett.* **14**: 123–127.

[9] Veblen, O. (1912/1913) "An application of modular equations in analysis situs", *Ann. of Math.* **2**: 86–94.

[10] Ford, L.R., Jr.; Fulkerson, D.R. (1962) *Flows in Networks*. Princeton University Press, Princeton, N.J.

[11] Zaragoza Martínez, F.J. (2005) "Linear programming relaxations of the mixed postman problem", *Morfismos* **9**: 21–34.

[12] Win, Z. (1989) "On the windy postman problem on Eulerian graphs", *Math. Programming* **44**: 97–112.

[13] Grötschel, M.; Lovász, L.; Schrijver, A. (1981) "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* **1**: 169–197.

[14] Zaragoza Martínez, F. J. (2008) "Series-parallel graphs are windy postman perfect", *Discrete Mathematics* **308**: 1366–1374.

[15] Fernandes, C.G.; Lee, O.; Wakabayashi, Y. (2009) "Minimum cycle cover and chinese postman problems on mixed graphs with bounded tree-width", *Discrete Applied Mathematics* **157**: 272 – 279.

[16] Tohyama, H.; Adachi, A. (1996) "Complexity of a restricted Chinese postman problem", *Trans. Inform. Process. Soc. Japan* **37**: 1886–1896.

[17] Zaragoza Martínez, F.J. (2003) *Postman Problems on Mixed Graphs*. Ph.D. Thesis, University of Waterloo, Waterloo, On.

[18] Khachiyan, L.G. (1979) "A polynomial algorithm in linear programming", *Dokl. Akad. Nauk SSSR* **244**: 1093–1096.