



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL  
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

ESTUDIO DE UN MODELO DE EVASIÓN EN EL TRANSPORTE PÚBLICO

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE  
OPERACIONES  
MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO

BASTIÁN MATÍAS BAHAMONDES PIZARRO

PROFESOR GUÍA:  
JOSÉ RAFAEL CORREA HAEUSSLER

PROFESOR CO-GUÍA:  
ROBERTO MARIO COMINETTI COTTI-COMETTI

MIEMBROS DE LA COMISIÓN:  
JANNIK MATUSCHKE  
JOSÉ ANTONIO SOTO SAN MARTÍN

Agradece financiamiento de CONICYT y al Núcleo Milenio Información y Coordinación  
en Redes.

SANTIAGO DE CHILE  
ABRIL 2016

RESUMEN DE LA TESIS PARA OPTAR AL  
TÍTULO DE: Ingeniero Civil Matemático  
y grado de Magíster en Gestión de Operaciones.  
POR: Bastián Matías Bahamondes Pizarro  
FECHA: 06/04/2016  
PROFESOR GUÍA: José Rafael Correa Haeussler

## ESTUDIO DE UN MODELO DE EVASIÓN EN EL TRANSPORTE PÚBLICO

El problema de la evasión del pago del pasaje en el transporte público es transversal a distintos sistemas de transporte a lo largo del mundo, causando pérdidas a las empresas operadoras y al Estado. Ya que la instalación de barreras físicas y torniquetes no siempre es posible o deseable, diversos sistemas recurren al control del pago del pasaje a bordo. En esta tesis se modela el fenómeno de la evasión como un juego de Stackelberg sobre un grafo dirigido en el que interactúan dos jugadores: un inspector (el líder) y un evasor (el seguidor). Para controlar la evasión, el líder se ubica aleatoriamente sobre a lo más un arco de la red. Para ello, su estrategia consiste en fijar una distribución de probabilidad de inspección sobre los arcos del grafo. El seguidor, quien desea viajar entre dos nodos fijos sin pagar el pasaje, reacciona escogiendo como estrategia un camino, y lo recorre de manera que si atraviesa el arco donde se encuentra el inspector, deberá pagar una multa antes de continuar su viaje.

Dependiendo de su comportamiento después de pagar la multa, se hace la distinción entre un seguidor no adaptativo, que continúa su viaje por el camino escogido previamente sin modificar su ruta, y uno adaptativo, que continúa su trayecto a través de un camino mínimo con respecto a las distancias. Dada una distribución de probabilidad de inspección, los tiempos de viaje y el valor de la multa, el objetivo del seguidor es encontrar un camino de costo esperado mínimo. Mientras que para el seguidor no adaptativo este problema se reduce a encontrar un camino mínimo con respecto a una versión modificada de las distancias, la pregunta por la complejidad del problema del seguidor adaptativo queda abierta. En esta tesis se diseñan algoritmos pseudopolinomiales para este último problema en grafos acíclicos y en grafos generales, además de un algoritmo fuertemente polinomial en grafos generales para el caso en que la distribución de inspección es uniforme sobre los arcos. Por último, se presenta un esquema de aproximación a tiempo totalmente polinomial (FPTAS) en grafos generales, y se amplía un resultado previo de Correa et al., donde se muestra que el cociente entre los valores óptimos de los problemas no adaptativo y adaptativo es a lo más  $4/3$ , mostrando un ejemplo donde la cota es ajustada.

El líder, por su parte, es modelado de acuerdo a dos variantes. En la primera de ellas, su objetivo es maximizar el costo esperado del seguidor. Se prueba que cuando el seguidor es no adaptativo, este problema se puede resolver en tiempo polinomial, mientras que para el seguidor adaptativo se diseña un FPTAS. En la segunda variante, el líder busca maximizar el valor esperado de la multa recolectada. En el caso del seguidor no adaptativo, se demuestra que el problema de decisión asociado es NP-completo, mientras que para el seguidor adaptativo la pregunta por la complejidad se deja abierta.

## AGRADECIMIENTOS

No puedo comenzar sin agradecer a mi familia. A mi mamá por su amor infinito e incondicional, a mi papá por su ejemplo de vida y a mi hermana por apoyarme siempre. A mi abuelita Mimi por ser la más cariñosa y la mejor chef de Chile.

Agradezco también a mi profesor guía José Correa por haberme propuesto un tema tan entretenido como interesante; por su valioso aporte de ideas para esta tesis, por su inagotable paciencia y por ser un excelente guía y consejero.

A Jannik Matuschke, por haberme recibido cordialmente durante mi estadía en Roma y haber concebido tantas ideas creativas para abordar los problemas de este trabajo, que definitivamente no habría sido lo mismo sin él.

A los profesores Roberto Cominetti, Fernando Ordóñez y Daniel Espinoza por haber contribuido a mi formación y haberme hecho crecer académica y personalmente cuando trabajé con ellos.

A mis compañeros y amigos del DIM: Sebastián, Ricardo, Sylvia, Ignacio, Víctor, Felipe, Alfredo, Francisco, Nikolas, David, Gianfranco y Marco; con quienes mi paso por la universidad se hizo una experiencia muy entretenida y enriquecedora.

A mis compañeros del colegio y de la vida, con quienes he compartido tan buenos momentos y de seguro seguiremos haciéndolo: Alejandro, Carlos, Cristian, Emilio, Javier, Juan Carlos, Erick, Sebastián y Rodrigo.

A los funcionarios del DIM y del MGO, entre ellos Eterin, Oscar, María Inés, Linda y Fernanda por su enorme dedicación y hacer que todo funcione impecablemente.

Y a tantos otros y otras que no alcanzo a nombrar en estas líneas, pero que han hecho de mi vida una serie de aprendizajes, risas y coscorrones.

Finalmente, agradezco al Núcleo Milenio Información y Coordinación en Redes por permitirme gentilmente participar en sus actividades y a Conicyt por el financiamiento otorgado para llevar a cabo este trabajo.

# Tabla de contenido

Tabla de contenido	IV
Índice de figuras	v
<b>1 Introducción</b>	<b>1</b>
1.1 Introducción	1
1.2 Trabajo relacionado	2
1.3 Descripción del modelo	3
1.3.1 Problema de minimización del seguidor	4
1.3.2 Problema de maximización del líder	6
<b>2 Problema de minimización del seguidor adaptativo</b>	<b>7</b>
2.1 Un algoritmo pseudopolinomial para grafos acíclicos	7
2.2 Extensión a grafos generales	11
2.3 Un algoritmo pseudopolinomial para grafos generales	21
2.4 Una algoritmo fuertemente polinomial para la distribución uniforme	25
2.5 Un FPTAS para grafos generales	27
2.6 Impacto de la adaptatividad	32
<b>3 Problema de maximización del líder</b>	<b>33</b>
3.1 Problema de maximización del líder ante un seguidor no adaptativo	33
3.1.1 Maximización del costo esperado del seguidor	33
3.1.2 Maximización de la multa recolectada	34
3.2 Problema de maximización del líder ante un seguidor adaptativo	38
3.2.1 Maximización del costo esperado del seguidor	38
3.2.2 Maximización de la multa recolectada	44
<b>Bibliografía</b>	<b>45</b>

# Índice de figuras

2.1	Dado un paseo $W \in \mathcal{W}_{st}$ , el costo esperado del camino directo $P \in \mathcal{P}_{st}$ puede ser estrictamente mayor. . . . .	12
2.2	El ejemplo muestra que $R_s(q)$ puede tener una cantidad exponencial de puntos de quiebre. . . . .	19
2.3	La instancia de la figura muestra que la cota $\text{OPT}_N \leq \frac{4}{3}\text{OPT}_A$ es ajustada. . . . .	32
3.1	Subgrafo para una cláusula $(u_{i1} \vee u_{i2} \vee u_{i3})$ arbitraria. Las copias del arco superior tienen costo 1. Todos los demás arcos tienen costo 0. . . . .	35
3.2	En la figura, a modo de ejemplo, se muestra el grafo correspondiente la fórmula $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$ . Por simplicidad, sólo se especifican los costos de los arcos de costo distinto de 0. . . . .	36

# Capítulo 1

## Introducción

### 1.1. Introducción

El problema de la evasión del pago del pasaje en el transporte público es transversal a distintos sistemas de transporte a lo largo del mundo, generando pérdidas a las empresas operadoras y al Estado, quien usualmente financia a las primeras mediante subsidios. En Santiago, donde la tasa de evasión ronda el 26 % [18], anualmente se pierden alrededor de US\$ 415 millones por no pago del pasaje, mientras que el monto del subsidio asciende a los US\$ 750 millones [13]. En Berlín, si bien la tasa de evasión es cercana al 3.5 %, las pérdidas anuales son del orden de 250 millones de euros [19].

En el caso de las ciudades mencionadas, así como también en Londres, Amsterdam y Roma entre otras, la instalación de barreras físicas o torniquetes en cada uno de los buses y/o estaciones de trenes no siempre es posible o deseable y por ende, los sistemas se sustentan en la confianza en los pasajeros y el control del pago del pasaje a bordo. De esta manera, los pasajeros que son sorprendidos viajando sin haber pagado el pasaje se exponen a ser sancionados con una multa cuyo valor es significativamente más alto que el precio del pasaje.

El objetivo central de esta tesis es proponer y estudiar un modelo para la evasión del pago del pasaje en el transporte público, formulando el problema como un *juego de Stackelberg* que describe la interacción entre un inspector (el líder) y un evasor (el seguidor) en una red a través de un problema de optimización binivel. En el primer nivel, el líder decide la probabilidad de inspección para cada arco de la red, cuyo valor representa la frecuencia de las inspecciones realizadas por él en ese arco. El seguidor, por su parte, actúa en el segundo nivel y reacciona ante la decisión del líder escogiendo un camino que minimice el costo esperado de su viaje. El líder podrá ubicarse en a lo más un arco de la red para controlar la evasión, y si el seguidor escoge un camino que contiene a ese arco, será inspeccionado y antes de continuar su viaje deberá pagar una multa cuyo valor es independiente del arco donde es detectado.

Abordaremos dos variantes posibles para la naturaleza del líder: en una de ellas, su objetivo es maximizar la probabilidad de atrapar al evasor o equivalentemente, maximizar la fracción esperada de la multa recolectada. En la otra variante, el líder busca que el seguidor incurra en el costo esperado más alto posible. Del mismo modo, consideraremos dos variantes que describen el comportamiento del evasor. En la primera de ellas, el seguidor escoge un camino antes de comenzar su trayecto y viaja a través de él independientemente de si es inspeccionado o no. En la segunda, ante el evento de ser inspeccionado, el evasor se adapta y prosigue su viaje a través de un camino mínimo con respecto a las distancias, pues se asume que la multa incluye un pasaje para llegar al destino. Diremos que en este caso el seguidor es adaptativo, mientras que en el primero es no adaptativo.

## 1.2. Trabajo relacionado

El trabajo de esta tesis se sitúa en el contexto de problemas de tipo *Stackelberg network pricing*, en los cuales el líder fija el precio de un subconjunto de arcos de la red y el seguidor escoge el camino más corto definido como la suma de los costos de traslado más el precio de los arcos recorridos. Se sabe que este problema es *NP-hard* [4, 16], y más aún, es difícil de aproximar [11].

Tal vez la variante de mayor similitud al modelo de este trabajo fue la estudiada por Correa, Harks, Kreuzen y Matuschke en [6]. En ella, el líder asigna probabilidades de inspección sobre los arcos sujetas a una restricción global de presupuesto sobre la suma de ellas, que denota un número limitado de inspectores. El problema del seguidor también se presenta en sus versiones adaptativa y no adaptativa, para las cuales diseñan algoritmos exactos y de aproximación respectivamente. En ese modelo, a diferencia del que estudiaremos en esta tesis, para un arco fijo en el camino del seguidor, el evento en que la inspección ocurre en dicho arco es independiente del evento en que el seguidor no ha sido inspeccionado antes. En nuestro modelo, por el contrario, la probabilidad de inspección que el seguidor adaptativo percibe se amplifica a medida que recorre el camino escogido y no ha encontrado al inspector. Con respecto al problema del líder, los autores estudian dos variantes: una donde el valor del pasaje es fijado exógenamente y otra donde puede ser determinado por el líder. Para la segunda versión, diseñan un algoritmo de aproximación basado en programación lineal y utilizan un procedimiento de búsqueda local para realizar un estudio computacional de todas las variantes del problema.

Un modelo previo al anterior, de Borndörfer et al. en [3] aborda el problema suponiendo que hay múltiples seguidores, donde cada uno representa un flujo de usuarios y bajo la fuerte suposición de que tanto las rutas que sigue cada flujo y sus respectivas demandas son conocidas *a priori*. Ellos formulan un problema de programación lineal para encontrar la distribución de inspectores que maximiza la multa recolectada y un problema entero mixto para minimizar la cantidad de evasores.

Otro trabajo relacionado es el de Washburn y Wood [20]. En él, el problema de la evasión es modelado como un juego de suma cero donde el inspector busca maximizar la

probabilidad de atrapar al evasor a la vez que éste escoge un camino que minimiza dicha probabilidad. Ellos muestran que el equilibrio del juego puede ser encontrado en tiempo polinomial usando técnicas de flujo en redes.

Fuera del ámbito de los modelos de evasión, un problema de similares características al del seguidor adaptativo es el *Online replacement path problem*, estudiado por Adjashvili, Oriolo y Senatore en [1]. En él, un mecanismo de ruteo desea enviar un paquete entre dos nodos de una red a través de un camino más corto, mientras que un adversario puede hacer que uno de los arcos falle de manera *online*, después de que el ruteo haya comenzado, forzando al mecanismo a seguir su ruta por un camino alternativo. Los autores probaron que para la versión en que el adversario revela el arco que falla justo antes que el mecanismo de ruteo lo atraviese, minimizar el peor tiempo de llegada puede ser resuelto en tiempo polinomial.

En un contexto relacionado, los problemas de caminos mínimos aditivos donde existe una *trade-off* entre dos funciones de costo han sido estudiados extensivamente en la literatura. Tal es el caso del *Restricted shortest path* (también conocido como *Bicriterion shortest path*) [8, 10, 14] y el *Parametric shortest path* [15, 5], por citar algunos ejemplos.

### 1.3. Descripción del modelo

Modelaremos la red de transporte mediante un grafo dirigido  $G = (V, E)$  y un vector  $c \in \mathbb{Z}_+^E$  que representa los tiempos de viaje en cada arco, medidos en unidades monetarias. Sobre este grafo estudiaremos el fenómeno de la evasión mediante un *juego de Stackelberg*, en el cual interactúan dos jugadores: un evasor (el seguidor), que desea trasladarse entre dos nodos  $s$  y  $t$ , y un inspector u operador de la red (el líder) que se ubicará aleatoriamente sobre a lo más un arco del grafo para controlar la evasión. Para ello, escogerá una distribución de probabilidades sobre los arcos  $\pi \in \mathbb{R}^E$ , con  $\pi \geq 0$  y  $\sum_{e \in E} \pi_e \leq 1$ , donde  $(1 - \sum_{e \in E} \pi_e)$  corresponde a la probabilidad de que el inspector decida no controlar la evasión. Supondremos un régimen estacionario donde el evasor ha aprendido el valor de  $\pi$  y lo utiliza como parte su *input*. De este modo, el evasor escogerá un camino entre  $s$  y  $t$  y lo recorrerá de manera que si atraviesa el arco donde se encuentra el inspector, será multado con una infracción de costo  $F > 0$  antes de continuar su viaje. Dado que el inspector se ubica en a lo más un arco, el evasor podrá ser multado a lo más una vez durante su trayecto.

En lo que sigue, dados dos nodos  $u, v \in V$ , diremos que  $W$  es un  $(u, v)$ -paseo si es una secuencia finita  $(e_1, \dots, e_k)$  con  $e_i = (v_i, v_{i+1}) \in E$ ,  $v_1 = u$  y  $v_{k+1} = v$ . En esta secuencia, pueden haber nodos y/o arcos repetidos. Si además de lo anterior, todos los  $v_i$  son distintos, diremos que  $W$  es un  $(u, v)$ -camino. Si  $W_1$  es un  $(u, v)$ -paseo y  $W_2$  es un  $(v, w)$  paseo, denotaremos por  $W_1 \circ W_2$  al  $(u, w)$ -paseo formado por su concatenación. Usualmente utilizaremos la letra  $W$  para referirnos a paseos y  $P$  para caminos. Cuando el contexto sea lo suficientemente claro, nos referiremos a paseos y caminos omitiendo el



prefijo que denota su origen y destino.

Si  $W = (e_1, \dots, e_k)$  es un  $(u, v)$ -paseo, y  $v_i, v_j$  son dos nodos de  $W$  tales que  $v_i$  aparece antes que  $v_j$  en la secuencia que define a  $W$ , denotaremos por  $W[v_i, v_j]$  al subpaseo formado por los arcos de  $W$  comprendidos entre los nodos  $v_i$  y  $v_j$ , es decir,  $W[v_i, v_j] = (e_i, \dots, e_{j-1})$ . De este modo, la *probabilidad de inspección* del paseo  $W$ , que llamaremos  $\pi(W)$ , está dada por

$$\pi(W) = \sum_{\substack{i=1 \\ e_i \notin W[u, v_i]}}^k \pi_{e_i}.$$

Esta probabilidad corresponde a la suma de las probabilidades de todos los arcos distintos de  $W$ . Por otro lado, definiremos el costo de  $W$  como  $c(W)$ , el cual corresponde a la suma de los costos de sus arcos, incluyendo posibles repeticiones, es decir,

$$c(W) = \sum_{i=1}^k c_{e_i}.$$

Bajo las definiciones anteriores, si  $P$  es un camino, se tiene que  $\pi(P) = \sum_{e \in P} \pi_e$  y  $c(P) = \sum_{e \in P} c_e$ . Dado un  $(u, v)$ -camino  $P$ , diremos que  $P$  es un  $(u, v)$ -*shortest path* si es un camino mínimo con respecto a los costos  $c$ . En ese caso, denotaremos  $c(P) = \text{SP}_c(u, v)$ . Finalmente, nos referiremos por  $\mathcal{W}_{uv}$  y  $\mathcal{P}_{uv}$  al conjunto de  $(u, v)$ -paseos y  $(u, v)$ -caminos de  $G$  respectivamente.

### 1.3.1. Problema de minimización del seguidor

A continuación describiremos la reacción del seguidor ante una estrategia de inspección escogida por el líder. Dependiendo de su comportamiento después de pagar la multa, distinguiremos entre un seguidor adaptativo y uno no adaptativo.

#### Seguidor no adaptativo

En la versión no adaptativa, la estrategia del seguidor es un camino  $P \in \mathcal{P}_{st}$ , el cual recorre de manera que si atraviesa el arco donde se encuentra el inspector, pagará una multa  $F > 0$  y continuará su recorrido sin modificar la ruta escogida. Denotando por  $f_{N, \pi}(P)$  al costo esperado de un camino  $P \in \mathcal{P}_{st}$  cuando la distribución de probabilidad de inspección es  $\pi$ , se tiene que

$$f_{N, \pi}(P) = c(P) + F\pi(P).$$

Llamaremos  $\text{FMP}_N$  (*Non Adaptive Follower's Minimization Problem*) al problema de minimizar el costo esperado del seguidor en su versión no adaptativa, es decir

$$\min_{P \in \mathcal{P}_{st}} f_{N, \pi}(P). \quad (\text{FMP}_N)$$

Notemos que  $f_{N,\pi}(P) = \sum_{e \in P} (c_e + F\pi_e)$ , con lo cual el problema  $\text{FMP}_N$  se reduce a encontrar un  $(s, t)$ -shortest path en  $G$ , donde el costo de cada arco  $e \in E$  está dado por  $c_e + F\pi_e$ . Este problema puede ser resuelto en tiempo fuertemente polinomial utilizando el algoritmo de Dijkstra [2], el cual puede ser implementado usando *Fibonacci heaps* [7] para alcanzar un tiempo de ejecución de  $O(m + n \log n)$ .

### Seguidor adaptativo

En la versión adaptativa, una estrategia del seguidor es un camino  $P = (e_1, \dots, e_k) \in \mathcal{P}_{st}$  que recorre de manera adaptativa ante el evento de encontrarse con el inspector, esto es, si es controlado en el arco  $e_i = (v_i, v_{i+1})$ , pagará la multa  $F > 0$  y al continuar su trayecto lo hará por un *shortest path* entre  $v_{i+1}$  y  $t$ . Cuando el evasor no es controlado, simplemente recorrerá  $P$ . El hecho que motiva esta variante del comportamiento del seguidor es que, una vez controlado el evasor, la multa incluye la adquisición de un pasaje para continuar su viaje. Según esta estrategia, el costo esperado del camino  $P$  bajo la distribución  $\pi$ , que llamaremos  $f_{A,\pi}(P)$ , estará dado por

$$f_{A,\pi}(P) = c_{e_1} + \pi_{e_1} (F + \text{SP}_c(v_2, t)) + (1 - \pi_{e_1}) \left\{ c_{e_2} + \frac{\pi_{e_2}}{1 - \pi_{e_1}} (F + \text{SP}_c(v_3, t)) + \left(1 - \frac{\pi_{e_2}}{1 - \pi_{e_1}}\right) \left[ c_{e_3} + \left(\frac{\pi_{e_3}}{1 - \pi_{e_1} - \pi_{e_2}}\right) (F + \text{SP}_c(v_4, t)) + \dots \right] \dots \right\}.$$

La expresión anterior considera que la probabilidad de inspección que el evasor percibe cuando atraviesa el arco  $e_j$  y no ha sido detectado en los arcos previos se amplifica por un factor  $\frac{1}{1 - \sum_{i=1}^{j-1} \pi_{e_i}}$  al reducir su incertidumbre sobre la ubicación del inspector a medida que recorre  $P$ . De manera más compacta, se llega a que

$$f_{A,\pi}(P) = \sum_{i=1}^k \left(1 - \sum_{j=1}^{i-1} \pi_{e_j}\right) c_{e_i} + \pi_{e_i} (F + \text{SP}_c(v_{i+1}, t)). \quad (1.1)$$

En esta fórmula, el  $i$ -ésimo sumando corresponde al evento en que el evasor llega a  $v_i$  sin ser detectado, en cuyo caso recorre  $e_i$ . El costo esperado del evasor puede ser calculado de manera equivalente considerando los eventos en que es controlado en cada arco de  $P$ , así como también cuando no es inspeccionado y por ende no paga la multa, obteniéndose

$$f_{A,\pi}(P) = \sum_{i=1}^k \left( \pi_{e_i} \left( \sum_{j=1}^i c_{e_j} + F + \text{SP}_c(v_{i+1}, t) \right) \right) + (1 - \pi(P)) c(P). \quad (1.2)$$

Llamaremos  $\text{FMP}_A$  (*Adaptive Follower's Minimization Problem*) al problema de minimizar el costo esperado del seguidor en su versión adaptativa, es decir

$$\min_{P \in \mathcal{P}_{st}} f_{A,\pi}(P), \quad (\text{FMP}_A)$$

Cabe notar que si  $P$  es un  $(s, t)$ -shortest path, la propiedad de suboptimalidad implica que para todo  $v \in V(P)$ ,  $P[v, t]$  es un  $(v, t)$ -shortest path, por lo que si el seguidor adaptativo

es controlado, no necesita modificar su ruta. En este caso se tiene que  $f_{A,\pi}(P) = f_{N,\pi}(P)$ . Por otro lado, como  $F + SP_c(s, t)$ , corresponde al costo de tomar un  $(s, t)$ -shortest path y pagar la multa, para una distribución  $\pi$  fija y un camino  $P^* \in \mathcal{P}_{st}$  tal que  $f_{A,\pi}(P^*) = \min_{P \in \mathcal{P}_{st}} f_{A,\pi}(P)$  se tienen en general las siguientes desigualdades:

$$0 \leq f_{A,\pi}(P^*) \leq f_{N,\pi}(P^*) \leq F + SP_c(s, t). \quad (1.3)$$

### 1.3.2. Problema de maximización del líder

Estudiaremos dos variantes que modelan la naturaleza del líder. En la primera, el operador de la red busca determinar una distribución  $\pi$  sobre los arcos de manera tal que el costo esperado en que incurre el evasor sea lo más alto posible. Llamaremos  $LMP_X^{\text{cost}}$  (*Leader's Maximization Problem*) a este problema:

$$\begin{aligned} & \max_{\pi \in \mathbb{R}^E} \min_{P \in \mathcal{P}_{st}} f_{X,\pi}(P) \\ \text{s.a.} \quad & \pi \geq 0, \sum_{e \in E} \pi_e \leq 1. \end{aligned} \quad (LMP_X^{\text{cost}})$$

En la segunda variante, el líder determina  $\pi$  de manera de maximizar el valor esperado de la multa recolectada, el cual cuando el evasor escoge el camino  $P \in \mathcal{P}_{st}$  es  $F\pi(P)$ . Bajo este enfoque, el líder resuelve el problema  $LMP_X^{\text{fine}}$ :

$$\begin{aligned} & \sup_{\pi \in \mathbb{R}^E} F\pi(P^*) \\ \text{s.a.} \quad & P^* \in \operatorname{argmin}_{P \in \mathcal{P}_{st}} f_{X,\pi}(P) \\ & \pi \geq 0, \sum_{e \in E} \pi_e \leq 1. \end{aligned} \quad (LMP_X^{\text{fine}})$$

En ambos problemas,  $X \in \{A, N\}$ , dependiendo de la naturaleza del seguidor.

# Capítulo 2

## Problema de minimización del seguidor adaptativo

Como vimos en 1.3.1, el problema del seguidor no adaptativo  $FMP_N$  se reduce a calcular un *shortest path*. En este capítulo estudiaremos el problema del seguidor en su versión adaptativa. El *input* de este problema estará dado por una tupla  $(G, s, t, c, F, \pi)$ , donde  $G = (V, E)$  es un grafo dirigido con  $|V| = n$  y  $|E| = m$ ,  $s, t \in V$  son los nodos de origen y destino,  $c \in \mathbb{Z}_+^E$  es el vector de los tiempos de viaje sobre los arcos (en unidades monetarias),  $F \in \mathbb{Z}_+$  es el valor de la multa y  $\pi \in \mathbb{Q}^E$  es una distribución de probabilidades sobre los arcos que satisface  $\pi \geq 0$  y  $\sum_{e \in E} \pi_e \leq 1$ . Para representar la red, para cada nodo  $v \in V$  utilizaremos una lista de adyacencia de arcos  $\delta^+(v)$  que contiene a los arcos emanan de  $v$ , esto es  $\delta^+(v) = \{(v, w) \in E : w \in V\}$ . Como medida del tamaño de los números involucrados, denotaremos por  $c_{\max}$  al máximo costo de un arco, y por  $p_{\max}$  al mayor numerador o denominador que aparece en la representación de  $\pi$ . De esta manera, el tamaño del *input* queda definido por los parámetros  $m, n, \log(c_{\max}), \log(p_{\max})$  y  $\log(F)$ .

### 2.1. Un algoritmo pseudopolinomial para grafos acíclicos

En esta sección supondremos que  $G$  es un grafo acíclico. Se sabe que este tipo de grafos se caracteriza por la existencia de un *orden topológico* sobre sus vértices, esto es, un orden tal que para cada arco  $(u, v) \in E$ , se tiene que  $u < v$ . Más aún, este orden puede ser encontrado en tiempo  $O(m + n)$  [2]. Sin pérdida de generalidad, supondremos que los nodos de  $G$  satisfacen  $s = v_1 < v_2 < \dots < v_n = t$ . Utilizaremos este orden para diseñar un algoritmo basado en programación dinámica.

Dados  $q \in [0, 1]$  y  $v \in V$ , sea  $f_v(q)$  el mínimo costo esperado de un camino entre  $v$  y  $t$  dado que el evasor ha llegado a  $v$  sin haber sido controlado a través de un camino cuya

probabilidad de inspección es  $q$ . Bajo esta definición,  $f_t(q) = 0$  para todo  $q \in [0, 1]$  y

$$f_v(q) = \min_{e=(v,w) \in \delta^+(v)} \left\{ c_e + \frac{\pi_e}{1-q} (F + \text{SP}_c(w, t)) + \left(1 - \frac{\pi_e}{1-q}\right) f_w(q + \pi_e) \right\}$$

para todo  $v \in V \setminus \{t\}$  y  $q \in [0, 1)$ . Multiplicando por  $(1 - q)$  se obtiene

$$(1 - q)f_v(q) = \min_{e=(v,w) \in \delta^+(v)} \{(1 - q)c_e + \pi_e (F + \text{SP}_c(w, t)) + (1 - (q + \pi_e)) f_w(q + \pi_e)\}.$$

Denotando por  $R_v(q) = (1 - q)f_v(q)$ , obtenemos la siguiente relación de recurrencia:

$$R_v(q) = \min_{e=(v,w) \in \delta^+(v)} \{(1 - q)c_e + \pi_e (F + \text{SP}_c(w, t)) + R_w(q + \pi_e)\}.$$

Por comodidad, definiremos  $R_t(q) = 0$  para todo  $q \geq 0$ . Al expandir la recurrencia anterior, obtenemos el siguiente resultado:

**Lema 2.1.** *Para todo  $i = 1, \dots, n$  y  $q \geq 0$ ,  $R_{v_i}(q) = \min_{P \in \mathcal{P}_{v_i t}} \{-qc(P) + f_{A,\pi}(P)\}$ .*

*Demostración.* Por inducción sobre el orden topológico. Para  $v_n = t$ , el resultado es trivial. Sea  $i < n$  y supongamos que para todo  $i < j \leq n$  se cumple que  $R_{v_j}(q) = \min_{P \in \mathcal{P}_{v_j t}} \{-qc(P) + f_{A,\pi}(P)\}$ . Entonces

$$\begin{aligned} R_{v_i}(q) &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \{(1 - q)c_e + \pi_e (F + \text{SP}_c(v_j, t)) + R_{v_j}(q + \pi_e)\} \\ &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \left\{ (1 - q)c_e + \pi_e (F + \text{SP}_c(v_j, t)) + \min_{P \in \mathcal{P}_{v_j t}} \{-(q + \pi_e)c(P) + f_{A,\pi}(P)\} \right\} \\ &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \min_{P \in \mathcal{P}_{v_j t}} \{-q(c_e + c(P)) + c_e + \pi_e (F + \text{SP}_c(v_j, t)) - \pi_e c(P) + f_{A,\pi}(P)\} \\ &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \min_{P \in \mathcal{P}_{v_j t}} \{-qc(e \circ P) + f_{A,\pi}(e \circ P)\} \\ &= \min_{P \in \mathcal{P}_{v_i t}} \{-qc(P) + f_{A,\pi}(P)\}, \end{aligned}$$

donde la cuarta igualdad se debe a la forma de la función objetivo (ver 1.1) y la última igualdad es porque  $P \in \mathcal{P}_{v_i t}$  si y sólo si es de la forma  $e \circ P'$  con  $e = (v_i, v_j) \in \delta^+(v_i)$  y  $P' \in \mathcal{P}_{v_j t}$ .  $\square$

El Lema 2.1 muestra que  $R_{v_i}(q)$  puede ser descrita como un mínimo de funciones lineales afines en  $q$  indexadas por los caminos en  $\mathcal{P}_{v_i t}$ . Como la pendiente de cada una de estas funciones corresponde al costo de algún camino, dado un costo  $k$  fijo, podemos escoger en la descripción a aquella función que domine a las demás funciones asociadas a caminos del mismo costo  $k$ . Para acotar la cantidad de costos posibles de un camino, notemos que para un nodo  $v_i$  fijo, el  $(v_i, t)$ -camino de mayor costo posible cuesta a lo más  $(n - i)c_{\text{máx}}$ . Con esto se logra una representación más eficiente de  $R_{v_i}(q)$ , como se muestra en la siguiente propiedad:

**Propiedad 2.2.** Para todo  $i = 1, \dots, n$  y  $q \geq 0$ ,  $R_{v_i}(q) = \min_{0 \leq k \leq (n-i)c_{\max}} \{-qk + \phi_{v_i}^k\}$ ,  
donde  $\phi_{v_n}^0 = 0$  y

$$\phi_{v_i}^k = \min_{\substack{e=(v_i, v_j) \in \delta^+(v_i) \\ c_e \leq k \leq c_e + (n-j)c_{\max}}} \left\{ c_e + \pi_e (F + \text{SP}_c(v_j, t) - (k - c_e)) + \phi_{v_j}^{k-c_e} \right\}, \quad (2.1)$$

$i = 1, \dots, n-1$ ,  $k = 0, \dots, (n-i)c_{\max}$ .

*Demostración.* Por inducción sobre el orden topológico. El caso  $v_n = t$  es trivial. Sea  $i < n$  y supongamos entonces que la propiedad es cierta para todo  $i < j \leq n$ . Entonces, para todo  $q \geq 0$ ,

$$\begin{aligned} R_{v_i}(q) &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \left\{ (1-q)c_e + \pi_e (F + \text{SP}_c(v_j, t)) + R_{v_j}(q + \pi_e) \right\} \\ &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \left\{ (1-q)c_e + \pi_e (F + \text{SP}_c(v_j, t)) + \min_{0 \leq \ell \leq (n-j)c_{\max}} \left\{ -(q + \pi_e)\ell + \phi_{v_j}^\ell \right\} \right\} \\ &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \min_{0 \leq \ell \leq (n-j)c_{\max}} \left\{ -q(c_e + \ell) + c_e + \pi_e (F + \text{SP}_c(v_j, t)) - \pi_e \ell + \phi_{v_j}^\ell \right\} \\ &= \min_{e=(v_i, v_j) \in \delta^+(v_i)} \min_{c_e \leq k \leq c_e + (n-j)c_{\max}} \left\{ -qk + c_e + \pi_e (F + \text{SP}_c(v_j, t)) - \pi_e(k - c_e) + \phi_{v_j}^{k-c_e} \right\} \\ &= \min_{\substack{e \in \delta^+(v_i) \\ c_e \leq k \leq \max_{e \in \delta^+(v_i)} c_e + (n-j)c_{\max}}} \min_{\substack{e=(v_i, v_j) \in \delta^+(v_i) \\ c_e \leq k \leq c_e + (n-j)c_{\max}}} \left\{ -qk + c_e + \pi_e (F + \text{SP}_c(v_j, t)) \right. \\ &\quad \left. - \pi_e(k - c_e) + \phi_{v_j}^{k-c_e} \right\} \\ &= \min_{0 \leq k \leq (n-i)c_{\max}} \left\{ -qk + \min_{\substack{e=(v_i, v_j) \in \delta^+(v_i) \\ c_e \leq k \leq c_e + (n-j)c_{\max}}} \left\{ c_e + \pi_e (F + \text{SP}_c(v_j, t)) - \pi_e(k - c_e) + \phi_{v_j}^{k-c_e} \right\} \right\}, \end{aligned}$$

donde la última igualdad es producto de que si  $k < \min_{e \in \delta^+(v_i)} c_e$  ó  $k > (n-j)c_{\max} + \max_{e \in \delta^+(v_i)} c_e$ , entonces

$$\{e = (v_i, v_j) \in \delta^+(v_i) : c_e \leq k \leq (n-j)c_{\max} + c_e\} = \emptyset.$$

De la última igualdad se desprende también el valor de los coeficientes  $\phi_{v_i}^k$  en (2.1).  $\square$

La Propiedad 2.2 muestra que  $R_{v_i}(q)$  queda completamente determinada por el valor de los coeficientes  $\phi_{v_i}^k$ ,  $k = 0, \dots, (n-i)c_{\max}$ . Con esto, para resolver el problema FMP<sub>A</sub> en grafos acíclicos, proponemos el siguiente algoritmo que calcula recursivamente el valor de dichos coeficientes utilizando el orden topológico y retorna un  $(s, t)$ -camino  $P^*$  tal que  $f_{A, \pi}(P^*) = R_s(0)$ :

---

**Algoritmo 1** Algoritmo para  $FMP_A$  en grafos acíclicos.
 

---

- 1: Calcular  $SP_c(u, t)$  para todo  $u \in V$ .
  - 2: Ordenar  $V$  en orden topológico. Remover los vértices anteriores a  $s$  y posteriores a  $t$  en el orden. Sean  $s = v_1 < \dots < v_n = t$  los vértices ordenados resultantes.
  - 3: Sea  $\phi_{v_n}^0 \leftarrow 0$ .
  - 4: Sean  $\phi_{v_i}^k \leftarrow \infty$  para todo  $i = 1, \dots, n-1$  y  $k = 0, \dots, (n-i)c_{\text{máx}}$ .
  - 5: Sean  $\text{suc}(v_i, k) \leftarrow \emptyset$  para todo  $i = 1, \dots, n-1$  y  $k = 0, \dots, (n-i)c_{\text{máx}}$ .
  - 6: **for**  $i = 1, \dots, n-1$  **do**
  - 7:     **for**  $k = 0, \dots, (n-i)c_{\text{máx}}$  **do**
  - 8:         **for all**  $e = (v_{n-i}, v_j) \in \delta^+(v_{n-i})$  **do**
  - 9:             **if**  $c_e \leq k \leq c_e + (n-j)c_{\text{máx}}$  **then**
  - 10:                 **if**  $\phi_{v_{n-i}}^k > c_e + \pi_e(F + SP_c(v_j, t) - (k - c_e)) + \phi_{v_j}^{k-c_e}$  **then**
  - 11:                      $\phi_{v_{n-i}}^k \leftarrow c_e + \pi_e(F + SP_c(v_j, t) - (k - c_e)) + \phi_{v_j}^{k-c_e}$ .
  - 12:                      $\text{suc}(v_{n-i}, k) = (v_j, k - c_e)$ .
  - 13: Sea  $\ell \in \{0, \dots, (n-1)c_{\text{máx}}\}$  tal que  $\phi_{v_1}^\ell = \min_{0 \leq k \leq (n-1)c_{\text{máx}}} \phi_{v_1}^k$ .
  - 14: Retornar el camino  $P^*$  formado por los sucesores de  $(v_1, \ell)$ .
- 

**Propiedad 2.3.** El  $(s, t)$ -camino  $P^*$  retornado por el Algoritmo 1 satisface  $f_{A, \pi}(P^*) = \min_{P \in \mathcal{P}_{st}} f_{A, \pi}(P)$ .

*Demostración.* Denotemos por  $P^* = (e_{\alpha_1}, \dots, e_{\alpha_k})$ , con  $e_{\alpha_i} = (v_{\alpha_i}, v_{\alpha_{i+1}})$ ,  $\alpha_1 = 1$ ,  $\alpha_{k+1} = n$  al camino retornado por el Algoritmo 1. Entonces,  $P^*$  es un  $(s, t)$ -camino de costo  $c(P^*) = \ell$  formado por los sucesores de la forma  $(v_{\alpha_i}, \ell - \sum_{j=1}^{i-1} c_{\alpha_j})_{i=1}^{k+1}$ . Entonces, para todo  $i = 1, \dots, k$  se tiene

$$\phi_{v_{\alpha_i}}^{\ell - \sum_{j=1}^{i-1} c_{\alpha_j}} = c_{e_{\alpha_i}} + \pi_{e_{\alpha_i}} \left( F + SP_c(v_{\alpha_{i+1}}, t) - \left( \ell - \sum_{j=1}^i c_{e_{\alpha_j}} \right) \right) + \phi_{v_{\alpha_{i+1}}}^{\ell - \sum_{j=1}^i c_{e_{\alpha_j}}}.$$

Luego,

$$\begin{aligned} f_{A, \pi}(P^*) &= \sum_{i=1}^k \left( \pi_{e_{\alpha_i}} \left( \sum_{j=1}^i c_{e_{\alpha_j}} + F + SP_c(v_{\alpha_{i+1}}, t) \right) \right) + (1 - \pi(P)) c(P) \\ &= \sum_{i=1}^k c_{e_{\alpha_i}} + \pi_{e_{\alpha_i}} \left( F + SP_c(v_{\alpha_{i+1}}, t) - \sum_{j=i+1}^k c_{e_{\alpha_j}} \right) \\ &= \sum_{i=1}^k c_{e_{\alpha_i}} + \pi_{e_{\alpha_i}} \left( F + SP_c(v_{\alpha_{i+1}}, t) - \left( \ell - \sum_{j=i}^i c_{e_{\alpha_j}} \right) \right) \\ &= \sum_{i=1}^k \phi_{v_{\alpha_i}}^{\ell - \sum_{j=1}^{i-1} c_{\alpha_j}} - \phi_{v_{\alpha_{i+1}}}^{\ell - \sum_{j=1}^i c_{e_{\alpha_j}}} \\ &= \phi_{v_1}^\ell - \phi_{v_n}^0 \\ &= \min_{P \in \mathcal{P}_{st}} f_{A, \pi}(P). \end{aligned}$$

□

*Análisis de complejidad.* La complejidad del Algoritmo 1 está dominada por los pasos 1, 4 y 6 - 12. El paso 1 se puede realizar en tiempo  $O(m + n)$  en grafos acíclicos [2], mientras que el paso 4 toma  $O(n^2 c_{\text{máx}})$  operaciones. Los pasos 6 - 12 incurren en un costo de

$$\begin{aligned}
\sum_{i=1}^{n-1} \sum_{k=0}^{(n-i)c_{\text{máx}}} \sum_{e=(v_{n-i}, v_j) \in \delta^+(v_{n-i})} O(1) &= \sum_{i=1}^{n-1} \sum_{k=0}^{(n-i)c_{\text{máx}}} |\delta^+(v_{n-i})| O(1) \\
&= \sum_{i=1}^{n-1} ((n-i)c_{\text{máx}} + 1) |\delta^+(v_{n-i})| O(1) \\
&\leq \sum_{i=1}^{n-1} (nc_{\text{máx}} + 1) |\delta^+(v_{n-i})| O(1) \\
&\leq (nc_{\text{máx}} + 1)mO(1) \\
&= O(mnc_{\text{máx}})
\end{aligned}$$

operaciones. Esto nos da un tiempo de ejecución de  $O((n^2 + mn)c_{\text{máx}})$ , el cual es pseudopolinomial en el tamaño del *input*.

**Teorema 2.4.** *El problema  $\text{FMP}_A$  en grafos acíclicos puede ser resuelto en tiempo  $O((n^2 + mn)c_{\text{máx}})$ .*

## 2.2. Extensión a grafos generales

El objetivo de esta sección es extender las ideas que nos permitieron diseñar el Algoritmo 1 al contexto de grafos generales.

Para abordar el problema del seguidor adaptativo en grafos generales, se debe considerar la posibilidad de que éste decida atravesar más de una vez un mismo arco (por ejemplo, cuando la ruta escogida posee un ciclo). Para ello, extendemos la función objetivo para evaluar el costo esperado de un paseo  $W = (e_1, \dots, e_k) \in \mathcal{W}_{st}$ , con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = s$  y  $v_{k+1} = t$  por

$$f_{A,\pi}(W) = \sum_{\substack{i=1 \\ e_i \notin W[v_1, v_i]}}^k \left( \pi_{e_i} \left( \sum_{j=1}^i c_{e_j} + F + \text{SP}_c(v_{i+1}, t) \right) \right) + (1 - \pi(W))c(W).$$

La expresión anterior corresponde al promedio ponderado de los costos en que incurre el evasor al recorrer  $W$ . Si  $W$  contiene un arco repetido, la probabilidad de inspección cuando éste es atravesado después de la primera vez es nula, pues el evasor ha aprendido que ese arco no está siendo inspeccionado. Por ello,  $f_{A,\pi}(W)$  pondera los costos asociados



a los arcos distintos de  $W$ , lo cual se denota por  $e_i \notin W[v_1, v_i]$  en la suma principal. Alternativamente, escribiremos

$$f_{A,\pi}(W) = \sum_{\substack{e=(v,w) \in W \\ e \notin W[s,v]}} \left( \pi_e \left( \sum_{e' \leq_W e} c_{e'} + F + \text{SP}_c(w, t) \right) \right) + (1 - \pi(W)) c(W),$$

donde  $e' \leq_W e$  denota al conjunto de los arcos que preceden a  $e$  (incluyéndolo) en la secuencia que define a  $W$ .

Intuitivamente, debiese esperarse que dado un paseo  $W \in \mathcal{W}_{st}$ , el costo esperado del camino directo  $P \in \mathcal{P}_{st}$  subyacente a  $W$  debiese ser menor. Sin embargo, esto en general no es así, como lo muestra el siguiente ejemplo:

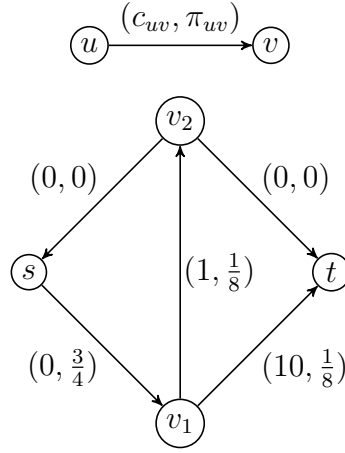


Figura 2.1: Dado un paseo  $W \in \mathcal{W}_{st}$ , el costo esperado del camino directo  $P \in \mathcal{P}_{st}$  puede ser estrictamente mayor.

En el ejemplo de la Figura 2.1, para una multa  $F = 7$ , el paseo  $\{(s, v_1), (v_1, v_2), (v_2, s), (s, v_1), (v_1, t)\}$  tiene costo esperado 9.25, mientras que el camino directo  $\{(s, v_1), (v_1, t)\}$  tiene costo esperado 9.375.

No obstante lo anterior, el siguiente lema asegura que la estrategia óptima del evasor es siempre un camino simple.

**Lema 2.5.** *Sea  $u \in V$ . Entonces, para todo  $W \in \mathcal{W}_{ut}$ , existe  $P \in \mathcal{P}_{ut}$  tal que  $f_{A,\pi}(P) \leq f_{A,\pi}(W)$ .*

*Demostración.* Para  $u \in V$  fijo, sea  $W$  un  $(u, t)$ -paseo y  $P$  un  $(u, t)$ -camino directo sub-

yacente a  $W$ . Se tiene que

$$\begin{aligned}
f_{A,\pi}(W) &= \sum_{\substack{e=(v,w)\in W \\ e\notin W[u,v]}} \pi_e \left( \sum_{e'\leq W e} c_{e'} + F + \text{SP}_c(w,t) \right) + (1 - \pi(W)) c(W) \\
&= \sum_{\substack{e=(v,w)\in W \\ e\notin W[u,v] \\ e\in P}} \pi_e \left( \sum_{e'\leq W e} c_{e'} + F + \text{SP}_c(w,t) \right) \\
&\quad + \sum_{\substack{e=(v,w)\in W \\ e\notin W[u,v] \\ e\notin P}} \pi_e \left( \sum_{e'\leq W e} c_{e'} + F + \text{SP}_c(w,t) \right) + (1 - \pi(W)) c(W) \\
&\geq \sum_{e=(v,w)\in P} \pi_e \left( \sum_{e'\leq P e} c_{e'} + F + \text{SP}_c(w,t) \right) + (\pi(W) - \pi(P)) (F + \text{SP}_c(u,t)) \\
&\quad + (1 - \pi(W)) c(W) \\
&\geq \sum_{e=(v,w)\in P} \pi_e \left( \sum_{e'\leq P e} c_{e'} + F + \text{SP}_c(w,t) \right) + (1 - \pi(P)) \min \{F + \text{SP}_c(u,t), c(W)\}.
\end{aligned}$$

Luego, si  $c(W) > F + \text{SP}_c(u,t)$ , la última desigualdad implica que

$$f_{A,\pi}(W) \geq \pi(P) (F + \text{SP}_c(u,t)) + (1 - \pi(P)) (F + \text{SP}_c(u,t)) = F + \text{SP}_c(u,t),$$

y este último término corresponde a al menos el costo esperado de un  $(u,t)$ -shortest path. En caso contrario, si  $c(W) \leq F + \text{SP}_c(u,t)$ , como  $c(P) \leq c(W)$ , la misma desigualdad mencionada implica que  $f_{A,\pi}(W) \geq f_{A,\pi}(P)$ .  $\square$

El hecho que la expresión que define a  $f_{A,\pi}(W)$  considere sólo la primera aparición de cada arco de  $W$  hace que el cálculo  $f_{A,\pi}$  sea complicado de implementar eficientemente desde un punto de vista algorítmico, pues se haría necesario llevar la cuenta de los arcos previamente recorridos para evitar sumar costos asociados a arcos repetidos. Para solucionar esto, proponemos estudiar la siguiente versión modificada de  $f_{A,\pi}$ , que llamaremos  $\hat{f}_{A,\pi}$ , definida para un paseo  $W = (e_1, \dots, e_k) \in \mathcal{W}_{st}$ , con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = s$  y  $v_{k+1} = t$  por:

$$\hat{f}_{A,\pi}(W) = \sum_{i=1}^k \left( \pi_{e_i} \left( \sum_{j=1}^i c_{e_j} + F + \text{SP}_c(v_{i+1}, t) \right) \right) + (1 - \hat{\pi}(W)) c(W),$$

donde  $\hat{\pi}(W) = \sum_{i=1}^k \pi_{e_i}$ . De manera alternativa, podemos escribir  $\hat{\pi}(W) = \sum_{e\in W} \pi_e$  y

$$\hat{f}_{A,\pi}(W) = \sum_{e=(v,w)\in W} \left( \pi_e \left( \sum_{e'\leq W e} c_{e'} + F + \text{SP}_c(w,t) \right) \right) + (1 - \hat{\pi}(W)) c(W).$$

La función  $\hat{f}_{A,\pi}(W)$  corresponde al promedio ponderado de todos los posibles costos en que incurre el evasor, ponderando incluso el costo en los arcos repetidos como si estos se

recorrieran por primera vez. En el mismo sentido,  $\hat{\pi}(W)$  corresponde a la probabilidad de inspección de  $W$  contando los arcos como si todos ellos fueran distintos. Claramente,  $\hat{f}_{A,\pi}(W)$  y  $\hat{\pi}(W)$  coinciden con  $f_{A,\pi}(W)$  y  $\pi(W)$  respectivamente cuando  $W$  no contiene arcos repetidos. Además,  $\hat{f}_{A,\pi}(W)$  podría ser negativo cuando  $\hat{\pi}(W) > 1$ . La siguiente propiedad es fácil de verificar y omitiremos su demostración.

**Propiedad 2.6.** Las siguientes expresiones son equivalentes a  $\hat{f}_{A,\pi}(W)$ :

- (i)  $\sum_{e=(v,w) \in W} \left( \pi_e \left( c(W[s, w]) + F + SP_c(w, t) \right) \right) + (1 - \hat{\pi}(W)) c(W),$
- (ii)  $\sum_{e=(v,w) \in W} c_e + \pi_e \left( F + SP_c(w, t) - c(W[w, t]) \right),$
- (iii)  $\sum_{e=(v,w) \in W} \left( 1 - \hat{\pi}(W[s, v]) \right) c_e + \pi_e \left( F + SP_c(w, t) \right).$

Hasta ahora, en la definición de  $f_{A,\pi}$ , así como en la de  $\hat{f}_{A,\pi}$ , siempre hemos supuesto que el nodo de destino del paseo sobre el cual se calculan es  $t$ . Por conveniencia, extendemos esta definición para un  $(u, v)$ -paseo  $W$ , con  $v$  no necesariamente igual a  $t$ , a través de las fórmulas

$$f_{A,\pi}(W) = \sum_{\substack{e=(x,w) \in W \\ e \notin W[u,x]}} \pi_e \left( \sum_{e' \leq_w e} c_{e'} + F + SP_c(w, t) \right) + (1 - \pi(W)) c(P),$$

$$\hat{f}_{A,\pi}(W) = \sum_{e=(x,w) \in W} \pi_e \left( \sum_{e' \leq_w e} c_{e'} + F + SP_c(w, t) \right) + (1 - \hat{\pi}(W)) c(P).$$

En este caso,  $f_{A,\pi}(W)$  corresponde al costo esperado de  $W$  tomando en cuenta que si el seguidor es inspeccionado en algún arco  $e = (x, w) \in W$ , entonces paga la multa  $F$  y se desvía hacia  $t$  por un  $(w, t)$ -shortest path, y si no es inspeccionado, sólo recorre  $W$ . Claramente, cuando  $v = t$  esta definición coincide la definición que teníamos para el costo esperado de un seguidor adaptativo. Por su parte,  $\hat{f}_{A,\pi}(W)$  corresponde a la extensión análoga para este caso. Bajo las definiciones anteriores, es directo verificar la siguiente propiedad de descomposición para  $\hat{f}_{A,\pi}$ :

**Propiedad 2.7.** Sea  $u \in V$  y  $W \in \mathcal{W}_{ut}$ . Entonces, para todo  $v \in V(W)$ ,

$$\hat{f}_{A,\pi}(W) = \hat{f}_{A,\pi}(W[u, v]) - \hat{\pi}(W[u, v])c(W[v, t]) + \hat{f}_{A,\pi}(W[v, t]).$$

Los dos lemas siguientes muestran la relación entre  $f_{A,\pi}$  y  $\hat{f}_{A,\pi}$ .

**Lema 2.8.** Sea  $u \in V$  y  $W \in \mathcal{W}_{ut}$ . Entonces,

$$\hat{f}_{A,\pi}(W) - f_{A,\pi}(W) = \sum_{\substack{e=(v,w) \in W \\ e \in W[u,v]}} \pi_e (F + SP_c(w, t) - c(W[w, t])).$$

*Demostración.* De la definición de  $\hat{f}_{A,\pi}(W)$  y  $f_{A,\pi}(W)$ , se tiene:

$$\begin{aligned} \hat{f}_{A,\pi}(W) - f_{A,\pi}(W) &= \sum_{\substack{e=(v,w) \in W \\ e \in W[u,v]}} \pi_e \left( \sum_{e' \leq_W e} c_{e'} + F + \text{SP}_c(w, t) \right) - (\hat{\pi}(P) - \pi(P)) c(W) \\ &= \sum_{\substack{e=(v,w) \in W \\ e \in W[u,v]}} \pi_e \left( \sum_{e' \leq_W e} c_{e'} + F + \text{SP}_c(w, t) \right) - \sum_{\substack{e=(v,w) \in W \\ e \in W[u,v]}} \pi_e c(W) \\ &= \sum_{\substack{e=(v,w) \in W \\ e \in W[u,v]}} \pi_{e_i} (F + \text{SP}_c(w, t) - c(W[w, t])). \end{aligned}$$

□

**Lema 2.9.** Sean  $u \in V$  y  $W \in \mathcal{W}_{ut}$ . Si  $c(W) \leq F + \text{SP}_c(u, t)$  ó  $\hat{\pi}(W) \leq 1$ , entonces  $\hat{f}_{A,\pi}(W) \geq \min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ .

*Demostración.* Sean  $u \in V$  y  $W \in \mathcal{W}_{ut}$ . Supongamos que  $c(W) \leq F + \text{SP}_c(u, t)$ . Entonces  $c(W[v, t]) \leq F + \text{SP}_c(v, t)$  para todo  $v \in V(P)$  (de lo contrario, sumando  $c(W[u, v])$  a ambos lados se llega a una contradicción). Del Lema 2.8, se obtiene que  $\hat{f}_{A,\pi}(W) \geq f_{A,\pi}(W)$ . Además, del Lema 2.5, existe  $P' \in \mathcal{P}_{ut}$  tal que  $f_{A,\pi}(W) \geq f_{A,\pi}(P')$ , con lo cual  $\hat{f}_{A,\pi}(W) \geq \min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ .

Ahora supongamos que  $\hat{\pi}(W) \leq 1$ . De la parte anterior, podemos suponer que  $c(W) > F + \text{SP}_c(u, t)$ . Entonces,

$$\begin{aligned} \hat{f}_{A,\pi}(W) &= \sum_{e=(v,w) \in W} \left( \pi_e \left( \sum_{e' \leq_W e} c_{e'} + F + \text{SP}_c(w, t) \right) \right) + (1 - \pi(W)) c(W) \\ &\geq \hat{\pi}(W) (F + \text{SP}_c(u, t)) + (1 - \hat{\pi}(W)) c(W) \\ &\geq F + \text{SP}_c(u, t), \end{aligned}$$

y como  $F + \text{SP}_c(u, t)$  es al menos el costo esperado de cualquier  $(u, t)$ -shortest path, se tiene que  $\hat{f}_{A,\pi}(W) \geq \min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ . □

La siguiente propiedad, cuya demostración omitiremos por ser consecuencia directa del Lema 2.9, permite, dado  $u \in V$ , reducir el problema de minimizar  $f_{A,\pi}$  sobre todos los caminos  $P \in \mathcal{P}_{ut}$  a minimizar  $\hat{f}_{A,\pi}$  sobre todos los paseos  $W \in \mathcal{W}_{ut}$  que cumplan  $\hat{\pi}(W) \leq 1$ .

**Propiedad 2.10.** Para todo  $u \in V$ ,

$$\min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P) = \min_{\substack{W \in \mathcal{W}_{ut} \\ \hat{\pi}(W) \leq 1}} \hat{f}_{A,\pi}(W).$$

El hecho de poder reducirnos a trabajar con  $\hat{f}_{A,\pi}$  nos permite explorar propiedades de suboptimalidad para esta función, de las cuales  $f_{A,\pi}$  carece. Las propiedades de suboptimalidad suelen ser útiles en el diseño de algoritmos eficientes para encontrar caminos mínimos en grafos.

**Propiedad 2.11.** Sea  $u \in V$  y  $P^*$  un  $(u, t)$ -camino tal que  $f_{A,\pi}(P^*) = \min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ . Entonces, para todo  $v \in V(P^*)$ ,

$$-\pi(P^*[u, v])c(P^*[v, t]) + f_{A,\pi}(P^*[v, t]) = \min_{\substack{W \in \mathcal{W}_{vt} \\ \hat{\pi}(W) \leq 1 - \pi(P^*[u, v])}} -\pi(P^*[u, v])c(W) + \hat{f}_{A,\pi}(W).$$

*Demostración.* Denotemos  $P^* = (e_1, \dots, e_k)$  con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = u$  y  $v_{k+1} = t$ . Para el nodo  $v_1$ ,  $\pi(P^*[v_1, v_1]) = 0$  y el resultado se tiene por la Propiedad 2.10. Para  $v_i$  con  $i > 1$ , supongamos que la propiedad es falsa. Luego, existe un paseo  $W \in \mathcal{W}_{v_i t}$ , con  $\hat{\pi}(W) \leq 1 - \pi(P^*[u, v_i])$  tal que

$$-\pi(P^*[u, v_i])c(W) + \hat{f}_{A,\pi}(W) < -\pi(P^*[u, v_i])c(P^*[v_i, t]) + f_{A,\pi}(P^*[v_i, t]).$$

Sumando a ambos lados  $\hat{f}_{A,\pi}(P^*[u, v_i])$ , queda

$$\hat{f}_{A,\pi}(P^*[u, v_i] \circ W) < f_{A,\pi}(P^*)$$

(ver Propiedad 2.7). Luego, como el paseo  $P^*[u, v_i] \circ W$  cumple que  $\hat{\pi}(P^*[u, v_i] \circ W) = \pi(P^*[u, v_i]) + \hat{\pi}(W) \leq 1$ , de la Propiedad 2.10, se concluye que

$$\min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P) = \min_{\substack{W \in \mathcal{W}_{ut} \\ \hat{\pi}(W) \leq 1}} \hat{f}_{A,\pi}f(W) \leq \hat{f}_{A,\pi}(P^*[u, v_i] \circ W) < f_{A,\pi}(P^*),$$

lo que contradice la optimalidad de  $P^*$ . □

A continuación, buscamos hacer uso de la Propiedad 2.11 con el fin de adaptar el algoritmo para grafos acíclicos estudiado en la sección 2.1 al contexto de grafos generales. Al no contar con un orden topológico sobre los vértices, no es posible calcular las etiquetas funcionales  $R_u(q)$  de manera recursiva. Para solucionar esto, proponemos un algoritmo de tipo *Bellman-Ford*, el cual maneja una aproximación de estas etiquetas que mejora en cada iteración. Para ello, inicializamos la función  $R_t(q)$  como

$$R_t(q) = \begin{cases} 0 & \text{si } 0 \leq q \leq 1 \\ +\infty & \text{si } q > 1. \end{cases}$$

El valor  $R_t(q) = \infty$  cuando  $q > 1$  se hace necesario para descartar los paseos que llegan a  $t$  con una probabilidad de inspección irrealizable. Para el resto de los nodos, actualizaremos su etiqueta funcional propagándola a través de cada arco  $e = (v, w)$  mediante la fórmula

$$R_v(q) \leftarrow \min\{R_v(q), (1 - q)c_e + \pi_e(F + \text{SP}_c(w, t)) + R_w(q + \pi_e)\} \quad \forall q \geq 0.$$

El algoritmo resultante, que calcula el valor óptimo del problema  $\text{FMP}_A$  en grafos generales, se presenta a continuación:

---

**Algoritmo 2** Algoritmo para  $FMP_A$  en grafos generales.

---

- 1: Calcular  $SP_c(u, t)$  para todo  $u \in V$ .
  - 2: Sea  $R_t : \mathbb{R}_+ \mapsto \mathbb{R} \cup \{\infty\}$  definida por  $R_t(q) = \begin{cases} 0 & \text{si } q \leq 1 \\ \infty & \text{si } q > 1. \end{cases}$
  - 3: Sea  $R_u : \mathbb{R}_+ \mapsto \mathbb{R} \cup \{\infty\}$  definida por  $R_u(q) = \infty$  para todo  $u \in V \setminus \{t\}$ .
  - 4: **for**  $i = 1, \dots, n - 1$  **do**
  - 5:     **for all**  $e = (v, w) \in E$  **do**
  - 6:          $R_v(q) \leftarrow \min\{R_v(q), (1 - q)c_e + \pi_e(F + SP_c(w, t)) + R_w(q + \pi_e)\}$  para todo  $q \geq 0$ .
  - 7: Retornar  $R_s(0)$ .
- 

En el Algoritmo 2, diremos que ocurre una relajación del arco  $e = (v, w)$  cada vez que el algoritmo actualiza el valor de  $R_v$  a partir de  $R_w$  al visitar dicho arco. Denotaremos por una iteración a la secuencia de relajaciones sucesivas de todos los arcos del grafo que ocurre en cada ejecución del ciclo *for* asociado a la variable  $i$  en la línea 4. Los siguientes resultados tienen como objetivo demostrar la correctitud del algoritmo propuesto.

**Propiedad 2.12.** En el Algoritmo 2, después de cada relajación, se cumple que

$$R_u(q) \geq \min_{\substack{W \in \mathcal{W}_{ut} \\ \hat{\pi}(W) \leq 1 - q}} \left\{ -qc(W) + \hat{f}_{A,\pi}(W) \right\} \quad \forall q \geq 0 \quad \forall u \in V. \quad (2.2)$$

*Demostración.* Por inducción sobre el número de relajaciones. Al inicializar las etiquetas funcionales  $R_u(q)$ ,  $u \in V$ , la desigualdad (2.2) se cumple trivialmente. Supongamos ahora que ésta se sigue cumpliendo a lo largo de las primeras  $k - 1$  relajaciones, y veamos que se preserva después de la  $k$ -ésima. Digamos que esta relajación ocurre al visitar el arco  $e = (v, w)$ . Llamemos  $\bar{R}_v$  a la etiqueta funcional asociada a  $v$  al inicio de la  $k$ -ésima relajación, y  $R_v$  a la que se obtiene al final de ésta. Notemos que  $\bar{R}_v$  es la única función que puede cambiar después de esta relajación, de modo que

$$R_v(q) = \min \left\{ \bar{R}_v(q), (1 - q)c_e + \pi_e(F + SP_c(w, t)) + R_w(q + \pi_e) \right\} \quad \forall q \geq 0.$$

De la hipótesis de inducción se tiene que  $\bar{R}_v(q) \geq \min_{\substack{W \in \mathcal{W}_{v,t} \\ \hat{\pi}(W) \leq 1 - q}} \left\{ -qc(W) + \hat{f}_{A,\pi}(W) \right\}$  para todo  $q \geq 0$ , por lo cual basta probar que

$$(1 - q)c_e + \pi_e(F + SP_c(w, t)) + R_w(q + \pi_e) \geq \min_{\substack{W \in \mathcal{W}_{v,t} \\ \hat{\pi}(W) \leq 1 - q}} \left\{ -qc(W) + \hat{f}_{A,\pi}(W) \right\} \quad \forall q \geq 0.$$

Esta desigualdad es directa para todo  $q \geq 0$  tal que  $R_w(q + \pi_e) = \infty$ . Consideremos entonces  $q \geq 0$  tal que  $R_w(q + \pi_e) < \infty$ . Luego, de la hipótesis de inducción existe  $W^* \in \mathcal{W}_{wt}$  tal que  $\hat{\pi}(W^*) \leq 1 - q - \pi_e$  y

$$-(q + \pi_e)c(W^*) + \hat{f}_{A,\pi}(W^*) = \min_{\substack{W \in \mathcal{W}_{v,t} \\ \hat{\pi}(W) \leq 1 - q - \pi_e}} \left\{ -(q + \pi_e)c(W) + \hat{f}_{A,\pi}(W) \right\}.$$

Con esto,

$$\begin{aligned}
& (1 - q)c_e + \pi_e(F + \text{SP}_c(w, t)) + R_w(q + \pi_e) \\
& \geq (1 - q)c_e + \pi_e(F + \text{SP}_c(w, t)) - (q + \pi_e)c(W^*) + \hat{f}_{A,\pi}(W^*) \\
& = -q(c_e + c(W^*)) + c_e + \pi_e(F + \text{SP}_c(w, t) - c(W^*)) + \hat{f}_{A,\pi}(W^*) \\
& = -qc(e \circ W^*) + \hat{f}_{A,\pi}(e \circ W^*) \\
& \geq \min_{\substack{W \in \mathcal{W}_{v,t} \\ \hat{\pi}(W) \leq 1-q}} \left\{ -qc(W) + \hat{f}_{A,\pi}(W) \right\}.
\end{aligned}$$

□

**Propiedad 2.13.** Sea  $u \in V$  y  $P^*$  un  $(u, t)$ -camino tal que  $f_{A,\pi}(P^*) = \min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ . Sea  $i \in \{0, \dots, |V(P)| - 1\}$ , y denotemos por  $v$  al  $(i + 1)$ -ésimo nodo de  $P^*$  contado desde  $t$  hacia atrás. Entonces, al finalizar la  $i$ -ésima iteración del Algoritmo 2, se cumple que

$$R_v(\pi(P^*[u, v])) = -\pi(P^*[u, v])c(P^*[v, t]) + f_{A,\pi}(P^*[v, t]).$$

En particular, después de la  $(|V(P)| - 1)$ -ésima iteración, se tiene que  $R_u(0) = \min_{P \in \mathcal{P}_{st}} f_{A,\pi}(P)$ .

*Demostración.* Por inducción sobre el número de iteraciones. En beneficio de la claridad de la demostración, denotemos  $P^* = (e_k, e_{k-1}, \dots, e_1)$ , donde  $e_i = (v_{i+1}, v_i)$ ,  $v_{k+1} = u$  y  $v_1 = t$ . Denotemos también  $\pi_i^* = \pi(P^*[u, v_i])$ . Al inicializar el algoritmo el resultado se tiene trivialmente. Supongamos ahora que el resultado es cierto al final de la  $(i - 1)$ -ésima iteración es decir

$$R_{v_i}(\pi_i^*) = -\pi_i^*c(P^*[v_i, t]) + f_{A,\pi}(P^*[v_i, t]),$$

y veamos que también lo es al final de la iteración  $i$ . De la Propiedad 2.12, al finalizar la  $i$ -ésima iteración se cumple en particular que

$$\begin{aligned}
R_{v_{i+1}}(\pi_{i+1}^*) & \geq \min_{\substack{W \in \mathcal{W}_{v_{i+1},t} \\ \hat{\pi}(W) \leq 1 - \pi_{i+1}^*}} \left\{ -\pi_{i+1}^*c(W) + \hat{f}_{A,\pi}(W) \right\} \\
& = -\pi_{i+1}^*c(P^*[v_{i+1}, t]) + f_{A,\pi}(P^*[v_{i+1}, t]),
\end{aligned}$$

donde la última igualdad es por la Propiedad 2.11. Durante la siguiente iteración, cuando el algoritmo relaja el arco  $e_i = (v_{i+1}, v_i)$ , notando que  $\pi_{i+1}^* + \pi_{e_i} = \pi_i^*$ , de la definición de  $R_{v_{i+1}}(q)$  y usando la hipótesis de inducción se obtiene

$$\begin{aligned}
R_{v_{i+1}}(\pi_{i+1}^*) & \leq (1 - \pi_{i+1}^*)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_i, t)) + R_{v_i}(\pi_{i+1}^* + \pi_{e_i}) \\
& = (1 - \pi_{i+1}^*)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_i, t)) + R_{v_i}(\pi_i^*) \\
& = (1 - \pi_{i+1}^*)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_i, t)) - \pi_i^*c(P^*[v_i, t]) + f_{A,\pi}(P^*[v_i, t]) \\
& = (1 - \pi_{i+1}^*)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_i, t)) - (\pi_{i+1}^* + \pi_{e_i})c(P^*[v_i, t]) + f_{A,\pi}(P^*[v_i, t]) \\
& = -\pi_{i+1}^*(c_{e_i} + c(P^*[v_i, t])) + c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_i, t)) - \pi_{e_i}c(P^*[v_i, t]) \\
& \quad + f_{A,\pi}(P^*[v_i, t]) \\
& = -\pi_{i+1}^*c(e_i \circ P^*[v_i, t]) + f_{A,\pi}(e_i \circ P^*[v_i, t]) \\
& = -\pi_{i+1}^*c(P^*[v_{i+1}, t]) + f_{A,\pi}(P^*[v_{i+1}, t]),
\end{aligned}$$

de donde se concluye que  $R_{v_{i+1}}(\pi_{i+1}^*) = -\pi_{i+1}^* c(P^*[v_{i+1}, t]) + f_{A,\pi}(P^*[v_{i+1}, t])$ . De acuerdo a lo anterior, al finalizar la  $i$ -ésima iteración se tiene que

$$R_u(0) = f_{A,\pi}(P^*) = \min_{P \in \mathcal{P}_{st}} f(P).$$

□

*Análisis de complejidad.* Para implementar el Algoritmo 2, es necesario almacenar las pendientes, coeficientes de posición y puntos de quiebre que describen a cada una de las funciones  $R_u(q)$ . Esto puede impactar negativamente en la eficiencia del algoritmo, pues el hecho que  $R_t(q) = \infty$  cuando  $q > 1$  puede inducir que  $R_s(q)$  tenga una cantidad exponencial de puntos de quiebre, como se muestra en el siguiente ejemplo:

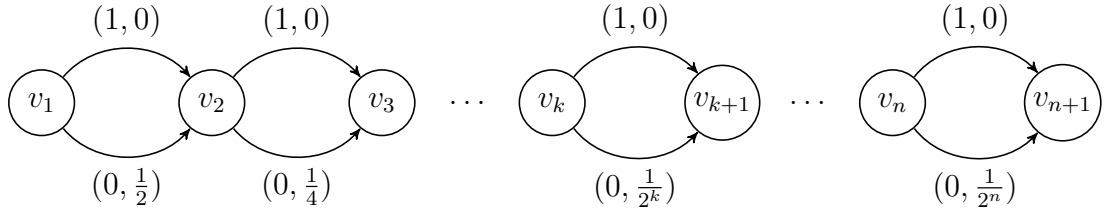


Figura 2.2: El ejemplo muestra que  $R_s(q)$  puede tener una cantidad exponencial de puntos de quiebre.

Consideremos el ejemplo de la Figura 2.2, para  $s = v_1$ ,  $t = v_{n+1}$  y un valor de la multa  $F = 1$ . Cada  $(v_k, v_{n+1})$ -camino puede ser codificado a través de un número binario de  $n$  dígitos, donde los primeros  $k - 1$  dígitos son iguales a 0, y para  $k \leq j \leq n$ , el dígito  $j$ -ésimo vale 1 si el  $j$ -ésimo arco del camino es el de arriba y 0 si es el de abajo. Para  $i = 0, \dots, 2^{n+1-k} - 1$ , denotemos por  $P_{k,i} = (x_{k,i}^1, \dots, x_{k,i}^n)$  al  $(v_k, v_{n+1})$ -camino cuya codificación representa al entero  $i$ , es decir  $x_{k,i}^j = 0$  para  $j < k$  y  $\sum_{j=k}^n x_{k,i}^j 2^j = i$ . Sea además  $\pi(P_{k,i}) = \sum_{j=k}^n (1 - x_{k,i}^j) \frac{1}{2^j}$  la probabilidad de inspección del camino  $P_{k,i}$ . Definiendo también  $\pi(P_{k,-1}) = 1$  y  $\pi(P_{k,2^{n+1-k}}) = 0$ , se tiene que

$$0 = \pi(P_{k,2^{n+1-k}}) \leq \dots \leq \pi(P_{k,i+1}) \leq \pi(P_{k,i}) \leq \dots \leq \pi(P_{k,-1}) = 1.$$

Los siguientes dos lemas son fáciles de verificar:

**Lema 2.14.** Para todo  $k = 1, \dots, n$  e  $i = 0, \dots, 2^{n-k}$ ,  $\pi(P_{k,i}) = \frac{1}{2^k} + \pi(P_{k+1,i})$  y  $\pi(P_{k+1,0}) \leq \frac{1}{2^k}$ .

**Lema 2.15.** Para todo  $k = 1, \dots, n+1$  y  $q \in [0, 1]$ ,  $R_{v_k}(q) \leq (1 - q) + \sum_{j=k}^n \frac{1}{2^j}$ .

**Propiedad 2.16.** Sea  $k \in \{1, \dots, n+1\}$  fijo. Entonces,

$$R_{v_k}(q) = \begin{cases} f_{A,\pi}(P_{k,0}) & \text{si } q = 0 \\ -qc(P_{k,i}) + f_{A,\pi}(P_{k,i}) & \text{si } q \in (1 - \pi(P_{k,i-1}), 1 - \pi(P_{k,i})], i = 0, \dots, 2^{n+1-k} \\ \infty & \text{si } q > 1. \end{cases}$$



*Demostración.* Por inducción sobre  $k$ . Para  $k = n + 1$  es directo de la inicialización de  $R_{v_{n+1}}(q)$ . Supongamos que la propiedad es cierta para  $k + 1$  y probemos que se tiene para  $k$ . Comencemos en primer lugar con  $q = 0$ . En este caso se tiene que  $q \leq 1 - \pi(P_{k+1,0})$ . Luego, de la hipótesis de inducción se tiene que

$$\begin{aligned} R_{v_k}(0) &= \min \left\{ 1 + R_{v_{k+1}}(0), \frac{1}{2^k} + R_{v_{k+1}} \left( \frac{1}{2^k} \right) \right\} \\ &= \min \left\{ 1 + f_{A,\pi}(P_{k+1,0}), \frac{1}{2^k} - \frac{1}{2^k} c(P_{k+1,0}) + f_{A,\pi}(P_{k+1,0}) \right\} \\ &= \frac{1}{2^k} - \frac{1}{2^k} c(P_{k+1,0}) + f_{A,\pi}(P_{k+1,0}) \\ &= f_{A,\pi}(0 \circ P_{k+1,0}) \\ &= f_{A,\pi}(P_{k,0}). \end{aligned}$$

Consideremos ahora  $0 < q \leq 1$ . Sea  $i \in \{0, \dots, 2^{n-k}\}$ . En este caso, se tiene que  $P_{k,i} = 0 \circ P_{k+1,i}$ . Sea  $q \in (1 - \pi(P_{k,i-1}), 1 - \pi(P_{k,i})]$ . Luego, del Lema 2.14, se tiene que  $q + \frac{1}{2^k} \in (1 - \pi(P_{k+1,i-1}), 1 - \pi(P_{k+1,i})]$  y que  $q \leq 1 - \frac{1}{2^k} \leq 1 - \pi(P_{k+1,0})$ . Del Lema 2.15, se tiene que

$$\begin{aligned} \frac{1}{2^k} + R_{v_{k+1}} \left( q + \frac{1}{2^k} \right) &\leq \frac{1}{2^k} + \left( 1 - q - \frac{1}{2^k} \right) + \sum_{j=k+1}^n \frac{1}{2^j} \\ &= (1 - q) + f_{A,\pi}(P_{k+1,0}) \\ &= (1 - q) + R_{v_{k+1}}(q). \end{aligned}$$

De la desigualdad anterior y la hipótesis de inducción sigue que

$$\begin{aligned} R_{v_k}(q) &= \min \left\{ (1 - q) + R_{v_{k+1}}(q), \frac{1}{2^k} + R_{v_{k+1}} \left( q + \frac{1}{2^k} \right) \right\} \\ &= \frac{1}{2^k} + R_{v_{k+1}} \left( q + \frac{1}{2^k} \right) \\ &= \frac{1}{2^k} - \left( q + \frac{1}{2^k} \right) c(P_{k+1,i}) + f_{A,\pi}(P_{k+1,i}) \\ &= -qc(0 \circ P_{k+1,i}) + f_{A,\pi}(0 \circ P_{k+1,i}) \\ &= -qc(P_{k,i}) + f_{A,\pi}(P_{k,i}). \end{aligned}$$

Por otro lado, si  $i \in \{2^{n-k}, \dots, 2^{n+1-k} - 1\}$ , entonces  $P_{k,i} = 1 \circ P_{k+1,i-2^{n-k}}$ . Además,  $q + \frac{1}{2^k} > 1$ , luego

$$\begin{aligned} R_{v_k}(q) &= \min \left\{ (1 - q) + R_{v_{k+1}}(q), \frac{1}{2^k} + R_{v_{k+1}} \left( q + \frac{1}{2^k} \right) \right\} \\ &= (1 - q) + R_{v_{k+1}}(q) \\ &= (1 - q) - qc(P_{k+1,i}) + f_{A,\pi}(P_{k+1,i}) \\ &= -qc(1 \circ P_{k+1,i}) + f_{A,\pi}(1 \circ P_{k+1,i}) \\ &= -qc(P_{k,i}) + f_{A,\pi}(P_{k,i}). \end{aligned}$$

Finalmente, el hecho que  $R_{v_k}(q) = \infty$  para  $q > 1$  es directo de la hipótesis de inducción.  $\square$

De la Propiedad 2.16 se obtiene que la función  $R_{v_1}(q)$  tiene al menos  $2^{n-1}$  puntos de quiebre, pues cada vez que  $i$  es par,  $P_{1,i+1}$  tiene exactamente un 1 más que  $P_{1,i}$  en su codificación. Luego en el punto  $q = 1 - \pi(P_{k,i})$ ,  $R_{v_1}(q)$  experimenta un cambio de pendiente ya que  $c(P_{1,i+1}) - c(P_{1,i}) = 1$ .

## 2.3. Un algoritmo pseudopolinomial para grafos generales

A continuación veremos resultados que nos permitirán diseñar un algoritmo pseudopolinomial para el problema del evasor en grafos generales. Dado  $u \in V$ , la siguiente propiedad permite reducir el problema de minimizar  $f_{A,\pi}$  sobre todos los caminos  $P \in \mathcal{P}_{ut}$  a minimizar  $\hat{f}_{A,\pi}$  sobre un conjunto de paseos  $W \in \mathcal{W}_{ut}$  cuyo costo  $c(W)$  está acotado por  $(n-1)c_{\text{máx}}$ .

**Propiedad 2.17.** Para todo  $u \in V$ ,

$$\min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P) = \min \left\{ \hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{ut}, c(W) \leq \min \{F + \text{SP}_c(u, t), (n-1)c_{\text{máx}}\} \right\}.$$

*Demostración.* Denotemos

$$\widehat{\text{OPT}} := \min \left\{ \hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{ut}, c(W) \leq \min \{F + \text{SP}_c(u, t), (n-1)c_{\text{máx}}\} \right\}.$$

Sea  $P^* \in \mathcal{P}_{ut}$  un  $(u, t)$ -camino tal que  $f_{A,\pi}(P^*) = \min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ . Primero veamos que sin pérdida de generalidad podemos suponer que  $c(P^*) \leq F + \text{SP}_c(u, t)$ . En efecto, se tiene que

$$\pi(P^*) (F + \text{SP}_c(u, t)) + (1 - \pi(P^*))c(P^*) \leq f_{A,\pi}(P^*) \leq F + \text{SP}_c(u, t),$$

de lo cual se obtiene

$$(1 - \pi(P^*))c(P^*) \leq (1 - \pi(P^*)) (F + \text{SP}_c(u, t)).$$

Luego, si  $\pi(P^*) < 1$ , entonces  $c(P^*) \leq F + \text{SP}_c(u, t)$ . Por otra parte, si  $\pi(P^*) = 1$ , entonces  $f_{A,\pi}(P^*) = F + \text{SP}_c(u, t)$ , y luego cualquier  $(u, t)$ -shortest path es también óptimo.

Ahora probaremos la propiedad. Notemos que  $c(P^*) \leq (n-1)c_{\text{máx}}$ , con lo cual  $f_{A,\pi}(P^*) \geq \widehat{\text{OPT}}$ .

Por otro lado, dado  $W \in \mathcal{W}_{ut}$  arbitrario tal que  $c(W) \leq \min \{F + \text{SP}_c(u, t), (n-1)c_{\text{máx}}\}$ , del Lema 2.9 se tiene que  $f_{A,\pi}(P^*) \leq \hat{f}_{A,\pi}(W)$ , por lo tanto,  $f_{A,\pi}(P^*) \leq \widehat{\text{OPT}}$ .  $\square$

En lo que sigue, definimos  $K_t := 0$  y  $K_u := \min\{F + \text{SP}_c(u, t), (n-1)c_{\max}\}$  para todo  $u \in V \setminus \{t\}$ . La siguiente propiedad de suboptimalidad de  $f_A$  será clave para diseñar un algoritmo pseudopolinomial para  $\text{FMP}_A$ :

**Propiedad 2.18.** Sea  $u \in V$  y  $k \in \{0, \dots, K_u\}$ . Supongamos que  $W^* \in \mathcal{W}_{ut}$  es un paseo de costo  $c(W^*) = k$  tal que  $\hat{f}_{A,\pi}(W^*) = \min\{\hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{ut}, c(W) = k\}$ . Entonces, para todo  $v \in V(W^*)$ ,  $\hat{f}_{A,\pi}(W^*[v, t]) = \min\{\hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{vt}, c(W) = c(W^*[v, t])\}$ .

*Demostración.* Denotemos  $W^* = (e_1, \dots, e_k)$  con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = u$  y  $v_{k+1} = t$ . Supongamos que para algún  $v_i$ , con  $i \in \{1, \dots, k+1\}$ , la propiedad es falsa. Luego, existe  $W' \in \mathcal{W}_{v_i t}$  con costo  $c(W') = c(W^*[v_i, t])$  y tal que

$$\hat{f}_{A,\pi}(W') < \hat{f}_{A,\pi}(W^*[v_i, t]). \quad (2.3)$$

Sumando  $\hat{f}_{A,\pi}(W^*[u, v_i]) - \hat{\pi}(W^*[u, v_i])c(W^*[v_i, t])$  a ambos lados, se obtiene

$$\hat{f}_{A,\pi}(W^*[u, v_i] \circ W') < \hat{f}_{A,\pi}(W^*)$$

(ver Propiedad 2.7), y como  $c(W^*[u, v_i] \circ W') = c(W^*[u, v_i]) + c(W^*[v_i, t]) = k$ , la desigualdad anterior contradice la optimalidad de  $W^*$ .  $\square$

**Propiedad 2.19.** Sean  $\phi_u^k$ ,  $u \in V$ ,  $k = 0, \dots, K_u$  etiquetas tales que

$$\phi_u^k = \min\{\hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{ut}, c(W) = k\}$$

para todo  $u \in V$  y  $k = 0, \dots, K_u$ . Sea  $u \in V$  y  $W^* \in \mathcal{W}_{ut}$  con costo  $c(W^*) = k$ . Entonces,  $\hat{f}_{A,\pi}(W^*) = \phi_u^k$  si y sólo si para todo  $e = (v, w) \in W^*$ ,

$$\phi_v^{c(W^*[v, t])} = c_e + \pi_e (F + \text{SP}_c(w, t) - c(W^*[w, t])) + \phi_w^{c(W^*[w, t])}. \quad (2.4)$$

*Demostración.* Denotemos  $W^* = (e_1, \dots, e_k)$  con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = s$  y  $v_{k+1} = t$ . Supongamos que  $\hat{f}_{A,\pi}(W^*) = \phi_u^k$ . Entonces, la Propiedad 2.18 implica que  $\hat{f}_{A,\pi}(W^*[v_i, t]) = \phi_{v_i}^{c(W^*[v_i, t])}$  para todo  $i = 1, \dots, k+1$ . Luego, para todo  $e_i = (v_i, v_{i+1})$ ,

$$\begin{aligned} \phi_{v_i}^{c(W^*[v_i, t])} &= \hat{f}_{A,\pi}(W^*[v_i, t]) \\ &= c_{e_i} + \pi_{e_i} (F + \text{SP}_c(v_{i+1}, t) - c(W^*[v_{i+1}, t])) + \hat{f}_{A,\pi}(W^*[v_{i+1}, t]) \\ &= c_{e_i} + \pi_{e_i} (F + \text{SP}_c(v_{i+1}, t) - c(W^*[v_{i+1}, t])) + \phi_{v_{i+1}}^{c(W^*[v_{i+1}, t])}. \end{aligned}$$

Supongamos ahora que para todo  $e_i = (v_i, v_{i+1})$  se cumple que

$$\phi_{v_i}^{c(W^*[v_i, t])} = c_{e_i} + \pi_{e_i} (F + \text{SP}_c(v_{i+1}, t) - c(W^*[v_{i+1}, t])) + \phi_{v_{i+1}}^{c(W^*[v_{i+1}, t])}.$$

Entonces,

$$\begin{aligned}
\hat{f}_{A,\pi}(W^*) &= \sum_{i=1}^k c_{e_i} + \pi_{e_i} (F + \text{SP}_c(v_{i+1}, t) - c(W^*[v_{i+1}, t])) \\
&= \sum_{i=1}^k \phi_{v_i}^{c(W^*[v_i, t])} - \phi_{v_{i+1}}^{c(W^*[v_{i+1}, t])} \\
&= \phi_{v_1}^{c(W^*[v_1, t])} - \phi_{v_{k+1}}^{c(W^*[v_{k+1}, t])} \\
&= \phi_u^{c(W^*)} - \phi_t^0 \\
&= \phi_u^k.
\end{aligned}$$

□

Basados en las Propiedades 2.18 y 2.19, proponemos un algoritmo de tipo Dijkstra para resolver el problema FMP<sub>A</sub> en grafos generales. En él, cada nodo  $u \in V$  posee un conjunto de etiquetas  $\phi_u^k$ , con  $k = 0, \dots, K_u$ , cuyo valor es una cota superior de  $\min \{ \hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{ut}, c(W) = k \}$  que mejora en cada iteración. El algoritmo mantiene un conjunto  $S$  de pares  $(u, k)$  cuyas etiquetas  $\phi_u^k$  tienen un valor permanente y su complemento  $\bar{S}$ , cuyos pares están asociados a etiquetas por actualizar. Comenzando con  $S = \emptyset$  y  $\bar{S} = \{(u, k) : u \in V, k = 0, \dots, K_u\}$ , en cada iteración, el algoritmo transfiere de  $\bar{S}$  a  $S$  un par  $(w, k) \in \bar{S}$  cuya etiqueta asociada es mínima. Luego, escanea todos los arcos  $e = (v, w) \in \delta^-(w)$  tales que  $c_e + k \leq K_v$ , y si  $\phi_v^{k+c_e} > c_e + \pi_e (F + \text{SP}_c(w, t) - k) + \phi_w^k$ , entonces actualiza  $\phi_v^{k+c_e}$  de modo que  $\phi_v^{k+c_e} = c_e + \pi_e (F + \text{SP}_c(w, t) - k) + \phi_w^k$  y asigna a  $(w, k)$  como sucesor de  $(v, k + c_e)$ . El pseudocódigo se presenta a continuación:

---

**Algoritmo 3** Algoritmo para FMP<sub>A</sub> en grafos generales.

---

- 1: Calcular  $\text{SP}_c(u, t)$  para todo  $u \in V$ .
  - 2: Sean  $K_t \leftarrow 0$ ,  $K_u \leftarrow \min\{F + \text{SP}_c(u, t), (n-1)c_{\max}\}$  para todo  $u \in V \setminus \{t\}$ .
  - 3: Sean  $\phi_t^0 \leftarrow 0$ ,  $\phi_u^k \leftarrow \infty$  para todo  $u \in V \setminus \{t\}$  y  $k = 0, \dots, K_u$ .
  - 4: Sean  $\text{suc}(u, k) \leftarrow \emptyset$  para todo  $u \in V$  y  $k = 0, \dots, K_u$ .
  - 5: Sean  $S \leftarrow \emptyset$ ,  $\bar{S} \leftarrow \{(u, k) : u \in V, k = 0, \dots, K_u\}$ .
  - 6: **while**  $\bar{S} \neq \emptyset$  **do**
  - 7:   Sea  $(w, k) \in \bar{S}$  tal que  $\phi_w^k = \min_{(u, \ell) \in \bar{S}} \phi_u^\ell$ .
  - 8:    $S \leftarrow S \cup \{(w, k)\}$ .
  - 9:    $\bar{S} \leftarrow \bar{S} \setminus \{(w, k)\}$ .
  - 10:   **for all**  $e = (v, w) \in \delta^-(w)$  **do**
  - 11:     **if**  $k + c_e \leq K_v$  **then**
  - 12:       **if**  $\phi_v^{k+c_e} > c_e + \pi_e (F + \text{SP}_c(w, t) - k) + \phi_w^k$  **then**
  - 13:          $\phi_v^{k+c_e} \leftarrow c_e + \pi_e (F + \text{SP}_c(w, t) - k) + \phi_w^k$ .
  - 14:          $\text{suc}(v, k + c_e) \leftarrow (w, k)$ .
  - 15:   Sea  $\ell \in \{0, \dots, K_t\}$  tal que  $\phi_s^\ell = \min_{0 \leq k \leq K_v} \phi_s^k$ .
  - 16:   Sea  $W$  el paseo formado por los sucesores de  $(s, \ell)$ .
  - 17:   Sea  $P$  el  $(s, t)$ -camino directo subyacente a  $W$ .
  - 18:   Retornar  $P$ .
-

El siguiente invariante demuestra la correctitud del Algoritmo 3:

**Propiedad 2.20.** En cada iteración del Algoritmo 3 se cumple:

(i) Para todo  $u \in V$  y  $k = 0, \dots, K_u$  tales que  $(u, k) \in S$ ,

$$\phi_u^k = \min \left\{ \hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{ut}, c(W) = k \right\}.$$

(ii) Para todo  $u \in V$  y  $k = 0, \dots, K_u$  tales que  $(u, k) \in \bar{S}$ ,

$$\phi_u^k = \min \left\{ \hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{ut}, c(W) = k, \forall v \in \text{int}(V(W)), (v, c(W[v, t])) \in S \right\},$$

donde  $\text{int}(V(W))$  denota el conjunto de los nodos internos de  $W$ .

*Observación:* Si  $W = (e_1, \dots, e_k) \in \mathcal{W}_{ut}$ , con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = u$  y  $v_{k+1} = t$ , entonces el conjunto de nodos internos de  $W$  es  $\text{int}(V(W)) = \{v_2, \dots, v_k\}$ .

*Demostración.* Por inducción sobre el número de iteraciones. En una iteración cualquiera, el algoritmo transfiere de  $\bar{S}$  a  $S$  un par  $(w, k)$  tal que  $\phi_w^k = \min_{(u, \ell) \in \bar{S}} \phi_u^\ell$ . Probaremos primero que se preserva (i). De (ii), sabemos que

$$\phi_w^k = \min \left\{ \hat{f}_{A,\pi}(W) : W \in \mathcal{W}_{wt}, c(W) = k, \forall v \in \text{int}(V(W)), (v, c(W[v, t])) \in S \right\},$$

por lo que basta probar que para todo  $W \in \mathcal{W}_{wt}$  con  $c(W) = k$  y tal que alguno de sus nodos internos  $v'$  satisface  $(v', c(W[v', t])) \in \bar{S}$ , se cumple que  $\phi_w^k \leq \hat{f}_{A,\pi}(W)$ . Sea  $W = (e_1, \dots, e_k)$ , con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = w$  y  $v_{k+1} = t$  un paseo con estas características, y denotemos por  $v_j$  al último nodo con la propiedad mencionada en la secuencia de nodos que describe a  $W$ . Luego, el paseo  $W_j := W[v_j, t] \in \mathcal{W}_{v_j t}$  es tal que todos sus nodos internos  $u$  satisfacen  $(u, W_j[u, t]) \in S$ . Luego, de la elección de  $(w, k)$  y de (ii) se tiene que

$$\phi_w^k \leq \phi_{v_j}^{c(W_j)} \leq \hat{f}_{A,\pi}(W_j).$$

Además, como  $c(W) = k \leq K_w \leq F + \text{SP}_c(w, t)$ , se cumple que  $c(W[v_i, t]) \leq F + \text{SP}_c(v_i, t)$  para todo  $j = i, \dots, k + 1$ , luego

$$\hat{f}_{A,\pi}(W_j) \leq \sum_{j=1}^{j-1} c_{e_i} + \pi_{e_i} (F + \text{SP}_c(v_{i+1}, t) - c(W[v_{i+1}, t])) + \hat{f}_{A,\pi}(W_j) = \hat{f}_{A,\pi}(W),$$

por lo tanto,  $\phi_w^k \leq \hat{f}_{A,\pi}(W)$ .

Ahora veamos que se preserva (ii). Después de que el algoritmo etiqueta permanentemente a  $(w, k)$ , las etiquetas de algunos nodos en  $\bar{S} \setminus \{(w, k)\}$  pueden decrecer. Luego de transferir a  $(w, k)$  de  $\bar{S}$  a  $S$ , el algoritmo actualiza las etiquetas  $(v, k + c_e)$  para todo  $e = (v, w) \in \delta^-(w)$  con costo  $c_e \leq K_v - k$  y asigna  $\phi_v^{k+c_e} = c_e + \pi_e (F + \text{SP}_c(w, t) - k) + \phi_w^k$  si  $\phi_v^{k+c_e} > c_e + \pi_e (F + \text{SP}_c(w, t) - k) + \phi_w^k$ . Usando (i), dado  $(u, \ell) \in \bar{S} \setminus \{(w, k)\}$ , el  $(u, t)$ -paseo formado por  $(u, \ell)$  y sus sucesores satisface la Propiedad 2.19 y por lo tanto minimiza  $\hat{f}_{A,\pi}$  entre todos los paseos  $W \in \mathcal{W}_{ut}$  tales que  $c(W) = \ell$  y para todo nodo interno  $u \in \text{int}V(W)$ ,  $(u, c(W[u, t])) \in S$ .  $\square$

*Análisis de complejidad.* A continuación analizaremos la complejidad del Algoritmo 3. El número de etiquetas que éste maneja es a lo más

$$\sum_{u \in V} (K_u + 1) \leq n((n-1)c_{\max} + 1) = O(n^2 c_{\max}).$$

Además, en cada iteración, el algoritmo escoge una etiqueta mínima  $(w, k)$  en  $\bar{S}$ , cuyo cardinal decrece en una unidad con respecto a la iteración anterior. Utilizando *Fibonacci heaps*, podemos actualizar el valor de la etiqueta mínima en tiempo del orden del logaritmo de la cantidad de etiquetas en cada iteración, con lo cual se realiza una cantidad  $O(n^2 c_{\max} \log(n^2 c_{\max}))$  operaciones para seleccionar etiquetas. A continuación, actualiza las etiquetas de los pares  $(v, k)$  con  $e = (v, w) \in \delta^-(w)$  tales que  $k + c_e \leq K_v$ . Esta actualización se realiza a lo más

$$\begin{aligned} \sum_{w \in V} \sum_{k=0}^{K_w} |\{e = (v, w) \in \delta^-(w) : k + c_e \leq K_v\}| &\leq \sum_{w \in V} \sum_{k=0}^{K_w} |\delta^-(w)| \leq m((n-1)c_{\max} + 1) \\ &= O(mnc_{\max}) \end{aligned}$$

veces durante la ejecución del algoritmo. Esto nos da un tiempo total de

$$O(mnc_{\max} + n^2 c_{\max} \log(n^2 c_{\max})),$$

el cual resulta ser pseudopolinomial en el tamaño del *input*.

**Teorema 2.21.** *El problema  $FMP_A$  en grafos generales puede ser resuelto en tiempo  $O(mnc_{\max} + n^2 c_{\max} \log(n^2 c_{\max}))$ .*

## 2.4. Una algoritmo fuertemente polinomial para la distribución uniforme

Supongamos que la distribución  $\pi$  es tal que cada una de sus componentes puede asumir sólo dos valores posibles: o bien vale 0, o bien vale  $r$ , para un cierto  $r \in \mathbb{Q} \cap [0, 1]$  fijo. En este caso podemos aprovechar esta estructura para diseñar un algoritmo similar al Algoritmo 3 para calcular  $\min_{\substack{W \in \mathcal{W}_{vt} \\ \hat{\pi}(W) \leq 1}} \hat{f}_{A,\pi}(W)$ , que por la Propiedad 2.10, es igual a  $\min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ . Para

ello, notemos en primer lugar que en este caso el mayor valor posible de  $\hat{\pi}(W)$  sujeto a que  $\hat{\pi}(W) \leq 1$  es  $\lfloor \frac{1}{r} \rfloor r$ . Siguiendo el diseño del Algoritmo 3, utilizamos etiquetas  $\phi_v^p$ , con  $v \in V$  y  $p \in \{0, r, 2r, \dots, \lfloor \frac{1}{r} \rfloor r\}$ . Cada etiqueta  $\phi_v^p$  representará una cota superior para el valor  $\min_{\substack{W \in \mathcal{W}_{sv} \\ \hat{\pi}(W) = p}} \hat{f}_{A,\pi}(W)$ . Comenzando con  $\phi_s^0 = 0$  y  $\phi_v^p = \infty$  para todo  $(v, p) \in$

$V \times \{0, r, \dots, \lfloor \frac{1}{r} \rfloor r\} \setminus \{(s, 0)\}$ , definimos los conjuntos  $S = \emptyset$  y  $\bar{S} = V \times \{0, r, \dots, \lfloor \frac{1}{r} \rfloor r\}$ . A continuación, en cada iteración se seleccionará una etiqueta  $(v, p) \in \bar{S}$  tal que  $\phi_v^p = \min_{(w, \ell) \in \bar{S}} \phi_w^\ell$ , para transferirla de  $\bar{S}$  a  $S$  y escanear los arcos  $e = (v, w) \in \delta^+(v)$  tales que  $p + \pi_e \leq 1$ , y si  $\phi_w^{p+\pi_e} > (1-p)c_e + \pi_e(F + SP_c(w, t)) + \phi_v^p$ , hacemos la actualización

$\phi_w^{p+\pi_e} = (1-p)c_e + \pi_e(F + \text{SP}_c(w, t)) + \phi_v^p$  y asignamos a  $(w, p + \pi_e)$  como sucesor de  $(v, p)$ .

---

**Algoritmo 4** Algoritmo para  $\text{FMP}_A$  en grafos generales cuando  $\pi_e \in \{0, r\} \forall e \in E$ .

---

- 1: Calcular  $\text{SP}_c(u, t)$  para todo  $u \in V$ .
  - 2: Sean  $\phi_s^0 \leftarrow 0, \phi_u^p \leftarrow \infty$  para todo  $u \in V \setminus \{s\}$  y  $p = 0, r, \dots, \lfloor \frac{1}{r} \rfloor r$ .
  - 3: Sean  $\text{suc}(u, k) \leftarrow \emptyset$  para todo  $u \in V$  y  $p = 0, r, \dots, \lfloor \frac{1}{r} \rfloor r$ .
  - 4: Sean  $S \leftarrow \emptyset, \bar{S} \leftarrow V \times \{0, r, \dots, \lfloor \frac{1}{r} \rfloor r\}$ .
  - 5: **while**  $\bar{S} \neq \emptyset$  **do**
  - 6:     Sea  $(v, p) \in \bar{S}$  tal que  $\phi_v^p = \min_{(w, \ell) \in \bar{S}} \phi_w^\ell$ .
  - 7:      $S \leftarrow S \cup \{(v, p)\}$ .
  - 8:      $\bar{S} \leftarrow \bar{S} \setminus \{(v, p)\}$ .
  - 9:     **for all**  $e = (v, w) \in \delta^+(v)$  **do**
  - 10:         **if**  $p + \pi_e \leq 1$  **then**
  - 11:             **if**  $\phi_w^{p+\pi_e} > (1-p)c_e + \pi_e(F + \text{SP}_c(w, t)) + \phi_v^p$  **then**
  - 12:                  $\phi_w^{p+\pi_e} \leftarrow (1-p)c_e + \pi_e(F + \text{SP}_c(w, t)) + \phi_v^p$ .
  - 13:                  $\text{suc}(v, p) \leftarrow (w, p + \pi_e)$ .
  - 14:     Sea  $\ell \in \{0, r, \dots, \lfloor \frac{1}{r} \rfloor r\}$  tal que  $\phi_s^\ell = \min_{0 \leq p \leq \lfloor \frac{1}{r} \rfloor r} \phi_s^p$ .
  - 15:     Sea  $W$  el paseo formado por los sucesores de  $(s, \ell)$
  - 16:     **if**  $c(W) > F + \text{SP}_c(s, t)$  **then**
  - 17:         Sea  $P$  un  $(s, t)$ -shortest path.
  - 18:         Retornar  $P$ .
  - 19:     Sea  $P$  el  $(s, t)$ -camino directo subyacente a  $W$ .
  - 20:     Retornar  $P$ .
- 

De manera análoga al Algoritmo 3, se puede probar la correctitud del algoritmo propuesto, la cual resulta como consecuencia de la siguiente propiedad de suboptimalidad, cuya demostración sigue el mismo esquema que la de la Propiedad 2.18:

**Propiedad 2.22.** Sea  $u \in V$  y  $p \in \{0, r, 2r, \dots, \lfloor \frac{1}{r} \rfloor r\}$ . Supongamos que  $W^* \in \mathcal{W}_{su}$  es un paseo con  $\hat{\pi}(W^*) = p$  tal que  $\hat{f}_{A, \pi}(W^*) = \min \left\{ \hat{f}_{A, \pi}(W) : W \in \mathcal{W}_{su}, \hat{\pi}(W) = p \right\}$ . Entonces, para todo  $v \in V(W^*)$ ,  $\hat{f}_{A, \pi}(W^*[s, v]) = \min \left\{ \hat{f}_{A, \pi}(W) : W \in \mathcal{W}_{sv}, \hat{\pi}(W) = \hat{\pi}(W^*[s, v]) \right\}$ .

Una análisis de complejidad similar al del Algoritmo 3 nos permite afirmar que el número de etiquetas que maneja este algoritmo es  $O\left(\frac{n}{r}\right)$ , mientras que el número de actualizaciones de etiquetas es  $O\left(\frac{m}{r}\right)$ , lo cual mediante el uso de *Fibonacci Heaps* nos arroja una complejidad total de

$$O\left(\frac{m}{r} + \frac{n}{r} \log\left(\frac{n}{r}\right)\right).$$

En particular, si el vector  $\pi$  corresponde a la distribución uniforme, entonces  $\pi_e = r = \frac{1}{m}$  para todo  $e \in E$ , con lo cual el problema del evasor adaptativo puede ser resuelto en tiempo fuertemente polinomial.

**Teorema 2.23.** *Si  $\pi$  es la distribución uniforme, el problema  $\text{FMP}_A$  en grafos generales puede ser resuelto en tiempo  $O(m^2 + mn \log(mn))$ .*

## 2.5. Un FPTAS para grafos generales

En esta sección diseñaremos un FPTAS (*Fully Polynomial Approximation Scheme*) para el problema  $\text{FMP}_A$  en grafos generales. Un algoritmo  $\mathcal{A}$  es un FPTAS para un problema de optimización ( $P$ ) si dado un *input*  $I$  para ( $P$ ) y  $\varepsilon > 0$ ,  $\mathcal{A}$  encuentra en tiempo polinomial en el tamaño de  $I$  y  $1/\varepsilon$  una solución  $S$  para  $I$  que satisface

$$|val(I) - val(S)| \leq \varepsilon val(I),$$

donde  $val(I)$  es el valor óptimo de una solución para  $I$ .

De la desigualdad (1.3), si  $P^* \in \mathcal{P}_{st}$  es un camino tal que  $f_{A,\pi}(P^*) = \min_{P \in \mathcal{P}_{ut}} f_{A,\pi}(P)$ , entonces

$$0 \leq f_{A,\pi}(P^*) \leq F + \text{SP}_c(s, t) \leq F + (n-1)c_{\text{máx}}.$$

Con el fin de manejar una aproximación de  $f_{A,\pi}(P^*)$ , dividiremos el intervalo  $[0, F + (n-1)c_{\text{máx}}]$  en potencias de  $(1 + \frac{\varepsilon}{n})$ . Sea  $K = \left\lceil \log_{1+\frac{\varepsilon}{n}}(F + (n-1)c_{\text{máx}}) \right\rceil$ . Definimos los intervalos  $I_0 = [0, 1)$ ,  $I_k = \left[ \left(1 + \frac{\varepsilon}{n}\right)^{k-1}, \left(1 + \frac{\varepsilon}{n}\right)^k \right)$  para  $k = 1, \dots, K-1$  e  $I_K = \left[ \left(1 + \frac{\varepsilon}{n}\right)^{K-1}, \left(1 + \frac{\varepsilon}{n}\right)^K \right]$ . Cada intervalo  $I_k$ ,  $k \in \{0, \dots, K\}$  contendrá una lista de etiquetas  $(L_v^k)_{v \in V}$ , donde para cada  $v \in V$ ,  $L_v^k$  será de la forma  $(f, q, P)$ , con  $f \in I_k$  el costo esperado de un  $(s, v)$ -camino,  $q$  la probabilidad de inspección de este camino y  $P$  la lista ordenada de arcos que lo describen. Comenzando con  $L_s^0 = (0, 0, \emptyset)$ , en una iteración cualquiera el algoritmo recorrerá cada arco  $e = (v, w) \in E$  y propagará las etiquetas  $L_v^k$ ,  $k = 0, \dots, K$  a través de  $e$  generando etiquetas  $L_w^k$  de la forma  $(f', q', P')$ , con  $f' = f + (1 - q')c_e + \pi_e(F + \text{SP}_c(w, t))$ ,  $q' = q + \pi_e$  y  $P' = P \circ \{e\}$ . La propagación de  $(f, q, P)$  a través de  $e = (v, w)$  sólo será admisible si  $w \notin V(P)$ . Además, si la propagación de una etiqueta a través de un arco da lugar a dos etiquetas  $(f', q', P')$  y  $(f'', q'', P'')$  para un mismo nodo tales  $f'$  y  $f''$  pertenecen a un mismo intervalo  $I_k$ , se descartará aquella cuyo camino asociado contenga la menor probabilidad de inspección. Formalmente, el algoritmo es el siguiente:



---

**Algoritmo 5** FPTAS para  $FMP_A$  en grafos generales.
 

---

```

1: Calcular  $SP_c(v, t)$  para todo  $v \in V$ .
2: Sea  $K \leftarrow \left\lceil \log_{1+\frac{\varepsilon}{n}} (F + (n-1)c_{\max}) \right\rceil$ .
3: Sean  $L_v^k \leftarrow \emptyset$  para todo  $(v, k) \in V \times \{0, \dots, K\}$ .
4: Sea  $L_s^0 \leftarrow (0, 0, \emptyset)$ .
5: for  $i = 1, \dots, (n-1)$  do
6:   for all  $e = (v, w) \in E$  do
7:     for  $k = 0, \dots, K$  do
8:       if  $L_v^k \neq \emptyset$  then
9:         PUSH( $L_v^k, e$ ).
10: Sea  $(f^*, q^*, P^*) \in \operatorname{argmin} \{f : (f, q, P) \in L_t^k \text{ para algún } k\}$ .
11: Retornar  $P^*$ .

12: procedure PUSH( $L = (f, q, P), e = (v, w)$ )
13:   if  $w \notin V(P)$  then
14:     Sea  $f' \leftarrow f + (1-q)c_e + \pi_e (M + SP_c(w, t))$ .
15:     Sea  $q' \leftarrow q + \pi_e$ .
16:     Sea  $P' \leftarrow P \circ e$ .
17:     Sea  $k \leftarrow \min \left\{ \ell \in \mathbb{Z}_+ : f' < \left(1 + \frac{\varepsilon}{n}\right)^\ell \right\}$ .
18:     if  $L_w^k = \emptyset$  then
19:        $L_w^k \leftarrow (f', q', P')$ .
20:     else
21:       Sea  $(f'', q'', P'') \leftarrow L_w^k$ .
22:       if  $q' > q''$  then
23:          $L_w^k \leftarrow (f', q', P')$ .

```

---

A partir de la discusión anterior, se obtiene directamente el siguiente lema:

**Lema 2.24.** *Si el Algoritmo 5 crea una etiqueta  $(f, q, P)$  en un nodo  $v \in V$ , entonces  $P$  es un  $(s, v)$ -camino con  $f_{A, \pi}(P) = f$  y  $\pi(P) = q$ .*

En lo que sigue, sea  $P^* = (e_1, \dots, e_k)$ , con  $e_i = (v_i, v_{i+1})$ ,  $v_1 = s$  y  $v_{k+1} = t$  un  $(s, t)$ -camino tal que  $f_{A, \pi}(P^*) = \min_{P \in \mathcal{P}_{st}} f_{A, \pi}(P)$  y  $c(P^*) \leq F + SP_c(s, t)$  (la existencia de un camino óptimo de costo menor o igual a  $F + SP_c(s, t)$  se argumenta en la demostración de la Propiedad 2.17). Denotemos por  $f_i^* = f_{A, \pi}(P^*[s, v_i])$  y  $q_i^* = \pi(P^*[s, v_i])$ . Además supondremos sin pérdida de generalidad que para todo  $e \in E$ ,  $F\pi_e \geq 1$  si  $\pi_e > 0$ . Esto se puede lograr escalando  $c$  y  $F$  por un factor  $\alpha \in \mathbb{Z}_+$  adecuado. Notemos que al escalar  $M$  y  $c$  por  $\alpha$ , el costo esperado de todo camino se amplifica exactamente por  $\alpha$ . Más adelante veremos que podemos escoger  $\alpha$  de manera que no afecte drásticamente el tiempo de ejecución del algoritmo.

**Propiedad 2.25.** Al final de la iteración  $i$ -ésima del Algoritmo 5, existe una etiqueta  $(f_i, q_i, P_i)$  en el nodo  $v_i$  tal que

$$f_i \leq \left(1 + \frac{\varepsilon}{n}\right)^i f_i^* - (q_i^* - q_i)_+ \cdot c(P^*[v_i, t]).$$

En particular, al final de la  $(n - 1)$ -ésima iteración, existe una etiqueta  $(f, q, P)$  en  $t$  tal que

$$f_{A,\pi}(P) \leq \left(1 + \frac{\varepsilon}{n}\right)^{n-1} f_{A,\pi}(P^*).$$

*Observación:* Para  $x \in \mathbb{R}$ ,  $x_+$  denota la parte positiva de  $x$ , es decir,  $x_+ = \max\{x, 0\}$ .

*Demostración.* Por inducción sobre  $i$ . El caso  $i = 0$  es trivial por la inicialización del algoritmo. Supongamos que la propiedad es cierta para la iteración  $i$ -ésima y veamos que se cumple en la iteración  $i + 1$ . Consideremos  $e_i = (v_i, v_{i+1}) \in P^*$ . Supongamos primero que  $v_{i+1} \notin P_i$ . En este caso, la etiqueta  $(f_i, q_i, P_i)$  se propaga a través de  $e_i$ , obteniéndose una etiqueta  $(f_{i+1}, q_{i+1}, P_{i+1})$  con  $q_{i+1} = q_i + \pi_{e_i}$ ,  $P_{i+1} = P_i \circ e_i$  y

$$\begin{aligned} f_{i+1} &= f_i + (1 - q_i)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_{i+1}, t)) \\ &\leq \left(1 + \frac{\varepsilon}{n}\right)^i f_i^* - (q_i^* - q_i)_+ \cdot c(P^*[v_i, t]) + (1 - q_i)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_{i+1}, t)) \\ &= \left(1 + \frac{\varepsilon}{n}\right)^i f_i^* - (q_i^* - q_i)_+ \cdot (c_{e_i} + c(P^*[v_{i+1}, t])) + (1 - q_i)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_{i+1}, t)) \\ &= \left(1 + \frac{\varepsilon}{n}\right)^i f_i^* + (1 - \max\{q_i^*, q_i\})c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_{i+1}, t)) - (q_i^* - q_i)c(P^*[v_{i+1}, t]) \\ &\leq \left(1 + \frac{\varepsilon}{n}\right)^i (f_i^* + (1 - q_i^*)c_{e_i} + \pi_{e_i}(F + \text{SP}_c(v_{i+1}, t))) - (q_i^* - q_i)_+ \cdot c(P^*[v_{i+1}, t]) \\ &= \left(1 + \frac{\varepsilon}{n}\right)^i f_{i+1}^* - (q_{i+1}^* - q_{i+1})_+ \cdot c(P^*[v_{i+1}, t]). \end{aligned}$$

Por otro lado, si  $v_{i+1} \in P_i$ , entonces  $v_{i+1}$  aparece en el  $(s, v_i)$ -camino  $P_i$ . Luego,  $(f_i, q_i, P_i)$  proviene de la propagación de una etiqueta  $(f'_{i+1}, q'_{i+1}, P'_{i+1})$  que en alguna iteración anterior hubo en  $v_{i+1}$ , la cual puede o no seguir existiendo al final de la iteración  $i + 1$ . A continuación probaremos que

$$f'_{i+1} \leq \left(1 + \frac{\varepsilon}{n}\right)^i f_{i+1}^* - (q_{i+1}^* - q'_{i+1})_+ \cdot c(P^*[v_{i+1}, t]).$$

Denotemos por  $(e_{\alpha_1}, \dots, e_{\alpha_\ell})$ , con  $e_{\alpha_j} = (v_{\alpha_j}, v_{\alpha_{j+1}})$ ,  $v_{\alpha_1} = v_{i+1}$  y  $v_{\alpha_{\ell+1}} = v_i$  a la secuencia de arcos que define al subcamino  $P_i[v_{i+1}, v_i]$ . Entonces  $q_i = q'_{i+1} + \pi(P_i[v_{i+1}, v_i])$  y

recordando que  $c(P^*) \leq F + SP_c(s, t)$ , se tiene que

$$\begin{aligned}
f_i &= f'_{i+1} + \sum_{k=1}^{\ell} \left( 1 - q'_{i+1} - \sum_{j=1}^{k-1} \pi_{e_{\alpha_j}} \right) c_{e_{\alpha_k}} + \pi_{e_{\alpha_k}} (F + SP_c(v_{\alpha_{k+1}}, t)) \\
&= f'_{i+1} + \sum_{k=1}^{\ell} (1 - q'_{i+1} - \pi(P_i[v_{i+1}, v_i])) c_{e_{\alpha_k}} + \pi_{e_{\alpha_k}} (F + SP_c(v_{\alpha_{k+1}}, t)) + \sum_{k=1}^{\ell} \sum_{j=k}^{\ell} \pi_{e_{\alpha_j}} c_{e_{\alpha_k}} \\
&= f'_{i+1} + \sum_{k=1}^{\ell} (1 - q'_{i+1} - \pi(P_i[v_{i+1}, v_i])) c_{e_{\alpha_k}} + \pi_{e_{\alpha_k}} (F + SP_c(v_{\alpha_{k+1}}, t)) + \sum_{k=1}^{\ell} \sum_{j=1}^k \pi_{e_{\alpha_k}} c_{e_{\alpha_j}} \\
&= f'_{i+1} + \sum_{k=1}^{\ell} (1 - q'_{i+1} - \pi(P_i[v_{i+1}, v_i])) c_{e_{\alpha_k}} + \pi_{e_{\alpha_k}} \left( \sum_{j=1}^k c_{e_{\alpha_j}} + F + SP_c(v_{\alpha_{k+1}}, t) \right) \\
&\geq f'_{i+1} + \sum_{k=1}^{\ell} \pi_{e_{\alpha_k}} (F + SP_c(v_{i+1}, t)) \\
&\geq f'_{i+1} + \pi(P_i[v_{i+1}, v_i]) c(P^*[v_{i+1}, t]).
\end{aligned}$$

En otras palabras,  $f'_{i+1} \leq f_i - (q_i - q'_{i+1}) c(P^*[v_{i+1}, t])$ . Además, es fácil verificar que

$$\pi_{e_i} + \max\{q_i^*, q_i\} - q'_{i+1} = (q_{i+1}^* - q'_{i+1})_+.$$

Usando la hipótesis de inducción y las observaciones anteriores, sigue que

$$\begin{aligned}
f'_{i+1} &\leq f_i - (q_i - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&\leq \left( 1 + \frac{\varepsilon}{n} \right)^i f_i^* - (q_i^* - q_i)_+ \cdot c(P^*[v_i, t]) - (q_i - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&\leq \left( 1 + \frac{\varepsilon}{n} \right)^i f_i^* - (q_i^* - q_i)_+ \cdot c(P^*[v_{i+1}, t]) - (q_i - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&= \left( 1 + \frac{\varepsilon}{n} \right)^i f_i^* - (\max\{q_i^*, q_i\} - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&= \left( 1 + \frac{\varepsilon}{n} \right)^i (f_{i+1}^* - (1 - q_i^*) c_{e_i} - \pi_{e_i} (F + SP_c(v_{i+1}, t))) \\
&\quad - (\max\{q_i^*, q_i\} - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&\leq \left( 1 + \frac{\varepsilon}{n} \right)^i f_{i+1}^* - \pi_{e_i} (F + SP_c(v_{i+1}, t)) - (\max\{q_i^*, q_i\} - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&\leq \left( 1 + \frac{\varepsilon}{n} \right)^i f_{i+1}^* - \pi_{e_i} c(P^*[v_{i+1}, t]) - (\max\{q_i^*, q_i\} - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&= \left( 1 + \frac{\varepsilon}{n} \right)^i f_{i+1}^* - (\pi_{e_i} + \max\{q_i^*, q_i\} - q'_{i+1}) c(P^*[v_{i+1}, t]) \\
&= \left( 1 + \frac{\varepsilon}{n} \right)^i f_{i+1}^* - (q_{i+1}^* - q'_{i+1})_+ \cdot c(P^*[v_{i+1}, t]).
\end{aligned}$$

Si al final de la iteración  $(i + 1)$  la etiqueta  $(f'_{i+1}, q'_{i+1}, P'_{i+1})$  ha sido reemplazada, de manera que  $(f''_{i+1}, q''_{i+1}, P''_{i+1})$  es la última etiqueta que ocupa su lugar, entonces  $f'_{i+1}, f''_{i+1} \in I_k$  para algún  $k \in \{0, \dots, K\}$  y  $q''_{i+1} \geq q'_{i+1}$ . Si  $k = 0$ , entonces  $f'_{i+1} = 0$ . En efecto, como  $F\pi(P'_{i+1}) \leq f'_{i+1} < 1$ , se tiene que  $\pi(P'_{i+1}) = 0$ , ya que hemos supuesto que  $F\pi_e > 1$

cuando  $\pi_e > 0$ . Entonces,  $f'_{i+1} = c(P'_{i+1}) < 1$ , lo que implica que  $c(P'_{i+1}) = 0$ , es decir  $f'_{i+1} = 0$ . Por otro lado, si  $k > 0$ ,

$$\begin{aligned} f''_{i+1} &\leq \left(1 + \frac{\varepsilon}{n}\right) f'_{i+1} \\ &\leq \left(1 + \frac{\varepsilon}{n}\right)^{i+1} f_{i+1}^* - (q_{i+1}^* - q'_{i+1})_+ \cdot c(P^*[v_{i+1}, t]) \\ &\leq \left(1 + \frac{\varepsilon}{n}\right)^{i+1} f_{i+1}^* - (q_{i+1}^* - q''_{i+1})_+ \cdot c(P^*[v_{i+1}, t]), \end{aligned}$$

concluyéndose la demostración.  $\square$

*Análisis de complejidad.* La inicialización de las etiquetas toma  $O(nK)$  operaciones. Las líneas 5 a 9 implican una cantidad  $mnK$  de llamadas a  $\text{PUSH}(L, e)$ , la cual a su vez ejecuta  $O(K)$  operaciones en cada llamada. Esto nos da un complejidad total de  $O(mnK^2)$ . Para lograr que  $F\pi_e \geq 1$  para todo  $e \in E$  con  $\pi_e > 0$ , podemos escalar  $M$  y  $c$  por  $\alpha = \left\lceil \frac{1}{F\pi_{\min}} \right\rceil$ , con  $\pi_{\min} = \min \{\pi_e : e \in E, \pi_e > 0\}$ . A continuación veremos que  $K$  es polinomial en el tamaño del input y  $1/\varepsilon$ , aún después de este escalamiento. En este caso, se tiene que  $K = \left\lceil \log_{1+\frac{\varepsilon}{n}} (\alpha (F + (n-1)c_{\max})) \right\rceil$ . No es difícil probar que

- $\log(\alpha) \leq 1 + \log(p_{\max})$ .
- $\log(F + (n-1)c_{\max}) \leq 1 + \log(n) + \log(c_{\max}) + \log(F)$ .

A partir de estas desigualdades y del hecho que  $\log\left(1 + \frac{\varepsilon}{n}\right) \geq \frac{\varepsilon}{2n}$  cuando  $\varepsilon < 1$ , se tiene que

$$\begin{aligned} K &\leq \log_{1+\frac{\varepsilon}{n}} (\alpha (F + (n-1)c_{\max})) + 1 \\ &= \frac{\log(\alpha) + \log(F + (n-1)c_{\max})}{\log\left(1 + \frac{\varepsilon}{n}\right)} + 1 \\ &\leq \frac{2n}{\varepsilon} (\log(\alpha) + \log(F + (n-1)c_{\max})) + 1 \\ &\leq \frac{2n}{\varepsilon} (2 + \log(n) + \log(c_{\max}) + \log(p_{\max}) + \log(F)) + 1, \end{aligned}$$

y por lo tanto,  $K = O\left(\frac{n}{\varepsilon} (\log(n) + \log(c_{\max}) + \log(p_{\max}) + \log(F))\right)$ .

**Teorema 2.26.** *Existe un FPTAS para el problema  $\text{FMP}_A$  en grafos generales, que calcula una  $(1 + \frac{\varepsilon}{n})^{n-1}$ -aproximación en tiempo*

$$O\left(\frac{mn^3}{\varepsilon^2} (\log(n) + \log(c_{\max}) + \log(p_{\max}) + \log(F))^2\right).$$

## 2.6. Impacto de la adaptatividad

Sean  $\text{OPT}_A := \min_{P \in \mathcal{P}_{st}} f_{A,\pi}(P)$  y  $\text{OPT}_N := \min_{P \in \mathcal{P}_{st}} f_{N,\pi}(P)$ . Recordemos que la desigualdad (1.3) establece que

$$0 \leq f_{A,\pi}(P) \leq f_{N,\pi}(P) \leq F + \text{SP}_c(s, t) \quad \forall P \in \mathcal{P}_{st}.$$

A partir de ella, es directo que  $\text{OPT}_A \leq \text{OPT}_N$ . Además, si bien el modelo estudiado por Correa et al. en [6] es diferente al de este trabajo, los autores prueban que  $\text{OPT}_N \leq \frac{4}{3}\text{OPT}_A$ , y remarcan que la demostración sigue siendo válida cuando los caminos  $P \in \mathcal{P}_{st}$  tienen una probabilidad arbitraria  $\pi(P)$ . En particular, el resultado es cierto cuando las inspecciones en arcos específicos no son eventos independientes, que resulta ser el caso del problema en estudio. A continuación mostramos un ejemplo en que cota es ajustada. Consideremos la instancia de la Figura 2.3:

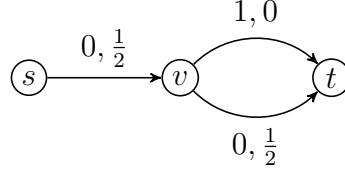


Figura 2.3: La instancia de la figura muestra que la cota  $\text{OPT}_N \leq \frac{4}{3}\text{OPT}_A$  es ajustada.

En la instancia de la Figura 2.3, sean  $e_0 = (s, v)$  y  $e_1, e_2$  los arcos  $(v, t)$  superior e inferior respectivamente. Para un valor de la multa  $F = 2$ , se tiene que  $\text{OPT}_A = \frac{3}{2}$  (donde el camino óptimo es  $P = \{e_0, e_1\}$ ) y  $\text{OPT}_N = 2$ , el cual es alcanzado por ambos caminos.

Como comentario final de este capítulo, remarcamos que la pregunta por la complejidad del problema del seguidor adaptativo sigue abierta, incluso en grafos acíclicos. En este caso, una pregunta interesante que queda sin respuesta tiene que ver con determinar si existe una cota superior subexponencial para la cantidad de puntos de quiebre de las funciones  $R_u(q)$ , o bien encontrar una instancia de ejecución exponencial del Algoritmo 1. Con respecto al problema en grafos generales, aparte de la pregunta por la complejidad, sería de interés diseñar un FPTAS basado en el redondeo de los costos de los arcos para luego utilizar el Algoritmo 3, cuyo tiempo de ejecución es pseudopolinomial.

# Capítulo 3

## Problema de maximización del líder

En este capítulo estudiaremos el problema de maximización que enfrenta el líder, definido en la sección 1.3.2. Definimos el *input* de este problema como una tupla  $(G, s, t, c, F)$ , donde  $G = (V, E)$  es un grafo dirigido con  $|V| = n$  y  $|E| = m$ ,  $s, t \in V$  son los nodos de origen y destino,  $c \in \mathbb{Z}_+^E$  es el vector de los tiempos de viaje sobre los arcos (en unidades monetarias) y  $F \in \mathbb{Z}_+$  es el valor de la multa. De igual manera que en el Capítulo 2, denotaremos por  $c_{\text{máx}}$  al máximo costo de un arco y  $F$  al mayor numerador o denominador que aparece en la representación de  $F$ . Así, el tamaño del *input* queda definido por los parámetros  $m, n, \log(c_{\text{máx}})$  y  $\log(F)$ .

### 3.1. Problema de maximización del líder ante un seguidor no adaptativo

#### 3.1.1. Maximización del costo esperado del seguidor

Comenzaremos estudiando el problema de maximización del líder para el caso en que el seguidor es no adaptativo. En primer lugar analizaremos el objetivo de maximizar el costo esperado del seguidor, es decir  $\text{LMP}_N^{\text{cost}}$ . En la sección 1.3.1, vimos que el problema de minimización del evasor en este caso puede ser resuelto en tiempo polinomial, pues se reduce a calcular un  $(s, t)$ -*shortest path* en el grafo donde el costo de cada arco  $e \in E$  es  $c_e + F\pi_e$ . Es bien sabido [2] que este problema puede formularse como un problema de programación lineal cuyo dual está dado por

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^V} \quad & \lambda_t - \lambda_s \\ \text{s.a.} \quad & \lambda_w - \lambda_v \leq c_e + F\pi_e \quad \forall e = (v, w) \in E. \end{aligned}$$

Agregando las restricciones asociadas a la distribución  $\pi$ , podemos reformular el problema  $\text{LMP}_N^{\text{cost}}$  como:

$$\begin{aligned} & \max_{\lambda \in \mathbb{R}^V, \pi \in \mathbb{R}^E} && \lambda_t - \lambda_s \\ \text{s.a.} & && \lambda_w - \lambda_v \leq c_e + F\pi_e \quad \forall e = (v, w) \in E \\ & && \sum_{e \in E} \pi_e \leq 1 \\ & && \pi_e \geq 0 \quad \forall e \in E. \end{aligned}$$

Se sabe que un problema de programación lineal con una cantidad polinomial de restricciones en el tamaño del *input* puede ser resuelto en tiempo polinomial [12], de lo cual concluimos el siguiente resultado:

**Teorema 3.1.** *El problema  $\text{LMP}_N^{\text{cost}}$  puede ser resuelto en tiempo polinomial en grafos generales.*

### 3.1.2. Maximización de la multa recolectada

A continuación abordaremos el problema de maximizar la multa recolectada ante un seguidor no adaptativo, es decir,  $\text{LMP}_N^{\text{fine}}$ . El problema de decisión asociado es el siguiente:

Dado  $r \in \mathbb{Q}$ , ¿Existen un vector  $\pi \in \mathbb{Q}^E$  y un camino  $P^* \in \mathcal{P}_{st}$  tales que

- (1)  $\sum_{e \in E} \pi_e \leq 1$
- (2)  $\pi \geq 0$
- (3)  $P^* \in \text{argmin}_{P \in \mathcal{P}_{st}} f_{N, \pi}(P)$

y  $F\pi(P^*) \geq r$ ?

Claramente, un certificado para este problema consiste en un vector  $\pi \in \mathbb{Q}^E$  y un camino  $P^* \in \mathcal{P}_{st}$ . Es fácil verificar si  $\pi$  satisface (1) y (2), mientras que (3) se puede verificar en tiempo polinomial resolviendo  $\text{FMP}_N$  usando a  $\pi$  como *input* (ver sección 1.3.1). Con esto, hemos probado que el problema pertenece a la clase NP.

En lo que sigue, probaremos que el problema de decisión asociado a  $\text{LMP}_N^{\text{fine}}$  es en realidad NP-completo. Para ello, nos inspiramos en un trabajo previo de Roch, Savard y Marcotte [16], en el cual se estudia el problema *Stackelberg shortest path*, un problema similar a  $\text{LMP}_N^{\text{fine}}$ . En él, el líder establece una asignación de precios en un subconjunto de arcos de un grafo. Posteriormente, el seguidor viaja entre dos nodos a través de un camino que minimiza la suma de los costos de traslado más el precio de los arcos recorridos. El objetivo del líder es asignar los precios a los arcos de manera de maximizar la recaudación obtenida del camino que escoge el seguidor.

Reduciremos desde 3-SAT. Una instancia de 3-SAT corresponde a una fórmula booleana en  $n$  variables, definida como la conjunción de  $m$  cláusulas, cada una compuesta por la disyunción de tres literales, donde un literal es una variable o su negación. El problema de decisión consiste en determinar si existe una asignación de verdad sobre las variables que sea satisfacible, es decir, que toda cláusula tenga al menos un literal con valor verdadero.

Dada una instancia de 3-SAT con  $m$  cláusulas y  $n$  variables  $x_1, \dots, x_n$ , construimos una instancia del problema de decisión asociado a  $\text{LMP}_N^{\text{fine}}$  como se explica a continuación.

El grafo  $G$  tendrá arcos simples así como también arcos múltiples entre algunos pares de nodos. La idea de incorporar arcos múltiples es forzar a que en la solución óptima cada uno de ellos tenga una probabilidad suficientemente pequeña. Por simplicidad, diremos que un arco es “múltiple” y de “costo”  $c$  cuando en realidad corresponda a un conjunto de arcos, cada uno de costo  $c$ , que compartan los mismos nodos de origen y destino.

Por cada cláusula  $(u_{i1} \vee u_{i2} \vee u_{i3})$  construiremos un subgrafo que da cuenta de los literales asociados a la cláusula mediante caminos de costo 0 en que uno de sus arcos es simple, más un cuarto camino formado por un arco múltiple de costo 1, como se ve en la Figura 3.1.

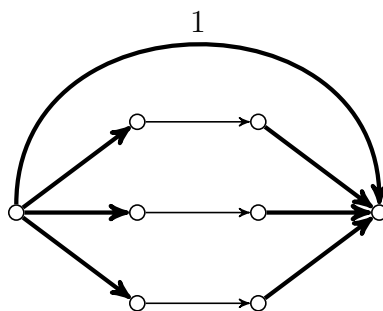


Figura 3.1: Subgrafo para una cláusula  $(u_{i1} \vee u_{i2} \vee u_{i3})$  arbitraria. Las copias del arco superior tienen costo 1. Todos los demás arcos tienen costo 0.

Estableceremos como nodo de origen  $s$  al primer nodo (en orden topológico) del subgrafo asociado a la primera cláusula. Para conectar dos cláusulas consecutivas, se utilizarán dos arcos, uno de ellos múltiple de costo 2 y el otro simple de costo 0. Adicionalmente, conectaremos un arco múltiple de costo 1 entre cada par de literales en cláusulas distintas que correspondan a una variable  $x_i$  y su negación  $\bar{x}_i$ . Por último, conectaremos la última cláusula con un arco múltiple de costo 0 a un nodo  $t$ , que será el nodo de destino (ver Figura 3.2).



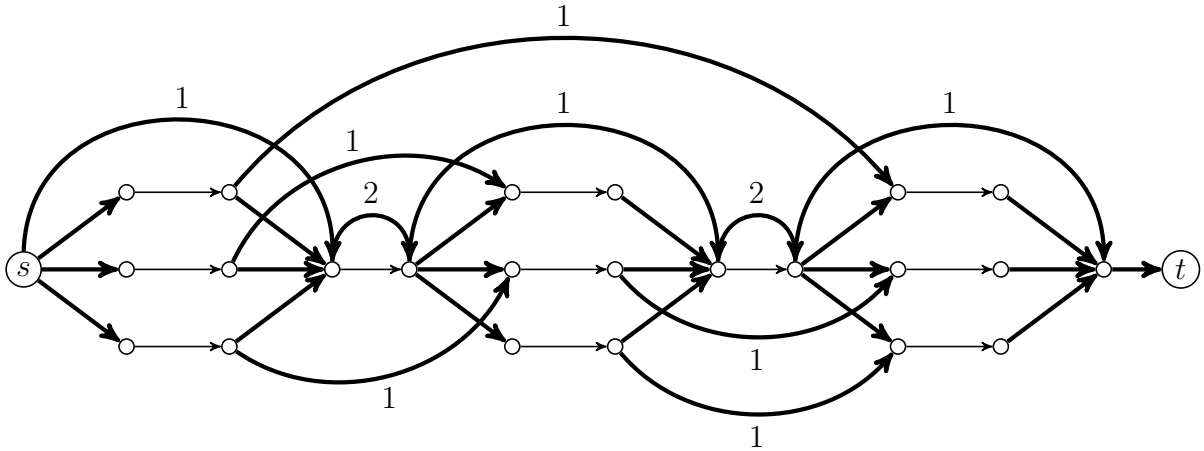


Figura 3.2: En la figura, a modo de ejemplo, se muestra el grafo correspondiente a la fórmula  $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$ . Por simplicidad, sólo se especifican los costos de los arcos de costo distinto de 0.

Finalmente, nos queda por especificar el valor de la multa  $F$ , el parámetro  $r$  y la cantidad de copias que conforman los arcos múltiples entre cada par de nodos según corresponda. Esta última será la misma para todos ellos, excepto para el arco múltiple que conecta la última cláusula con  $t$ , el cual tendrá una copia menos. Sea  $k$  el número de copias de este último arco. Sea  $C$  la suma de los costos de todos los arcos de  $G$ , contando sólo una vez el costo de cada arco múltiple. Definiremos  $F = (5 + 2C)m - 2$ ,  $k = F(3m - 1) + 1$  y  $r = 3m - 2$ . Como  $C \leq m + 2(m - 1) + (3m)^2$ , se tiene que  $C, F, k$  y  $r$  son polinomiales en el tamaño de la instancia de 3-SAT.

La siguiente propiedad se refiere a la instancia de  $\text{LMP}_N^{\text{fine}}$  que acabamos de construir.

**Propiedad 3.2.** Existen un vector  $\pi \in \mathbb{Q}^E$  y un camino  $P^* \in \mathcal{P}_{st}$  que satisfacen (1), (2) y (3) tales que  $F\pi(P^*) \geq 3m - 2$  si y sólo si existe una asignación satisfacible para 3-SAT.

*Demostración.* Supongamos que existen un vector  $\pi \in \mathbb{Q}^E$  y un camino  $P^* \in \mathcal{P}_{st}$  que satisfacen (1), (2) y (3) tales que  $F\pi(P^*) \geq 3m - 2$ .

Sin pérdida de generalidad, podemos suponer que en cada arco múltiple, todas sus copias tienen la misma probabilidad. En efecto, si  $\pi_e$  es la probabilidad mínima entre las copias de un arco múltiple  $e$ , podemos reducir la probabilidad de las copias que exceden  $\pi_e$  hasta dejarlas en este valor y el camino  $P^*$  seguirá satisfaciendo (3) y  $F\pi(P^*) \geq 3m - 2$ .

Más aún, podemos suponer, también sin pérdida de generalidad, que en cada arco múltiple de  $P^*$  distinto del arco final, todas sus copias tienen probabilidad 0. Esto, pues si  $e$  es un arco múltiple de  $P^*$  tal que sus  $(k + 1)$  copias tienen probabilidad  $\pi_e > 0$ , podemos reasignar esa cantidad  $(k + 1)\pi_e$  de probabilidad agregando  $\frac{k+1}{k}\pi_e$  a la probabilidad de cada una de las  $k$  copias del arco final, con lo cual  $P^*$  seguirá satisfaciendo (3) y  $F\pi(P^*) \geq 3m - 2$ .

Sea  $P$  el  $(s, t)$ -camino que en cada cláusula pasa por el arco múltiple de costo 1 y entre cada par de cláusulas consecutivas pasa por el arco múltiple de costo 2. Luego,  $f_{N,\pi}(P^*) \leq f_{N,\pi}(P)$ , es decir,

$$c(P^*) + F\pi(P^*) \leq c(P) + F\pi(P).$$

De las observaciones anteriores, en cada arco múltiple de  $P$ , cada copia tiene probabilidad a lo más  $\frac{1}{k}$ . Luego,

$$\begin{aligned} c(P^*) &\leq c(P) + F\pi(P) - F\pi(P^*) \\ &\leq 3m - 2 + F \left( \frac{3m - 1}{k} \right) - (3m - 2) \\ &= F \left( \frac{3m - 1}{k} \right). \end{aligned}$$

Notemos que por definición,  $F$  y  $k$  satisfacen  $F(3m - 1) < k$ . Entonces, la desigualdad anterior, junto al hecho que  $c_e \in \mathbb{Z}_+ \forall e \in E$ , implica que  $c(P^*) = 0$ . De esto se desprende que  $P^*$  debe tener la siguiente forma: pasa por cada cláusula a través de uno de los tres caminos de costo 0 y en cada conexión entre dos cláusulas consecutivas pasa por el arco de costo 0. A partir de esto, construimos la asignación satisficible para 3-SAT asignando valor verdadero al literal de cada cláusula asociada al camino por el cual pasa  $P^*$ , y estableciendo un valor de verdad arbitrario a las variables que eventualmente queden por asignar. Para concluir, basta probar que esta asignación es factible, i.e. que no existe una variable que aparezca en dos cláusulas distintas con valores de verdad opuestos. Para verificar esto, debemos probar que  $P^*$  no pasa por dos cláusulas distintas cuyos literales estén asociados a una variable y su negación.

En efecto, para obtener una contradicción, supongamos que ocurre lo anterior. Luego, existe un arco múltiple  $e$  de costo 1 que conecta las cláusulas asociadas ambos literales. Sea  $P'$  el camino que resulta de modificar  $P^*$  al tomar  $e$  en vez de los arcos que están entre ambos literales. Sea  $\ell \geq 0$  el número de cláusulas que  $P'$  se salta con respecto a  $P^*$ . Luego, el número de cláusulas en que  $P^*$  y  $P'$  coinciden es  $(m - \ell)$ , mientras que el número de arcos que conectan dos cláusulas consecutivas en que ambos caminos coinciden es  $(m - \ell - 2)$ . Además, en cada cláusula, por suboptimalidad del *shortest path*, el costo esperado de  $P^*$  es a lo más el costo esperado de tomar el arco múltiple de costo 1, mientras que en cada arco que conecta dos cláusulas consecutivas, el costo esperado de  $P^*$  es a lo más el costo esperado de tomar el arco múltiple de costo 2. Entonces,

$$\begin{aligned} f_{N,\pi}(P^*) &\leq f_{N,\pi}(P') \\ &\leq (m - \ell) \left( 1 + \frac{F}{k} \right) + (m - \ell - 2) \left( 2 + \frac{F}{k} \right) + 1 + \frac{F}{k} + \frac{F}{k} \\ &\leq m \left( 1 + \frac{F}{k} \right) + (m - 2) \left( 2 + \frac{F}{k} \right) + 1 + \frac{F}{k} + \frac{F}{k} \\ &= 3m - 3 + \frac{2mF}{k}. \end{aligned}$$

De la elección de  $F$  y  $k$  se tiene que  $2mF < k$ , luego  $f_{N,\pi}(P^*) < 3m - 2$ , pero  $f_{N,\pi}(P^*) = F\pi(P^*) \geq 3m - 2$ , lo cual es una contradicción.

Ahora supongamos que existe una asignación satisfacible para 3-SAT. Entonces cada cláusula contiene al menos un literal con valor verdadero. Construimos la distribución  $\pi$  de la siguiente manera: en cada cláusula, asignamos probabilidad  $\frac{1}{F}$  al arco simple asociado a uno de los literales que resulte verdadero y  $\frac{C+1}{F}$  a los otros dos. Además, en los arcos simples de costo 0 que conectan dos cláusulas consecutivas asignamos probabilidad  $\frac{2}{F}$ , y en todos los demás arcos de  $G$  (que resultan ser exactamente los arcos múltiples) asignamos probabilidad 0. Notemos que  $F$  fue escogido de tal forma que la suma de las probabilidades de todos los arcos en esta asignación es igual a 1. Sea  $P^*$  el  $(s, t)$ -camino definido por los literales cuya probabilidad es  $\frac{1}{F}$  y que pasa por los arcos de probabilidad  $\frac{2}{F}$  entre dos cláusulas consecutivas. Entonces  $P^* \in \operatorname{argmin}_{P \in \mathcal{P}_{st}} f_{N, \pi}(P)$ , pues  $f_N(P^*) = F\pi(P^*) = 3m - 2$  y cualquier otro  $(s, t)$ -camino tiene costo esperado al menos  $C + 1$ , excepto por el camino que resulta de tomar los arcos de costo 1 en cada cláusula y los de costo 2 entre dos cláusulas consecutivas, cuyo costo esperado es también  $3m - 2$ , pero la fracción de la multa que recolecta es 0.  $\square$

A partir de la Propiedad 3.2 hemos probado el siguiente teorema:

**Teorema 3.3.** *El problema de decisión asociado a  $\operatorname{LMP}_N^{\text{fine}}$  es NP-completo.*

## 3.2. Problema de maximización del líder ante un seguidor adaptativo

### 3.2.1. Maximización del costo esperado del seguidor

Ahora nos enfocaremos en el estudio del problema que enfrenta el líder ante un evasor adaptativo. Al igual que en la sección anterior, comenzaremos estudiando el objetivo de maximizar el costo esperado del seguidor, es decir,  $\operatorname{LMP}_A^{\text{cost}}$ . Dado  $P \in \mathcal{P}_{st}$  la función  $f_{A, \pi}(P)$  es lineal afín en  $\pi$ , así podemos reformular este problema como un problema de programación lineal:

$$\begin{aligned} & \underset{\lambda \in \mathbb{R}, \pi \in \mathbb{R}^E}{\text{máx}} && \lambda \\ \text{s.a.} &&& \lambda \leq f_{A, \pi}(P) \quad \forall P \in \mathcal{P}_{st} \\ &&& \sum_{e \in E} \pi_e \leq 1 \\ &&& \pi_e \geq 0 \quad \forall e \in E. \end{aligned} \tag{LP}_A^{\text{cost}}$$

Como vimos en la sección 2.5, el problema del evasor adaptativo  $\operatorname{FMP}_A$  admite un FPTAS. Basándonos en este resultado y en la formulación lineal de arriba, diseñaremos un FPTAS para  $\operatorname{LMP}_A^{\text{cost}}$ . Las ideas que desarrollaremos a continuación siguen de cerca los análisis publicados por Schulz et al. en [17].

Sea  $K$  un poliedro en  $\mathbb{R}^d$  y  $\varphi, \nu$  dos enteros positivos. Se dice que  $K$  tiene *complejidad de faceta* a lo más  $\varphi$  si existe un sistema de desigualdades con coeficientes racionales que

tiene como conjunto solución a  $K$  y el largo de la codificación de cada desigualdad del sistema es a lo más  $\varphi$ . Se dice que  $K$  tiene *complejidad de vértice* a lo más  $\nu$  si existen conjuntos finitos  $I$  y  $J$  de vectores racionales tales que  $K = \text{conv}(I) + \text{cono}(J)$  y el largo de la codificación de cada uno de los vectores en  $I$  y  $J$  es a lo más  $\nu$ . Ambos conceptos están relacionados mediante el siguiente lema:

**Lema 3.4** ([9, 6.2.4]). *Sea  $K \subseteq \mathbb{R}^d$  un poliedro.*

- (a) *Si  $K$  tiene complejidad de faceta a lo más  $\varphi$ , entonces  $K$  tiene complejidad de vértice a lo más  $4d^2\varphi$ .*
- (b) *Si  $K$  tiene complejidad de vértice a lo más  $\nu$ , entonces  $K$  tiene complejidad de faceta a lo más  $3d^2\nu$ .*

Un poliedro *bien descrito* es una tupla  $(K; d, \varphi)$ , donde  $K \subseteq \mathbb{R}^d$  es un poliedro cuya complejidad de faceta es a lo más  $\varphi$ .

Supongamos que  $(K; d, \varphi)$  es un poliedro racional, bien descrito y acotado en  $\mathbb{R}^d$ . En otras palabras,  $K$  es un polítopo cuya complejidad de faceta es a lo más  $\varphi$ . Sea  $\bar{K}$  una “aproximación” de  $K$  en un sentido arbitrario. Consideremos los siguientes problemas:

**Problema de separación aproximada fuerte para el polítopo  $K$  y su aproximación  $\bar{K}$  (S-APP-SEP):** dado  $y \in \mathbb{Q}^d$ , decidir si (i)  $y \in \bar{K}$ , o (ii) encontrar un hiperplano que separe a  $y$  de  $K$ : encontrar  $c \in \mathbb{Q}^d$ , con  $\|c\|_\infty = 1$  tal que  $c^T y > c^T x$  para todo  $x \in K$ .

**Problema de no vacuidad aproximada para el polítopo  $K$  y su aproximación  $\bar{K}$  (APP-NEMPT):** (i) encontrar  $y \in \bar{K}$ , o (ii) determinar que  $K$  es vacío.

El siguiente teorema, que citamos sin demostración, proviene del hecho que una versión aproximada del Método del Elipsoide puede ser utilizado con un oráculo para S-APP-SEP para resolver APP-NEMPT.

**Teorema 3.5** ([17, A.6]). *Supongamos que existe un algoritmo que puede resolver S-APP-SEP en tiempo polinomial en  $d$  y  $\varphi$ . Entonces existe un algoritmo que puede resolver APP-NEMPT en tiempo polinomial en  $d$  y  $\varphi$ .*

A continuación buscamos aplicar el Teorema 3.5 con el fin de diseñar un FPTAS para  $\text{LMP}_A^{\text{cost}}$ . Dado  $\varepsilon > 0$ , nuestro objetivo es encontrar una  $(1-\varepsilon)$ -aproximación para  $\text{LMP}_A^{\text{cost}}$  en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ . Para ello, definimos el poliedro  $Q \subseteq \mathbb{R}^{m+1}$  como la región factible del problema de programación lineal  $\text{LP}_A^{\text{cost}}$ :

$$Q := \left\{ (\pi, z) \in \mathbb{R}^m \times \mathbb{R} : \sum_{e \in E} \pi_e \leq 1, \pi_e \geq 0 \forall e \in E, z \leq f_{A,\pi}(P) \forall P \in \mathcal{P}_{st} \right\}.$$

Además, para cualquier  $\gamma \geq 0$  fijo, sea

$$Q_\gamma := \left\{ \pi \in \mathbb{R}^m : \sum_{e \in E} \pi_e \leq 1, \pi_e \geq 0 \forall e \in E, \gamma \leq f_{A,\pi}(P) \forall P \in \mathcal{P}_{st} \right\}.$$

En consecuencia, los problemas de separación aproximada fuerte y no vacuidad aproximada para los poliedros de la forma  $Q_\gamma$ , utilizando a  $Q_{(1-\varepsilon)\gamma}$  como su aproximación son:

**Problema de separación aproximada fuerte para  $Q_\gamma$  de una instancia  $(G, s, t, c, F)$  de  $\text{LMP}_A^{\text{cost}}$  y su aproximación  $Q_{(1-\varepsilon)\gamma}$  (S-APP-SEP- $Q_\gamma$ ):** dado  $\pi \in \mathbb{Q}^m$  tal que  $\sum_{e \in E} \pi_e \leq 1$  y  $\pi_e \geq 0 \forall e \in E$ , decidir si (i)  $\pi \in Q_{(1-\varepsilon)\gamma}$ , o (ii) encontrar un hiperplano que separe a  $\pi$  de  $Q_\gamma$ .

**Problema de no vacuidad aproximada para  $Q_\gamma$  de una instancia  $(G, s, t, c, F)$  de  $\text{LMP}_A^{\text{cost}}$  y su aproximación  $Q_{(1-\varepsilon)\gamma}$  (APP-NEMPT- $Q_\gamma$ ):** (i) encontrar  $\pi \in Q_{(1-\varepsilon)\gamma}$ , o (ii) determinar que  $Q_\gamma$  es vacío.

A continuación estimaremos la complejidad de faceta de  $Q$  y  $Q_\gamma$ . Recordemos que el largo de la codificación de un entero  $k \in \mathbb{Z}$  se define como  $\langle k \rangle := \lceil \log(|k| + 1) \rceil + 1$ . Bajo esta definición, las desigualdades  $\sum_{e \in E} \pi_e \leq 1$  y  $\pi_e \geq 0$  se pueden codificar en  $2m + 3$  y  $m + 3$  bits respectivamente. Por otro lado, dado  $P \in \mathcal{P}_{st}$ , la desigualdad  $z \leq f_{A,\pi}(P)$  se puede codificar en

$$\sum_{e=(v,w) \in P} \left\langle F + \text{SP}_c(w, t) - c(P[w, t]) \right\rangle + \langle 1 \rangle + \langle c(P) \rangle$$

bits. No es difícil verificar que

- $\left\langle F + \text{SP}_c(w, t) - c(P[w, t]) \right\rangle \leq 4 + \log(n) + \log(c_{\text{máx}}) + \log(F) \quad \forall e = (v, w) \in P.$
- $\langle c(P) \rangle \leq 3 + \log(n) + \log(c_{\text{máx}}).$

Con esto, se obtiene que

$$\sum_{e=(v,w) \in P} \left\langle F + \text{SP}_c(w, t) - c(P[w, t]) \right\rangle + \langle 1 \rangle + \langle c(P) \rangle \leq n(5 + \log(n) + \log(c_{\text{máx}}) + \log(F)).$$

De lo anterior, la complejidad de faceta de  $Q$  es a lo más

$$\varphi = 2m + 3 + n(5 + \log(n) + \log(c_{\text{máx}}) + \log(F)).$$

Análogamente se puede mostrar que la complejidad de faceta de  $Q_\gamma$  es a lo más

$$2m + 6 + n(5 + \log(n) + \log(c_{\text{máx}}) + \log(F)) + \langle \gamma \rangle.$$

Cuando la complejidad de faceta de  $Q_\gamma$  está acotada polinomialmente por  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$  y  $\log(F)$ , el Teorema 3.5 implica el siguiente teorema:

**Teorema 3.6.** *Fijemos  $\varepsilon > 0$ . Sea  $\gamma \geq 0$  tal que el largo de su codificación está polinomialmente acotado por  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$  y  $\log(F)$ . Supongamos que S-APP-SEP- $Q_\gamma$  puede ser resuelto en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ . Entonces, APP-NEMPT- $Q_\gamma$  puede ser resuelto en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ .*

*Observación:* si bien el Teorema 3.5 sólo nos permite concluir *a priori* que APP-NEMPT- $Q_\gamma$  puede ser resuelto en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$  y  $\log(F)$ , una revisión cuidadosa de su demostración nos permite afirmar también la polinomialidad en  $1/\varepsilon$ .

Utilizando el FPTAS para  $\text{FMP}_A$  dado por el Algoritmo 5, es posible resolver S-APP-SEP- $Q_\gamma$  en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ , como lo muestra el siguiente lema:

**Lema 3.7.** *Fijemos  $\varepsilon > 0$ . Sea  $\gamma \geq 0$  tal que el largo de su codificación está polinomialmente acotado por  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$  y  $\log(F)$ . Entonces APP-NEMPT- $Q_\gamma$  puede ser resuelto en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ .*

*Demostración.* Sean  $\varepsilon > 0$  y  $\pi \in \mathbb{R}^m$  tal que  $\sum_{e \in E} \pi_e \leq 1$  y  $\pi \geq 0$ . Consideremos el Algoritmo 5, el cual es un FPTAS para  $\text{FMP}_A$ , y ejecutémoslo en la instancia  $(G, s, t, c, F, \pi, \varepsilon)$ . Llamemos  $\bar{P}$  al camino obtenido y  $P^*$  al camino óptimo. Luego, si  $f_{A,\pi}(\bar{P}) \geq \gamma$ , entonces para todo  $P \in \mathcal{P}_{s,t}$ , se tiene que

$$\gamma \leq f_{A,\pi}(\bar{P}) \leq (1 + \varepsilon)f_{A,\pi}(P^*) \leq (1 + \varepsilon)f_{A,\pi}(P),$$

con lo cual  $(1 - \varepsilon)\gamma \leq f_{A,\pi}(P)$ , es decir,  $\pi \in Q_{(1-\varepsilon)\gamma}$ . Por otro lado, si  $f_{A,\pi}(\bar{P}) < \gamma$ , para todo  $\pi' \in Q_\gamma$  se cumple que

$$f_{A,\pi}(\bar{P}) < \gamma \leq f_{A,\pi'}(\bar{P}).$$

Luego, dado  $\varepsilon > 0$ , el Algoritmo 5 nos permite resolver S-APP-SEP- $Q_\gamma$  en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ . Del Teorema 3.6, se tiene que APP-NEMPT- $Q_\gamma$  puede ser resuelto en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ .  $\square$

A continuación proponemos el siguiente algoritmo como FPTAS para  $\text{LMP}_A^{\text{cost}}$ . Para ello, sea  $\mathcal{A}$  un algoritmo que dado  $\varepsilon > 0$ , resuelve APP-NEMPT- $Q_\gamma$  en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\text{máx}})$ ,  $\log(F)$  y  $1/\varepsilon$ . La existencia de este algoritmo está garantizada por el Lema 3.7.

---

**Algoritmo 6** FPTAS para  $\text{LMP}_A^{\text{cost}}$  en grafos generales.

---

1: Calcular  $\text{SP}_c(v, t)$  para todo  $v \in V$ .

2: Sean

$$a \leftarrow 4(m+1)^2(2m+3+n(5+\log(n)+\log(c_{\text{máx}})+\log(F))) \quad (3.1)$$

$$A \leftarrow 2^a \quad (3.2)$$

$$\Delta \leftarrow (2A)^{-2}. \quad (3.3)$$

3: Usando  $\mathcal{A}$ , encontrar  $\bar{\gamma} \in \mathbb{Q}$  vía búsqueda binaria en  $[0, F + (n-1)c_{\text{máx}}]$  tal que  $Q_{\bar{\gamma}+\Delta}$  es vacío, pero  $Q_{(1-\varepsilon)\bar{\gamma}}$  es no vacío. Sea  $\bar{\pi}$  el vector que  $\mathcal{A}$  encuentra en  $Q_{(1-\varepsilon)\bar{\gamma}}$ .

4: Encontrar  $p, q \in \mathbb{Z}$  tales que

$$1 \leq q \leq 2A \quad \text{y} \quad \left| \bar{\gamma} - \frac{p}{q} \right| < \frac{1}{2Aq}. \quad (3.4)$$

5: Usar  $\mathcal{A}$  para resolver APP-NEMPT- $Q_{p/q}$ . Si  $\mathcal{A}$  encuentra un vector en  $Q_{(1-\varepsilon)p/q}$ , denotamos a ese vector por  $\hat{\pi}$ .

6: Si  $\mathcal{A}$  encuentra un vector  $\hat{\pi}$  en  $Q_{(1-\varepsilon)p/q}$  en el paso 5, y  $\frac{p}{q} > \bar{\gamma}$ , entonces retornar  $(\pi, z) = (\hat{\pi}, (1-\varepsilon)p/q)$ .

7: En cualquier otro caso, retornar  $(\pi, z) = (\bar{\pi}, (1-\varepsilon)p/q)$ .

---

Ahora probaremos la correctitud del Algoritmo 6. Denotemos por  $z^*$  al valor óptimo de  $\text{LMP}_A^{\text{cost}}$ . En primer lugar, estableceremos que el algoritmo está bien definido mediante las siguientes afirmaciones:

1. *El intervalo de la búsqueda binaria del paso 3 es válido.* Sabemos que para todo vector  $\pi \in \mathbb{Q}^m$  tal que  $\sum_{e \in E} \pi_e \leq 1$  y  $\pi_e \geq 0 \forall e \in E$ , se cumple que  $0 \leq f_{A, \pi}(P) \leq F + \text{SP}_{s,t} \leq F + (n-1)c_{\text{máx}} \forall P \in \mathcal{P}_{st}$ , por lo que  $z^*$  se encuentra en el intervalo  $[0, F + (n-1)c_{\text{máx}}]$ .
2. *El largo de la codificación de todo valor de prueba  $\bar{\gamma}$  obtenido de la búsqueda binaria en el paso 3 está acotado polinomialmente por  $m, n, \log(c_{\text{máx}})$  y  $\log(F)$ .* Como la búsqueda binaria del paso 3 se lleva a cabo en el intervalo  $[0, F + (n-1)c_{\text{máx}}]$ , el numerador y denominador de cada valor de prueba  $\bar{\gamma}$  es no negativo. Además, la búsqueda binaria del paso 3 efectúa  $\left\lceil \log \left( \frac{F+(n-1)c_{\text{máx}}}{\Delta} \right) \right\rceil + 1$  iteraciones. Esto implica que el denominador de cada valor de prueba  $\bar{\gamma}$  es a lo más

$$2^{\left\lceil \log \left( \frac{F+(n-1)c_{\text{máx}}}{\Delta} \right) \right\rceil + 1} \leq 2^{2+\log \left( \frac{F+(n-1)c_{\text{máx}}}{\Delta} \right)} \leq \frac{4(F+(n-1)c_{\text{máx}})}{\Delta}.$$

Como la búsqueda binaria se lleva a cabo en el intervalo  $[0, F + \sum_{e \in E} c_e]$ , el numerador de cualquier valor de prueba  $\bar{\gamma}$  es a lo más  $\frac{4(F+(n-1)c_{\text{máx}})^2}{\Delta}$ . La afirmación sigue de (3.1) - (3.3).

3. El largo de la codificación de los enteros  $p$  y  $q$  calculados en el paso 4 está acotado polinomialmente por  $m$ ,  $n$ ,  $\log(c_{\max})$  y  $\log(F)$ . Por (3.4), y como  $A \geq 1$  y  $\bar{\gamma} \in [0, F + (n-1)c_{\max}]$ , se tiene que

$$\begin{aligned} p &< \bar{\gamma}q + \frac{1}{2A} \leq 2A(F + (n-1)c_{\max}) + 1 \\ p &> \bar{\gamma}q - \frac{1}{2F} \geq -\frac{1}{2}. \end{aligned}$$

Luego,  $|p| < 2A(F + (n-1)c_{\max}) + 1$ . Como  $q \leq 2A$ , la afirmación sigue de (3.1) - (3.3).

Finalmente, el siguiente lema muestra que el Algoritmo 6 retorna una  $(1-\varepsilon)$ -aproximación de  $\text{LMP}_A^{\text{cost}}$ .

**Lema 3.8.** *El Algoritmo 6 retorna un par  $(\pi, z)$  tal que  $\pi \in \mathbb{R}^m$  es una solución factible de  $\text{LP}_A^{\text{cost}}$  y  $z \geq (1-\varepsilon)z^*$ .*

*Demostración.* En primer lugar, probaremos que  $\max\{p/q, \bar{\gamma}\} \geq z^*$ , considerando dos casos:

1.  $\bar{\gamma} < z^* < \bar{\gamma} + \Delta$ . Consideremos  $p$  y  $q$  calculados en el paso 4 del algoritmo. Como para todo vector  $\pi \in \mathbb{Q}^m$  tal que  $\sum_{e \in E} \pi_e \leq 1$  y  $\pi_e \geq 0 \forall e \in E$ ,  $f_{A,\pi}$  toma valores racionales y  $z^* \in [0, F + (n-1)c_{\max}]$ , se tiene que  $z^* = \frac{r}{s}$  para algún  $r \in \mathbb{Z}_{\geq 0}$  y  $s \in \mathbb{Z}_{>0}$ . Además, la complejidad de faceta de  $Q$  es a lo más  $\varphi = 2m + 3 + n(5 + \log(n) + \log(c_{\max}) + \log(F))$ . Luego, del Lema 3.4, la complejidad de vértices de  $Q$  es a lo más  $4(m+1)^2\varphi$ , y por lo tanto  $s \in (0, 2^a) = (0, A)$ . Como

$$\frac{r}{s} - \bar{\gamma} = z^* - \bar{\gamma} < \Delta = \frac{1}{(2A)^2} \leq \frac{1}{2Aq},$$

se tiene que

$$\left| \frac{p}{q} - z^* \right| = \left| \frac{p}{q} - \frac{r}{s} \right| \leq \left| \frac{p}{q} - \bar{\gamma} \right| + \left| \bar{\gamma} - \frac{r}{s} \right| < \frac{1}{Aq} < \frac{1}{sq}.$$

Luego,  $z^* = \frac{p}{q}$ , con lo cual  $\max\{p/q, \bar{\gamma}\} \geq z^*$ .

2.  $z^* \leq \bar{\gamma}$ . Claramente,  $\max\{p/q, \bar{\gamma}\} \geq z^*$ .

A partir de lo anterior, mostramos ahora que la solución  $(\pi, z)$  que retorna el algoritmo es factible para  $\text{LP}_A^{\text{cost}}$  y que  $z \geq (1-\varepsilon)z^*$ . Consideremos los siguientes casos:

1.  $p/q > \bar{\gamma}$ . En este caso, se tiene que  $p/q \geq z^*$ . Consideremos el *output* de  $\mathcal{A}$  en el paso 5 del algoritmo:



- (a)  $\mathcal{A}$  encuentra  $\hat{\pi} \in Q_{(1-\varepsilon)p/q}$ . Entonces,  $(\pi, z) = (\hat{\pi}, (1-\varepsilon)p/q)$  es factible y  $z = (1-\varepsilon)p/q \geq z^*$ .
- (b)  $\mathcal{A}$  retorna que  $Q_{p/q}$  es vacío. Luego,  $z^* < p/q$ . Por los argumentos de arriba, se tiene que  $z^* \leq \bar{\gamma}$ . Entonces,  $(\pi, z) = (\bar{\pi}, (1-\varepsilon)\bar{\gamma})$  es factible y  $z = (1-\varepsilon)\bar{\gamma} \geq (1-\varepsilon)z^*$ .
2.  $p/q \leq \bar{\gamma}$ . En este caso, tenemos que  $z^* \leq \bar{\gamma}$ . Luego,  $(\pi, z) = (\bar{\pi}, (1-\varepsilon)\bar{\gamma})$  es factible y  $z \geq (1-\varepsilon)z^*$ .

□

*Análisis de complejidad.* Ahora analizamos el tiempo de ejecución del Algoritmo 6. El algoritmo hace un total de  $O\left(\log\left(\frac{F+(n-1)c_{\max}}{\Delta}\right)\right)$  llamadas a  $\mathcal{A}$ , el cual corre en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\max})$ ,  $\log(F)$  y  $1/\varepsilon$  en cada llamada. De (3.1) - (3.3), sigue que el tiempo total de ejecución de  $\mathcal{A}$  en el Algoritmo 6 es polinomial en  $m$ ,  $n$ ,  $\log(c_{\max})$ ,  $\log(F)$  y  $1/\varepsilon$ . Usando el método de fracciones continuas [9, pp. 134-137], encontrar enteros  $p$  y  $q$  que satisfacen (3.4) en el paso 4 del algoritmo puede hacerse en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\max})$  y  $\log(F)$ . Por lo tanto, el Algoritmo 6 corre en tiempo polinomial en  $m$ ,  $n$ ,  $\log(c_{\max})$ ,  $\log(F)$  y  $1/\varepsilon$ .

**Teorema 3.9.** *Existe un FPTAS para  $\text{LMP}_A^{\text{cost}}$ .*

### 3.2.2. Maximización de la multa recolectada

La pregunta por la complejidad del problema de maximización del valor esperado de la multa recolectada ante un seguidor adaptativo ( $\text{LMP}_A^{\text{fine}}$ ) permanece abierta. No obstante, creemos que el hecho de que el problema de decisión asociado a  $\text{LMP}_N^{\text{fine}}$  sea NP-completo debiese ser una evidencia de que el problema de decisión asociado a  $\text{LMP}_A^{\text{fine}}$  sea al menos tan difícil. Se intentó una demostración de la NP-completitud para este problema mediante la misma reducción desde 3-SAT hecha para  $\text{LMP}_N^{\text{fine}}$ , y si bien fue posible la construcción de un asignación satisfacible a partir de una distribución  $\pi \in \mathbb{Q}^E$  y un camino  $P^* \in \mathcal{P}_{st}$  que satisfacen (1), (2) y (3) tales que  $F\pi(P^*) \geq 3m - 2$  como en la Propiedad 3.2, la recíproca no prosperó.

# Bibliografía

- [1] D. Adjiashvili, G. Oriolo, and M. Senatore. The online replacement path problem. In *Algorithms–ESA 2013*, pages 1–12. Springer, 2013.
- [2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network flows. Technical report, DTIC Document, 1988.
- [3] R. Borndörfer, B. Omont, G. Sagnol, and E. Swarat. A stackelberg game to optimize the distribution of controls in transportation networks. In *Game Theory for Networks*, pages 224–235. Springer, 2012.
- [4] L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard. A bilevel model for toll optimization on a multicommodity transportation network. *Transportation Science*, 35(4):345–358, 2001.
- [5] P. J. Carstensen. The complexity of some problems in parametric linear and combinatorial programming. 1983.
- [6] J. R. Correa, T. Harks, V. J. Kreuzen, and J. Matuschke. Fare evasion in transit networks. *arXiv preprint arXiv:1405.2826*, 2014.
- [7] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.
- [8] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.
- [9] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.
- [10] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations research*, 17(1):36–42, 1992.
- [11] G. Joret. Stackelberg network pricing is hard to approximate. *Networks*, 57(2):117–120, 2011.
- [12] L. G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

- [13] La Tercera. Estudio: pérdidas por evasión representan cerca de la mitad del subsidio al Transantiago. <http://www.latercera.com/noticia/nacional/2015/02/680-616244-9-estudio-perdidas-por-evasion-representa-cerca-de-la-mitad-del-subsidio-al.shtml>. Acceso: Enero, 2016.
- [14] D. H. Lorenz and D. Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5):213–219, 2001.
- [15] E. V. Nikolova. *Strategic algorithms*. PhD thesis, Massachusetts Institute of Technology, 2009.
- [16] S. Roch, G. Savard, and P. Marcotte. An approximation algorithm for stackelberg network pricing. *Networks*, 46(1):57–67, 2005.
- [17] A. S. Schulz and N. A. Uhan. Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optimization*, 10(2):163–180, 2013.
- [18] Subsecretaría de Transportes, Gobierno de Chile. Índice de Evasión de Transantiago segundo trimestre de 2015. [http://www.fiscalizacion.cl/docs/informes\\_evasion/Evasion-May-Jul15.pdf](http://www.fiscalizacion.cl/docs/informes_evasion/Evasion-May-Jul15.pdf). Acceso: Enero, 2016.
- [19] The Local. Fare-dodging fine jumps from €40 to €60. <http://www.thelocal.de/20150630/fare-dodging-fine-increases-from-40-to-60>. Acceso: Enero, 2016.
- [20] A. Washburn and K. Wood. Two-person zero-sum games for network interdiction. *Operations Research*, 43(2):243–251, 1995.