



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

ANÁLISIS DETERMINISTA DE PROBLEMAS EN EMPAREJAMIENTO EN LÍNEA  
EN LOS QUE SE PERMITEN AUMENTACIONES

TESIS PARA OPTAR AL GRADO DE  
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS  
APLICADAS

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL MATEMÁTICO

MARIANO JOSÉ LLANCAMÁN MANSILLA

PROFESOR GUÍA:  
JOSÉ SOTO SAN MARTÍN

MIEMBROS DE LA COMISIÓN:  
MARTÍN MATAMALA VÁSQUEZ  
VÍCTOR IGNACIO VERDUGO SILVA

Este trabajo ha sido parcialmente financiado por CMM ANID BASAL FB210005

SANTIAGO DE CHILE  
2024

# Resumen

El objetivo principal de este trabajo de tesis es estudiar una familia de extensiones del problema de emparejamientos en línea: Dado un grafo bipartito  $G = (L, R, E)$  donde el lado izquierdo es conocido mientras que el lado derecho llega en línea en un orden predeterminado y desconocido, cada vértice de  $R$  solo puede emparejarse en el momento que llega y el objetivo es construir el emparejamiento de mayor tamaño posible. Este problema fue introducido y estudiado por Karp, Vazirani y Vazirani [4], quienes construyen un algoritmo aleatorio que logra en esperanza una  $1 - \frac{1}{e}$  aproximación sobre cualquier instancia. En este trabajo se estudian variaciones de este problema en las cuales, al recibir un vértice del lado en línea  $j \in R$ , el emparejamiento construido hasta ese momento puede ser aumentado con un camino aumentante comenzando en  $j$  de hasta un cierto largo. Se estudia además una familia de problemas en donde se impone un límite al número de veces que cada vértice puede ser reasignado.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Preeliminares . . . . .	1
1.2. Algoritmos en línea y tasa competitiva . . . . .	2
1.3. Problemas a estudiar . . . . .	3
1.4. Nuestros resultados y organización . . . . .	5
<b>2. Desempeño de Algoritmos Deterministas</b>	<b>6</b>
2.1. Esquema de demostración primal-dual . . . . .	7
2.2. Problemas con caminos de largo $K$ y sin límites de cambios . . . . .	8
2.3. Problemas con caminos de largo 3 . . . . .	16
<b>3. Cotas superiores</b>	<b>23</b>
<b>4. Pesos en el lado <i>offline</i></b>	<b>28</b>
<b>5. Conclusión</b>	<b>32</b>
<b>Bibliografía</b>	<b>34</b>

# Capítulo 1

## Introducción

### 1.1. Preliminares

Comenzaremos con algunas definiciones básicas de grafos y resultados conocidos.

**Definición 1.1** (Grafo) *Un grafo es un par ordenado  $(V, E)$  donde  $V$  es un conjunto finito y  $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$ .*

*Dado un grafo  $G$ ,  $V = V(G)$  es el conjunto de vértices de  $G$  y  $E = E(G)$  el conjunto de aristas. Una arista  $\{u, v\}$  será denotada  $uv$ , y llamaremos a  $u$  y  $v$  los extremos de  $uv$ .*

*Denotaremos al número de vértices  $|V(G)|$  como el orden de  $G$ .*

*Denotamos el conjunto de vecinos de  $v \in V$  por  $N(v) = N_G(v) := \{u \in V : uv \in E\}$ .*

**Definición 1.2** (Grafo inducido) *Sea  $G = (V, E)$  un grafo y  $U \subseteq V$  un subconjunto de sus vértices. Definimos el grafo inducido por  $U$  como  $G[U] = (U, \{uv \in E : u, v \in U\})$ .*

**Definición 1.3** (Grafo bipartito) *Un grafo  $(V, E)$  se dice bipartito si es posible dividir  $V$  en dos conjuntos,  $L$  y  $R$ , tal que toda arista en  $E$  tiene un extremo en  $L$  y uno en  $R$ . En este caso denotaremos a  $G$  como  $G(L, R, E)$ .*

**Definición 1.4** (Emparejamiento) *Un emparejamiento o matching en un grafo  $G = (V, E)$  es un conjunto de aristas  $\mathbf{m} \subseteq E$  disjuntas de a pares, es decir ningún par de aristas en  $\mathbf{m}$  comparten un extremo. Si  $uv \in \mathbf{m}$ , se dirá que  $\mathbf{m}$  empareja a  $u$  y  $v$ . Decimos también que  $\mathbf{m}$  cubre un vértice  $v \in V$  si existe algún  $u \in N(v)$  tal que  $uv \in \mathbf{m}$ .*

**Definición 1.5** (Emparejamiento de tamaño máximo) *Un emparejamiento  $m$  en un grafo  $G$  es de tamaño máximo o simplemente máximo si para cualquier emparejamiento  $m'$  en  $G$ ,  $|m| \geq |m'|$ .*

**Definición 1.6** (Número de matching) *El número de matching de un grafo  $G$ ,  $\nu(G)$ , es la*

cardinalidad de un matching máximo en  $G$ .

$$\nu(G) := \text{máx}\{|m| : m \text{ matching en } G\} \quad (1.1)$$

**Definición 1.7** (Camino) *Un camino  $P = v_0v_1\dots v_l$  en  $G$  es una secuencia de vértices únicos tales que para todo  $i \in \{0, 1, \dots, l-1\}$ , se tiene  $v_iv_{i+1} \in E$ . Se dirá que  $P$  es de largo  $l$  o que  $P$  es un  $l$ -camino, y denotaremos el largo de  $P$  por  $|P|$ . Diremos además que  $v_0$  y  $v_l$  son los extremos de  $P$ , y que los demás vértices son vértices interiores de  $P$ .*

**Definición 1.8** (Camino  $m$ -alternante)

Sea  $m$  un emparejamiento en un grafo  $G$ . Sea  $P = v_0v_1\dots v_l$  un camino en  $G$ .  $P$  es un camino  $m$ -alternante de largo  $l$  si  $v_0$  no está cubierto por  $m$  y las aristas de  $P$  alternan entre estar fuera y dentro de  $m$ .

**Definición 1.9** (Camino  $m$ -aumentante) *Sea  $m$  un emparejamiento en un grafo  $G$ . Sea  $P = v_0v_1\dots v_l$  un camino en  $G$ .  $P$  es un camino  $m$ -aumentante si es  $m$ -alternante y  $v_l$  no está cubierto por  $m$ . Es decir ninguno de los extremos de  $P$  están cubiertos y sus aristas alternan entre estar fuera y dentro de  $m$ .*

*Se dirá que  $m$  puede aumentarse a lo largo de  $P$ , lo que se refiere a retirar de  $m$  todas las aristas que forman parte de  $P$  y agregar a  $m$  todas las aristas de  $P$  que no estén en  $m$ . Este proceso resulta en un nuevo emparejamiento  $m'$  tal que  $|m'| = |m| + 1$ , independiente del largo del camino usado.*

*Identificando  $P$  con el conjunto de aristas que lo forman, con  $P \subseteq E$ , podemos decir que*

$$m' := m\Delta P = (m \setminus P) \cup (P \setminus m) \quad (1.2)$$

**Observación:** Un camino aumentante debe ser de largo impar.

**Proposición 1.10** (Lema de Berge [1] [5]) *Un emparejamiento  $m$  en  $G$  es máximo si y solo si no existen caminos  $m$ -aumentantes.*

**Definición 1.11** (Emparejamiento maximal) *Un emparejamiento  $m$  en un grafo  $G$  se dice maximal si toda arista en  $E(G)$  tiene al menos un extremo cubierto por  $m$ . En otras palabras,  $m$  es maximal si no es posible construir un emparejamiento de mayor tamaño agregando una arista a  $m$ .*

**Proposición 1.12** ([6]) *Si  $m$  es un emparejamiento maximal en  $G$ , entonces  $|m| \geq \frac{1}{2}\nu(G)$ .*

## 1.2. Algoritmos en línea y tasa competitiva

Un problema de optimización en línea es uno donde la instancia del problema se revela paso a paso, y se deben tomar decisiones con información incompleta manteniendo una invariante (por ejemplo, mantener un emparejamiento).

**Definición 1.13** (*F*-competitividad) *Sea un algoritmo  $\mathcal{A}$  para algún problema de maximización, y sea  $\mathcal{A}(I)$  el valor de la función objetivo obtenido por  $\mathcal{A}$  en una instancia  $I$  del problema. Sea además  $\text{OPT}(I)$  el máximo valor posible de alcanzar en la instancia  $I$ .*

*Para  $F \in [0, 1]$  se dirá que el algoritmo  $\mathcal{A}$  es  $F$ -competitivo o que alcanza una tasa competitiva de  $F$  si en cualquier instancia  $I$  el algoritmo  $\mathcal{A}$  logra una  $F$ -aproximación, es decir*

$$\mathcal{A}(I) \geq F \cdot \text{OPT}(I) \tag{1.3}$$

*Si  $\mathcal{A}$  es aleatorio, entonces  $\mathcal{A}$  es  $F$ -competitivo si*

$$\mathbb{E}[\mathcal{A}(I)] \geq F \cdot \text{OPT}(I) \tag{1.4}$$

*donde la esperanza es sobre la aleatoriedad de  $\mathcal{A}$ .*

A continuación se define el problema de emparejamiento en línea en grafos bipartitos planteado y estudiado por Karp, Vazirani y Vazirani [4] en 1990.

**Definición 1.14** (Problema de emparejamiento en línea) *Sea  $G(L, R, E)$  un grafo bipartito. Sea  $\pi$  un ordenamiento (permutación) de  $R$ . Los vértices de  $L$  son conocidos desde un principio mientras que los vértices de  $R$  llegan o son revelados en el orden  $\pi$ . Al llegar  $j \in R$  también se revelan todas las aristas incidentes a él, y se puede elegir un vecino para emparejarlo o dejarlo libre. Esta decisión es irrevocable, y un vértice  $j \in R$  no puede asignarse a un vértice que ya fue emparejado anteriormente. El objetivo es maximizar el tamaño del emparejamiento resultante tras llegar todos los vértices de  $R$ .*

*Abreviaremos este problema como OMP (online matching problem).*

En el mismo trabajo, los autores construyen un algoritmo aleatorio para OMP llamado RANKING, y demuestran que dicho algoritmo alcanza una tasa competitiva de  $1 - \frac{1}{e} = 0,632\dots$ . Además, muestran que tal algoritmo es óptimo.

**Definición 1.15** (Algoritmo RANKING) *El algoritmo RANKING para el problema de emparejamiento en línea es el siguiente.*

- *Inicialización: Elegir un ranking (permutación) de  $L$  de manera uniforme, es decir una función  $\sigma : L \rightarrow \{1, 2, \dots, |L|\}$ .*
- *Al llegar  $j \in R$ , emparejarlo con el  $i \in N(j)$  libre que minimice  $\sigma(i)$  (de existir).*

### 1.3. Problemas a estudiar

A continuación se plantean las familias de problemas a estudiar en este trabajo.

**Definición 1.16** (Problema de emparejamiento en línea con caminos aumentantes de largo  $K$  (OMP- $K$ )) *Al igual que en OMP, se tiene un grafo bipartito  $(L, R, E)$  donde  $L$  es conocido y*

$R$  llega de manera en línea en un orden  $\pi$  predeterminado y desconocido. Al llegar un vértice  $j \in R$ , el emparejamiento construido hasta el momento puede aumentarse a lo largo de un camino aumentante de largo menor o igual a  $K$  empezando en  $j$ .

Por ejemplo, si  $K = 3$  y se recibe un vértice  $l \in R$ . Si es que  $l$  tiene un vecino  $k$  que ya está emparejado a otro vértice  $j$  y además  $j$  tiene un vecino libre  $i \in L$ , entonces el emparejamiento construido puede ser aumentado con el camino  $ijkl$ , agregando al matching las aristas  $ij$  y  $kl$  y quitando la arista  $kj$  como en la figura 1.1.

Abreviaremos este problema como  $OMP-K$ .

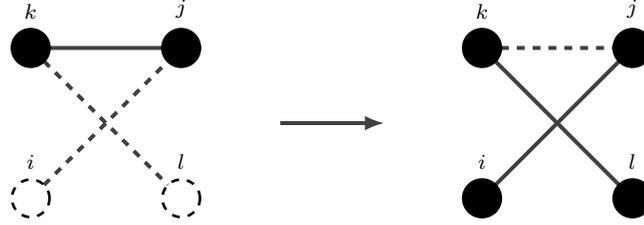


Figura 1.1: Al recibir  $l$ , se puede aumentar el matching con el camino  $ijkl$ .

**Observación:** A no ser que se indique lo contrario, en toda figura los vértices de  $L$  (lado *offline*) se dibujarán alineados verticalmente en el lado izquierdo y los de  $R$  (lado *online*) en el lado derecho.

Notamos que al usar un camino aumentante todos los vértices del camino salvo sus extremos deben cambiar su pareja, a lo que llamaremos ser re-emparejado o reasignado. En un problema de emparejamiento real es posible que se quiera minimizar o limitar el número de reasignaciones que cada vértice experimenta. Por ejemplo, si buscamos asignar pacientes a bloques de horario en un hospital en tiempo real, una reasignación implica cambiar la hora a la que un paciente está citado. Esto nos motiva a definir la siguiente familia de problemas.

**Definición 1.17** ( $OMP-K-(s, t)$ ) Sean  $s, t \in \mathbb{N} \cup \{\infty\}$ ,  $OMP-K-(s, t)$  es una variante de  $OMP-K$  en donde cada vértice de  $L$  (resp.  $R$ ) puede ser reasignado un máximo de  $s$  (resp.  $t$ ) veces a lo largo de la ejecución.

Si en una ejecución un vértice es reasignado el número máximo de veces permitido, decimos que queda agotado. Si una arista está en el matching y alguno de sus 2 extremos está agotado, decimos que está congelada. Esto último pues para retirar una arista del emparejamiento se deben re-emparejar ambos extremos.

Diremos que  $OMP-K$  es equivalente a  $OMP-K-(\infty, \infty)$ .

Una última observación importante es que en todos los problemas anteriores, una vez que un vértice (ya sea del lado *online* u *offline*) es emparejado, dicho vértice permanecerá emparejado por el resto de la ejecución. Esto es pues en el proceso de aumentar un matching con un camino aumentante todos los vértices interiores del camino permanecen emparejados.

Estudiaremos principalmente los problemas  $OMP-K$  (caminos aumentantes de largo  $K$  y sin límite de cambios),  $OMP-3-(1, 1)$  (caminos de largo 3 y cada vértice del grafo puede

cambiar una única vez) y OMP-3-(1,  $t$ ) (los vértices de  $L$  tienen un único cambio permitido y los de  $R$  tienen una cantidad fija  $t$ ). También estudiaremos una variante de OMP-3-(1, 1) en donde los vértices *offline* tienen peso y se busca maximizar el peso neto del emparejamiento construido.

## 1.4. Nuestros resultados y organización

En el capítulo 2 se establecen dos clases de algoritmos deterministas para problemas de emparejamiento, *greedy* y SAP, y se estudia el desempeño de estas clases de algoritmos en algunos de los problemas planteados anteriormente. Los resultados principales de esta sección son que, usando algoritmos deterministas simples, es posible alcanzar las siguientes tasas competitivas.

1. Si  $K = 2n - 1$  con  $n \in \mathbb{N}$ , en OMP- $K$  se puede lograr una tasa de  $\frac{n}{n+1}$ .
2. Si  $t \in \mathbb{N}$ , en OMP-3-(1,  $t$ ) se puede lograr una tasa de  $\frac{2t+1}{3t+2}$ .

En el capítulo 3 se estudian cotas superiores para el desempeño de algunos de estos problemas y se muestra que ningún algoritmo determinista puede superar las tasas competitivas dadas anteriormente para los problemas OMP- $K$  con  $K \in \mathbb{N}$  y OMP-3-(1, 1) (caso  $t = 1$ ).

Finalmente, en el capítulo 4 se plantea y estudia brevemente una variación del problema OMP-3-(1, 1) en donde los vértices del lado *offline*  $L$  tienen pesos no negativos, y se construye un algoritmo determinista que alcanza una tasa competitiva de  $\frac{1+2\sqrt{2}}{7} \approx 0,546918$ .

# Capítulo 2

## Desempeño de Algoritmos Deterministas

Comenzaremos esta sección con un breve análisis del desempeño de algoritmos deterministas para el problema de emparejamiento en línea original estudiado por Karp, Vazirani y Vazirani [4]. Este análisis no es muy interesante, pero ayudará a contextualizar los resultados principales del trabajo.

**Definición 2.1** (Algoritmo *greedy*) *Un algoritmo  $\mathcal{A}$  para un problema de emparejamiento en línea se dice greedy si siempre empareja un vértice entrante en cuanto sea posible.*

**Proposición 2.2** *Cualquier algoritmo greedy para OMP es  $\frac{1}{2}$ -competitivo.*

DEMOSTRACIÓN. Sea  $\mathcal{A}$  un algoritmo greedy para OMP y sea  $m$  el matching resultante de ejecutar  $\mathcal{A}$  sobre un grafo  $G$ . Es fácil ver que  $m$  es maximal. Por contradicción, si existiesen dos vértices  $i \in L$  y  $j \in R$  que sean vecinos y ambos sin emparejar, entonces  $i$  estuvo disponible al momento de recibir  $j$  y como  $\mathcal{A}$  es greedy  $j$  se debe haber emparejado (no necesariamente a  $i$ ). Como  $m$  es maximal, es una  $\frac{1}{2}$ -aproximación y  $\mathcal{A}$  es  $\frac{1}{2}$ -competitivo.  $\square$

**Proposición 2.3** *Ningún algoritmo determinista para OMP puede superar una tasa competitiva de  $\frac{1}{2}$ .*

DEMOSTRACIÓN. Consideremos los siguientes 2 grafos de orden 4 con orden de llegada  $r_1 r_2$ .

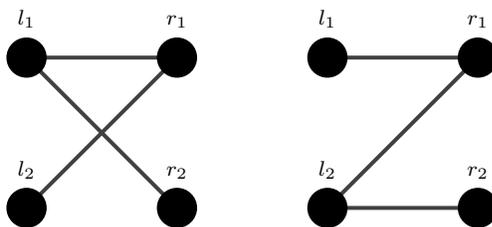


Figura 2.1: Dos grafos bipartitos de orden 4.

Sea  $\mathcal{A}$  un algoritmo determinista para OMP. Si  $\mathcal{A}$  no empareja  $r_1$  no puede lograr más que una  $\frac{1}{2}$ -aproximación en ninguno de los dos grafos. Si  $\mathcal{A}$  empareja  $r_1$  a  $l_1$ , no podrá emparejar  $r_2$  en el primer grafo, y lo mismo ocurre en el segundo grafo si empareja  $r_1$  a  $l_2$ . Por lo tanto no importa que decisión tome  $\mathcal{A}$ , solo logrará una  $\frac{1}{2}$ -aproximación en al menos uno de los dos grafos, y por lo tanto no puede superar una tasa competitiva de  $\frac{1}{2}$ .

□

Así vemos que para el problema de emparejamiento en línea original es fácil lograr de forma determinista una tasa competitiva de  $\frac{1}{2}$ , pero es imposible de superar. Buscaremos resultados de este estilo para los problemas de la familia OMP- $K$ -( $s, t$ ).

## 2.1. Esquema de demostración primal-dual

A continuación describiremos un esquema de demostración que será usado a lo largo de esta sección, utilizando propiedades de dualidad del problema de programación lineal asociado al problema de emparejamiento máximo. Este esquema fue introducido por Devanur, Jain y Kleinberg [2] quienes lo usaron para dar una nueva demostración de la tasa competitiva de RANKING.

**Definición 2.4** (Relajación PL del problema de emparejamiento máximo) *Sea  $G = (V, E)$  un grafo cualquiera. La relajación del problema de emparejamiento máximo en  $G$  es el siguiente problema de programación lineal  $P_G$ .*

$$\begin{aligned}
 (P_G) \quad & \text{máx}_{x_e, e \in E} \quad \sum_{e \in E} x_e \\
 \text{s.a.} \quad & \forall v \in V \quad \sum_{u \in N(v)} x_{uv} \leq 1 \\
 & \forall e \in E \quad x_e \geq 0
 \end{aligned} \tag{2.1}$$

Su dual  $D_G$  es el siguiente.

$$\begin{aligned}
 (D_G) \quad & \text{mín}_{y_v, v \in V} \quad \sum_{v \in V} y_v \\
 \text{s.a.} \quad & \forall uv \in E \quad y_u + y_v \geq 1 \\
 & \forall v \in V \quad y_v \geq 0
 \end{aligned} \tag{2.2}$$

Si  $G$  es bipartito con  $V = L \cup R$ , separaremos las variables duales en  $(\alpha_i)_{i \in L}$  y  $(\beta_j)_{j \in R}$  y por lo tanto el dual  $D_G$  será

$$\begin{aligned}
(D_G) \quad & \min_{\alpha_i, \beta_j, i \in L, j \in R} \sum_{i \in L} \alpha_i + \sum_{j \in R} \beta_j \\
\text{s.a.} \quad & \forall ij \in E \quad \alpha_i + \beta_j \geq 1 \\
& \forall i \in L, j \in R \quad \alpha_i, \beta_j \geq 0
\end{aligned} \tag{2.3}$$

Cuando busquemos demostrar que un algoritmo  $\mathcal{A}$  para algún problema de emparejamiento en línea alcanza una tasa competitiva de  $F$  procederemos como sigue. Consideremos una ejecución de  $\mathcal{A}$  en una instancia arbitraria  $(G, \pi)$ . El esquema corresponde a mantener en paralelo a la ejecución un vector dual no negativo e inicialmente igual a 0,  $y = (y_v)_{v \in V} = (\alpha_i, \beta_j)_{i \in L, j \in R}$  de tal forma que:

1. Cada vez que se agregue una arista al emparejamiento, el valor dual de  $y$  (la suma de todos sus componentes) aumenta a lo más en una constante  $\frac{1}{F}$ .
2. Al finalizar la ejecución el vector  $y$  es  $D_G$ -factible, es decir para toda arista  $ij \in E$ ,  $\alpha_i + \beta_j \geq 1$ .

Sea  $X$  el tamaño del matching resultante,  $Y$  el valor dual final del vector construido y  $OPT = \max P_G = \min D_G$ . El primer punto garantiza que  $Y \leq \frac{1}{F}X$ , y como  $y$  es  $D_G$ -factible  $OPT \leq Y$ . Además  $\nu(G) \leq OPT$  pues  $P_G$  es una relajación. Luego

$$\nu(G) \leq OPT \leq Y \leq \frac{1}{F}X \implies \frac{X}{\nu(G)} \geq F \tag{2.4}$$

y como esto es válido para cualquier instancia,  $\mathcal{A}$  es  $F$ -competitivo.

## 2.2. Problemas con caminos de largo $K$ y sin límites de cambios

En esta sección extenderemos la proposición 2.2 a los problemas OMP- $K$  con  $K \in \mathbb{N}$ . Recordemos que este problema es una extensión de OMP donde, al recibir un vértice  $j \in R$ , el matching construido puede aumentarse con un camino aumentante de largo menor o igual a  $K$  comenzando en  $j$ .

Primero veremos el caso con  $K = 3$ .

**Proposición 2.5** *Sea  $\mathcal{A}$  un algoritmo greedy para OMP-3, y sea  $m$  el matching resultante de ejecutar  $\mathcal{A}$  en un grafo  $G$ . Entonces  $m$  es un matching maximal y sin caminos aumentantes de largo 3.*

DEMOSTRACIÓN. Sean  $\mathcal{A}$ ,  $G$  y  $m$  como en el enunciado. Podemos concluir que  $m$  es maximal con el mismo argumento que en la proposición 2.2. Supongamos buscando una contradicción que al terminar la ejecución de  $\mathcal{A}$  existe un 3-camino aumentante  $ijkl$ , es decir existen

- Un vértice  $i \in L$  sin emparejar.

- Una arista  $kj \in m$  con  $i \in N(j)$ .
- Un vértice  $l \in N(k)$  sin emparejar.

Consideremos el instante en que  $\mathcal{A}$  recibe  $l \in R$ . Como  $l \in R$  no fue emparejado, sabemos que en ese instante

1.  $k \in N(l)$  estaba emparejado pues de lo contrario  $\mathcal{A}$  podría haber emparejado  $l$ .
2. la arista  $kj$  no estaba en el matching construido hasta el momento pues de lo contrario  $\mathcal{A}$  podría haber emparejado  $l$  usando el camino  $ijkl$ .

Digamos que en este instante  $k$  estaba emparejado a  $x \in R \setminus \{j\}$ . Como  $l$  no se pudo emparejar, en este instante todos los vecinos de  $x$  estaban emparejados. Luego  $kx$  no puede salir del matching, lo que contradice que en un tiempo posterior  $k$  debe estar emparejado a  $j$ .

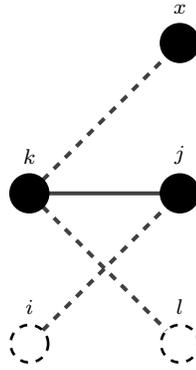


Figura 2.2: Estado al terminar la ejecución de  $\mathcal{A}$ .

□

**Proposición 2.6** *Sea  $G$  un grafo cualquiera. Si  $m$  es un matching de  $G$  maximal y sin caminos aumentantes de largo 3, entonces  $|m| \geq \frac{2}{3}\nu(G)$ .*

DEMOSTRACIÓN. Sea  $m$  un emparejamiento en  $G$  sin caminos aumentantes de largo 1 ó 3. Sea  $\text{OPT} = \max(P_G) = \min(D_G)$ . Por el esquema primal-dual descrito anteriormente, nos basta construir un vector  $(y_v)_{v \in V}$   $D_G$ -factible tal que  $\sum_{v \in V} y_v \leq \frac{3}{2}|m|$ .

Primero veremos el caso en que  $G$  no tiene ciclos de largo 3 (por ejemplo, que  $G$  es bipartito). Sea  $y = (y_v)_{v \in V}$  definido por

$$y_v = \begin{cases} 0 & \text{si } v \text{ no es emparejado por } m \\ 1 & \text{si } v \text{ es emparejado por } m \text{ y tiene algún vecino que no lo es} \\ \frac{1}{2} & \text{si } v \text{ y todos sus vecinos son emparejados por } m \end{cases}$$

Primero,  $y$  es dual-factible pues la existencia de una arista  $uv$  tal que  $y_u + y_v < 1$  implica que al menos uno de los 2 valores, digamos  $y_v$ , es 0 por lo que  $v$  no está emparejado. Como

$m$  es maximal  $u$  debe estar emparejado y como tiene un vecino sin emparejar ( $v$ )  $y_u$  es 1, lo que contradice  $y_u + y_v < 1$ .

Segundo,  $\sum y \leq \frac{3}{2}|m|$ . Como solo los vértices emparejados por  $m$  aportan a  $\sum y$  y cada arista  $uv \in m$  aporta exactamente  $y_u + y_v$ , nos basta ver que para toda arista  $uv \in m$ ,  $y_u + y_v \leq \frac{3}{2}$ . Supongamos que existe una arista en  $m$  tal que  $y_u + y_v > \frac{3}{2}$ . La única forma en que esto ocurra es que  $y_u = y_v = 1$ , por lo que  $u$  y  $v$  deben tener vecinos sin emparejar. Como no hay 3-ciclos en  $G$  estos vecinos no son el mismo vértice y entonces junto a  $u$  y  $v$  formarían un camino aumentante de largo 3. Luego cada arista de  $m$  aporta a lo más  $\frac{3}{2}$  a  $\sum y$ , lo que termina la demostración para grafos  $G$  sin ciclos de largo 3.

Si ahora  $G$  es un grafo general, la asignación anterior no funciona en la siguiente situación: existe una arista  $uv \in m$  y un vértice  $w$  sin emparejar que es vecino de  $u$  y de  $v$  como en la Figura 2.3. En la construcción anterior la arista  $uv$  aportaría 2 a  $\sum y$ , que supera el límite por arista de  $\frac{3}{2}$ .

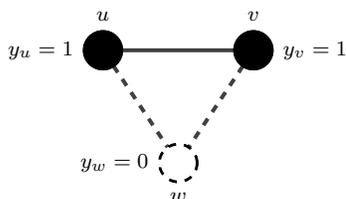


Figura 2.3: Grafo en el que la asignación inicial no funciona.

Notamos que si  $uv \in m$  y existe un vértice  $w$  que es vecino de ambos extremos y no está emparejado, entonces ni  $u$  ni  $v$  pueden tener otro vecino sin emparejar distinto de  $w$ . Esto pues si, por ejemplo, existe un vértice  $\hat{w} \in N(u)$  que no está emparejado,  $\hat{w}uvw$  sería un 3-camino aumentante.

Para terminar la demostración para el caso general modificaremos la asignación dual. Empezamos con el mismo vector  $(y_v)_{v \in V}$  anterior y revisamos cada arista  $uv \in m$ . Si  $u$  y  $v$  comparten un vecino que no está emparejado  $w_{uv}$  (recordemos que de existir  $w_{uv}$  es único), fijamos  $y_u = y_v = y_{w_{uv}} = \frac{1}{2}$  como en la figura 2.4. Notar que para 2 aristas  $e_1$  y  $e_2$  es posible que  $w_{e_1} = w_{e_2}$ .

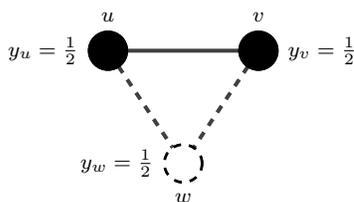


Figura 2.4: Asignación dual modificada.

Si consideramos  $y_{w_e}$  dentro del valor aportado por la arista  $e \in m$  nuevamente concluimos que ninguna arista del matching contribuye más de  $\frac{3}{2}$  a  $\sum_{v \in V} y_v$ . Notar que si múltiples aristas comparten un mismo  $w_e$  podemos decir que una de ellas aporta  $\frac{3}{2}$  y las demás aportan solo  $1 < \frac{3}{2}$ . Por lo tanto  $\sum_{v \in V} y_v \leq \frac{3}{2}|m|$ .

Por otro lado, sea  $uv \in E$  y supongamos que  $y_u + y_v < 1$ . Nuevamente podemos suponer que  $y_v = 0$  y, como  $m$  es maximal,  $y_u$  está emparejado y  $y_u = \frac{1}{2}$ . Como  $u$  tiene un vecino sin emparejar la única forma en que  $y_u = \frac{1}{2}$  es que  $u$  sea parte de un ciclo de largo 3 con  $u\hat{u} \in m$  y  $w = w_{u\hat{u}}$  sin emparejar. Pero ya vimos que  $u$  no puede tener un vecino sin emparejar distinto de  $w$ , y  $w \neq v$  pues  $y_w = \frac{1}{2}$ . Luego por contradicción,  $y_u + y_v \geq 1$  y el vector dual es  $D_G$ -factible.

□

Las proposiciones 2.5 y 2.6 nos permiten concluir lo siguiente.

**Corolario 2.7** *Cualquier algoritmo determinista greedy para OMP-3 alcanza una tasa competitiva de  $\frac{2}{3}$ .*

A continuación buscaremos extender este resultado para un valor de  $K$  general. Sin embargo veremos que un algoritmo *greedy* no es suficiente. En efecto, consideremos el caso  $K = 7$  y el siguiente grafo con  $L = \{l_1, l_2, \dots, l_7\}$  y  $R = \{r_1, r_2, \dots, r_7\}$  con orden de llegada  $r_1 r_2 \dots r_7$ .

En el lado izquierdo de la figura 2.5 se muestra el estado de un algoritmo greedy en el instante en que recibe el último vértice  $r_7$ . Han llegado los vértices  $r_1, r_2, r_3, r_4$  y  $r_5$  y cada uno eligió su pareja actual directamente. El sexto vértice en llegar,  $r_6$ , no pudo ser emparejado pues su único vecino ya está ocupado y el único camino aumentante existente, que termina en  $l_5$ , es de largo 9. Sin embargo el vértice  $r_7$  sí puede emparejarse, y el algoritmo utiliza el camino aumentante  $r_7 l_6 r_5 l_3 r_3 l_4 r_4 l_5$ , de largo 7.

En el lado derecho de la figura 2.5 se muestra el estado del emparejamiento tras usar este camino, y se observa que al invertir sus aristas se genera en el grafo un nuevo camino aumentante de largo 7:  $r_6 l_1 r_1 l_2 r_2 l_3 r_3 l_5$ .

Este ejemplo muestra que, dado un valor de  $K$  arbitrario, un algoritmo greedy para OMP- $K$  **no** necesariamente resultará siempre en un emparejamiento sin caminos aumentantes de largo menor o igual a  $K$ .

Como un algoritmo *greedy* cualquiera no es suficiente para garantizar un matching sin caminos aumentantes de largo menor o igual a  $K$ , introduciremos la siguiente clase de algoritmos.

**Definición 2.8** (Algoritmo SAP) *Un algoritmo para un problema de emparejamiento en línea que permite aumentaciones se dice SAP (Shortest augmenting path) si siempre utiliza uno de los caminos aumentantes más cortos disponibles.*

En particular para el caso  $K = 3$  donde solo se permite usar caminos aumentantes de largo 1 o 3, un algoritmo SAP solo usará un 3-camino aumentante si es que todos los vecinos del vértice recibido ya han sido emparejados.

Es importante notar que en la figura 2.5 era posible emparejar a  $r_7$  usando un camino de largo 3,  $r_7 l_6 r_5 l_7$ , por lo que el algoritmo usado en el contraejemplo **no** es SAP.

A continuación veremos que un algoritmo greedy y SAP es suficiente para el resultado

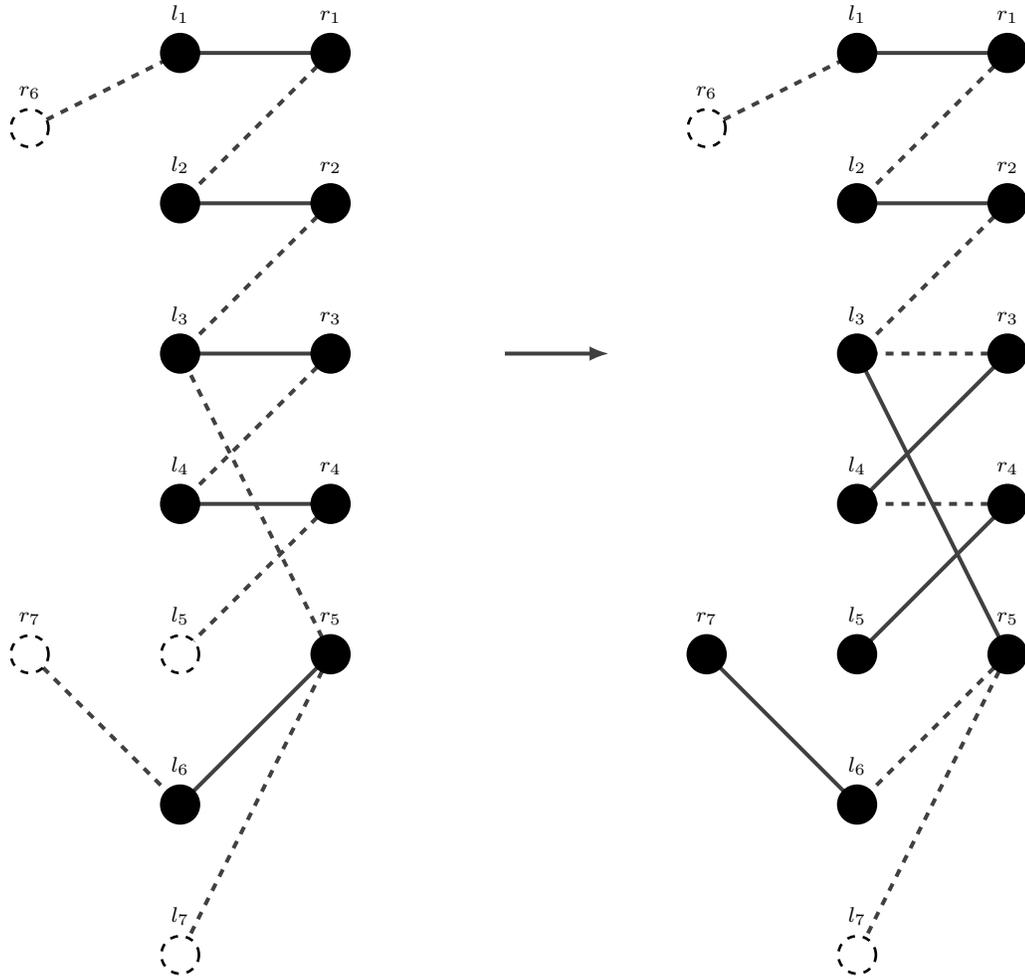


Figura 2.5: Estados antes y después de usar el camino  $r_7l_6r_5l_3r_3l_4r_4l_5$ .

que busquemos.

**Proposición 2.9** *Sea  $\mathcal{A}$  un algoritmo greedy y SAP para OMP- $K$ , y sea  $m$  el matching resultante de ejecutar  $\mathcal{A}$  en un grafo  $G$ . Entonces  $m$  es un matching sin caminos aumentantes de largo menor o igual a  $K$ .*

Para esta demostración utilizaremos el siguiente resultado conocido para el cual Hopcroft y Karp [3] dieron una demostración en 1973.

**Proposición 2.10** *Sean  $m_1$  y  $m_2$  dos matchings en un grafo  $G$  con  $|m_1| < |m_2|$ . Entonces  $m_1 \Delta m_2$  contiene al menos  $|m_2| - |m_1|$  caminos  $m_1$ -aumentantes disjuntos.*

DEMOSTRACIÓN. (Proposición 2.9)

Consideremos la ejecución de  $\mathcal{A}$  en un grafo  $G$ . En esta demostración definiremos  $\tau_j^-$  como un instante de tiempo en que ya se recibió el  $j$ -ésimo vértice  $r_j$  pero *antes* de que  $\mathcal{A}$  realice una acción, y  $\tau_j^+$  el instante de tiempo *después* de que  $\mathcal{A}$  realice una acción con  $r_j$  pero antes de recibir el próximo vértice.

Demostremos por inducción sobre el orden de llegada que el matching construido hasta cada tiempo  $\tau_j^+$  no tiene caminos aumentantes de largo menor o igual a  $K$  (solo considerando los vértices de  $L$  y los que hayan llegado de  $R$ ).

El primer vértice en llegar siempre es emparejado (mientras tenga al menos un vecino). Supongamos ahora que en el tiempo  $\tau_j^+$  no existen caminos aumentantes de largo menor o igual a  $K$  y, por contradicción, supongamos que en el tiempo  $\tau_{j+1}^+$  existe un camino aumentante  $P = p_0 p_1 \dots p_l$  con  $l \leq K$  y  $p_0 \in R$ .

Si  $\mathcal{A}$  **no** empareja a  $r_{j+1}$  y  $r_{j+1} = p_0$ , entonces  $\mathcal{A}$  debió haber emparejado a  $r_{j+1}$  pues pudo haber usado  $P$  y  $\mathcal{A}$  es greedy. Si  $\mathcal{A}$  **no** empareja a  $r_{j+1}$  y  $r_{j+1} \neq p_0$ , entonces todas las aristas de  $P$  también existían en  $\tau_j^+$  y por lo tanto  $P$  también era un camino aumentante en dicho tiempo, lo que contradice la hipótesis de inducción.

Digamos ahora que  $\mathcal{A}$  **sí** empareja a  $r_{j+1}$  usando un camino aumentante  $Q = q_0 \dots q_m$  con  $q_0 = r_{j+1}$ .

Consideremos ahora los siguientes 3 emparejamientos en el grafo  $G_{j+1} := G[L \cup \{r_1, \dots, r_{j+1}\}]$ .

- $m_1$  es el emparejamiento construido al momento de recibir  $r_{j+1}$ , antes de utilizar  $Q$ .
- $m_2$  es el emparejamiento resultante de utilizar  $Q$ , es decir  $m_2 = m_1 \Delta Q$ .
- $m_3$  es el emparejamiento que resulta de aumentar  $m_2$  con  $P$ , que es un matching pues  $P$  es un camino  $m_2$ -aumentante. Es decir  $m_3 = m_2 \Delta P$ .

Vemos que  $|m_3| = |m_1| + 2$ . Luego por la proposición 2.10  $m_1 \Delta m_3$  contiene 2 caminos  $m_1$ -aumentantes disjuntos, digamos  $\bar{P}$  y  $\bar{Q}$ . Como la diferencia simétrica es asociativa y conmutativa tenemos que

$$m_1 \Delta m_3 = m_1 \Delta m_1 \Delta P \Delta Q = \emptyset \Delta P \Delta Q = P \Delta Q \quad (2.5)$$

Además observamos lo siguiente.

$$|\bar{P}| + |\bar{Q}| \leq |m_1 \Delta m_3| = |P \Delta Q| = |P| + |Q| - 2|P \cap Q| \leq |P| + |Q| \quad (2.6)$$

Notamos que de entre los vértices que forman  $P$  y  $Q$  los únicos que no están cubiertos por  $m_1$  son sus extremos  $p_0, q_0 \in R$  y  $p_l, q_m \in L$ . Luego solo estos vértices pueden ser los extremos de  $\bar{P}$  y  $\bar{Q}$ , y en particular  $p_0 \neq q_0$  pues  $\bar{P}$  y  $\bar{Q}$  son disjuntos. Digamos que  $\bar{P}$  comienza en  $p_0$  y  $\bar{Q}$  comienza en  $q_0$ .

Como  $p_0 \neq q_0$ , el camino  $m_1$ -aumentante  $\bar{P}$  existía en el tiempo  $\tau_j^+$ . Por la hipótesis de inducción tenemos que  $|\bar{P}| > K$  o equivalentemente  $|\bar{P}| \geq K + 2$ .

Por otro lado  $\bar{Q}$  era un camino disponible para emparejar  $q_0 = r_{j+1}$ , y como  $\mathcal{A}$  es SAP y eligió utilizar el camino  $Q$ , tenemos que  $|\bar{Q}| \geq |Q|$ . Por la desigualdad 2.6 concluimos que

$$|P| + |Q| \geq |\bar{P}| + |\bar{Q}| \geq K + 2 + |Q| \quad (2.7)$$

y por lo tanto  $|P| \geq K + 2$ , lo que contradice que  $|P| \leq K$ . Con esto se concluye la inducción y la demostración. □

Ya vimos que un emparejamiento maximal siempre es una  $\frac{1}{2}$ -aproximación, y que uno sin caminos aumentantes de largo 3 es además una  $\frac{2}{3}$ -aproximación. A continuación extenderemos estos resultados a emparejamientos sin caminos aumentantes de largo menor o igual a algún valor  $K$ , aunque solo para emparejamientos en grafos bipartitos.

El siguiente teorema junto con la proposición 2.6 son resultados ya conocidos. Hopcroft y Karp [3] dieron una demostración puramente combinatorial de este hecho para grafos generales (no solo bipartitos) en 1973. En este trabajo daremos una demostración que, hasta donde sabemos, es original y utiliza el esquema primal-dual.

**Teorema 2.11** *Sea  $G = (L, R, E)$  un grafo bipartito. Sea  $m$  un matching en  $G$  tal que no existen caminos  $m$ -aumentantes de largo menor o igual a  $K = 2n - 1$ . Entonces  $|m| \geq \frac{n}{n+1}\nu(G)$ .*

DEMOSTRACIÓN. Procediendo por el esquema de demostración primal-dual, construiremos un vector dual en  $G$  tal que:

- $\forall uv \in E, y_u + y_v \geq 1$
- Si  $v$  no está cubierto por  $m$ ,  $y_v = 0$
- $\forall uv \in m, y_u + y_v \leq \frac{n+1}{n}$

Sea  $\hat{G}_m$  una orientación de  $G$  (un grafo dirigido con los mismos vértices y aristas) tal que las aristas de  $m$  están orientadas desde  $L$  hasta  $R$ , y las aristas de  $E \setminus m$  van desde  $R$  hasta  $L$ . Así todo camino  $m$ -aumentante en  $G$  es un camino dirigido de  $R$  a  $L$  en  $\hat{G}_m$ .

Además, recordemos que un camino  $m$ -alternante en  $G$  es un camino que comienza en un vértice no cubierto por  $m$  y cuyas aristas alternan entre estar fuera y dentro de  $m$ .

Para  $v \in V$ , definimos  $d_v$  como largo del camino  $m$ -alternante más corto en  $G$  que termina en  $v$ . Notamos que  $d_v = 0$  si y solo si  $v$  no es cubierto por  $m$  y que  $d_v = \infty$  si no existe un camino alternante que termine en  $v$ . Se da un ejemplo de esto en la figura 2.6.

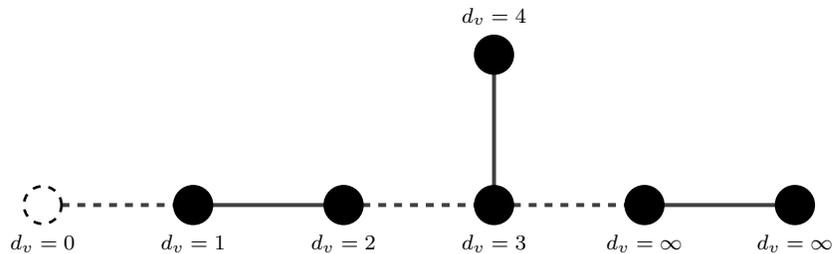


Figura 2.6: Ejemplo de valores de  $d_v$ .

Si  $d_v < \infty$ , denotaremos por  $P_v$  a un camino alternante que termine en  $v$  de largo mínimo. Notamos que si el extremo inicial de  $P_v$  es  $v_0$ ,  $P_v$  es un camino dirigido en  $\hat{G}_m$  entre  $v$  y  $v_0$  en alguna de las 2 direcciones. Como  $v_0$  no está cubierto por  $m$  la primera arista de  $P_v$  no está en  $m$  y por lo tanto la dirección de este camino dirigido está determinada por el conjunto al que pertenece  $v_0$ .

Sea  $f : \mathbb{N}_0 \cup \{\infty\} \rightarrow \mathbb{R}$  definida por

$$f(k) = \begin{cases} \frac{k}{2n} & \text{si } k \text{ es par y } k < n \\ 1 - \frac{k-1}{2n} & \text{si } k \text{ es impar y } k < n \\ \frac{1}{2} & \text{si } k \geq n \text{ o } k = \infty \end{cases} \quad (2.8)$$

El valor de las variables duales  $y_v$  está dado por  $y_v := f(d_v)$ . Notamos las siguientes propiedades.

1.  $f$  es decreciente cuando se restringe a los impares.
2. Para todo  $k$ ,  $f(k) + f(k+1) \in [1, \frac{n+1}{n}]$ . Es decir si  $d_u$  y  $d_v$  son consecutivos,  $y_u + y_v \in [1, \frac{n+1}{n}]$ .
3. Si  $k \geq n$  o  $k$  es impar,  $f(k) \geq \frac{1}{2}$ .
4.  $d_v$  es par si y solo si  $P_v$  termina con una arista de  $m$ .

Las propiedades 1., 3. y 4. son fáciles de verificar. Para verificar 2. se puede observar que  $f$  satisface la siguiente relación de recurrencia para  $k \leq n$

$$f(0) = 0 \quad (2.9)$$

$$f(k) = 1 + \frac{1 + (-1)^k}{2n} - f(k-1) \quad (2.10)$$

y que  $2f(n) \in [1, \frac{n+1}{n}]$ .

Sea  $uv \in E \setminus m$ , probaremos que  $y_u + y_v \geq 1$ . Supongamos buscando una contradicción que  $y_u + y_v < 1$ .

Si uno de los 2 vértices (digamos  $u$ ) no está cubierto por  $m$ ,  $v$  debe estarlo pues  $m$  es maximal, lo que implicaría  $d_u = 0$  y  $d_v = 1$ , y por lo tanto  $y_u = 0$  y  $y_v = 1$  lo que contradice  $y_u + y_v < 1$ . Por lo tanto  $u$  y  $v$  deben ambos estar cubiertos por  $m$ . Como  $uv \notin m$ , están cubiertos por aristas distintas  $au \in m$  y  $bv \in m$ . Como  $y_u + y_v < 1$ , uno de los dos valores  $y_u$  y  $y_v$  debe ser menor que  $\frac{1}{2}$ . Digamos  $y_u < \frac{1}{2}$ , entonces por 3.  $d_u < n$  y es par. Luego  $P_u$  existe y por 4. termina con la arista  $au$ . Como  $uv \notin m$  el camino  $P_u v$  es alternante y por lo tanto  $P_v$  también existe y  $d_v \leq d_u + 1$ . Por 2.  $d_u$  y  $d_v$  no pueden ser consecutivos. Si  $d_v \leq d_u - 2$  y  $d_v$  es par, entonces  $P_v u$  sería un mejor camino alternante que  $P_u$ . Si  $d_v \leq d_u - 2$  y  $d_v$  es impar, entonces por 1. y por 2.  $y_v + y_u \geq f(d_u - 1) + f(d_u) \geq 1$ . Luego  $d_v = d_u < n$ .

Sea ahora  $u_0$  el punto inicial de  $P_u$  y  $v_0$  el de  $P_v$ , veremos que siempre debe haber un camino  $m$ -aumentante entre  $u_0$  y  $v_0$  de largo menor o igual a  $2n - 1$ , lo que contradice la hipótesis del enunciado. Si  $P_u$  y  $P_v$  son disjuntos forman el camino buscado junto a la arista

$uv$ . Supongamos ahora que  $P_u$  y  $P_v$  se intersectan, y digamos que  $u \in L$  y  $v \in R$ . Como tanto  $d_v$  como  $d_u$  son pares,  $u_0 \in L$  y  $v_0 \in R$ . Viendo los caminos en  $\hat{G}_m$ , como son alternantes ambos son caminos dirigidos.  $P_u$  apunta desde  $u$  a  $u_0$  mientras que  $P_v$  apunta de  $v_0$  a  $v$ . Si ahora tomamos el camino dirigido que comienza en  $v_0$ , viaja por  $P_v$  hasta el primer punto de intersección con  $P_u$ , y continua por  $P_u$  hasta  $u_0$ , este es un camino dirigido de  $v_0$  a  $u_0$  en  $\hat{G}_m$  y por lo tanto es un camino  $m$ -aumentante en el grafo original. Además como solo usa aristas de  $P_u$  y  $P_v$  tiene largo menor a  $2n - 1$ . Esto contradice la hipótesis de la proposición y por lo tanto  $y_u + y_v \geq 1$  para toda arista  $uv$  fuera de  $m$ .

Sea ahora  $uv \in m$ , probaremos que  $y_u + y_v \in [1, \frac{n+1}{n}]$ . Supongamos buscando una contradicción lo contrario. Los valores  $d_v$  y  $d_u$  no pueden ambos ser  $\infty$  pues  $f(\infty) = \frac{1}{2}$ . Digamos que  $d_u$  es finito, es decir existe  $P_u$ .

Supongamos que  $d_u$  es par. Es decir  $P_u$  termina con la arista  $uv \in m$  y por lo tanto  $P_u - u$  es un camino alternante válido para  $v$ , por lo que  $P_v$  existe y  $d_v < d_u$ . El mismo argumento muestra que  $d_v$  no puede ser par a la vez pues ello implicaría  $d_v > d_u$ , así que  $d_v$  es impar y  $P_v$  termina con una arista fuera de  $m$ . Como el camino válido más corto para  $u$  termina con  $uv \in m$  y el para  $v$  con una arista fuera de  $m$ , necesariamente  $d_u = d_v + 1$ , lo que es una contradicción por la propiedad 2.

Luego  $d_u$  es impar. Esto implica que  $P_u$  termina con una arista fuera de  $m$  y por lo tanto  $P_u v$  es un camino válido para  $v$ . Por lo tanto  $P_v$  existe y  $d_v < \infty$ , y por el mismo argumento anterior  $d_v$  debe ser impar. Además si alguno de los dos valores fuera menor, digamos  $d_v \leq d_u - 2$ , entonces  $P_v u$  sería un mejor camino que  $P_u$  así que  $d_v = d_u$ . Además como  $f(k) = \frac{1}{2}$  para todo  $k \geq n$ , tenemos que  $d_u = d_v < n$ . Concluimos con el mismo argumento que para  $uv \notin m$ , mostrando que entre los caminos  $P_u$ ,  $P_v$  y la arista  $uv$  siempre debe haber un camino aumentante de largo menor o igual a  $2n - 1$ . Esto contradice la hipótesis del enunciado y por lo tanto  $y_u + y_v \in [1, \frac{n+1}{n}]$ .

□

La proposición 2.9 junto con el teorema 2.11 nos permiten concluir lo siguiente.

**Corolario 2.12** *Sea  $K = 2n - 1$  con  $n \in \mathbb{N}$ . Cualquier algoritmo determinista greedy y SAP para OMP- $K$  es  $\frac{n}{n+1}$ -competitivo.*

### 2.3. Problemas con caminos de largo 3

En esta sección estudiaremos algunos de los problemas de la familia OMP-3- $(s, t)$  con  $s, t \in \mathbb{N} \cup \{\infty\}$ . Recordemos que esto quiere decir:

- Pueden usarse caminos aumentantes de largo 1 o 3.
- Los vértices del lado *offline*  $L$  tienen permitidos  $s$  cambios cada uno.
- Los vértices del lado *online*  $R$  tienen permitidos  $t$  cambios cada uno.

Al usar un camino aumentante  $P$  para aumentar el matching, todos los vértices interiores de  $P$  incurren cambios, y por lo tanto mientras más corto sea el camino usado, menos cambios se realizan. Como hay un número máximo de cambios permitidos para cada vértice, es razonable utilizar nuevamente algoritmos SAP.

Primero estudiaremos el desempeño de un algoritmo determinista, greedy y SAP específicamente en el caso  $s = t = 1$ . Damos dos observaciones importantes. Primero, a diferencia de OMP-3, un algoritmo *greedy* para este problema no necesariamente resulta en un emparejamiento libre de caminos aumentantes de largo 3. En efecto, es posible que el emparejamiento construido tenga caminos aumentantes de largo 3 en donde uno o ambos de sus vértices interiores estén congelados. Debido a esto no es posible utilizar la misma asignación dual que en la proposición 2.6.

Segundo, al utilizar un 3-camino aumentante  $ijkl$ ,  $j$  y  $k$  son agotados por lo que los cuatro vértices del camino quedan congelados. Además, la única forma en que dicho camino puede ser utilizado es que  $j \in R$  haya sido emparejado directamente a  $k \in L$  al llegar.

**Proposición 2.13** *Sea  $\mathcal{A}$  un algoritmo greedy y SAP para OMP-3-(1,1). Entonces  $\mathcal{A}$  es  $\frac{3}{5} = 0,60$ -competitivo.*

DEMOSTRACIÓN. Sea  $\mathcal{A}$  un algoritmo como el descrito y un grafo bipartito  $G = (L, R, E)$ . Consideremos una ejecución de  $\mathcal{A}$  en  $G$ , y construiremos en paralelo un vector  $(y_v)_{v \in V} = (\alpha_i, \beta_j)_{i \in L, j \in R}$  que sea D-factible y tal que al agregar una arista al matching, el valor de este vector dual (la suma de todas las variables) aumente en  $\frac{5}{3}$ .

La asignación de las variables duales es como sigue:

- Todas las variables se inicializan en 0.
- Si al llegar  $j$  se empareja directamente a  $i$ , entonces  $\beta_j = 1$  y  $\alpha_i = \frac{2}{3}$  como en la Figura 2.7



Figura 2.7:  $j$  llega y se empareja directamente con  $i$ .

- Si al llegar  $l \in R$  se aumenta el matching con un camino  $ijkl$ , se fijan  $\alpha_i = \alpha_k = \beta_j = 1$  y  $\beta_l = \frac{1}{3}$  como en la Figura 2.8. Notamos que necesariamente  $j \in R$  debe haberse emparejado directamente a  $k$  cuando llegó, y por lo tanto el valor dual total aumenta en exactamente  $\frac{5}{3}$ .
- Al terminar, se revisan todas las aristas  $ij$  en el matching que no tengan extremos agotados (es decir que  $j$  al llegar se haya emparejado directamente a  $i$ ), y que por lo tanto  $\alpha_i = \frac{2}{3}$  y  $\beta_j = 1$ . Si todos los vecinos de  $j$  fueron emparejados, entonces estos valores se invierten a  $\alpha_i = 1$  y  $\beta_j = \frac{2}{3}$  como en la Figura 2.9.

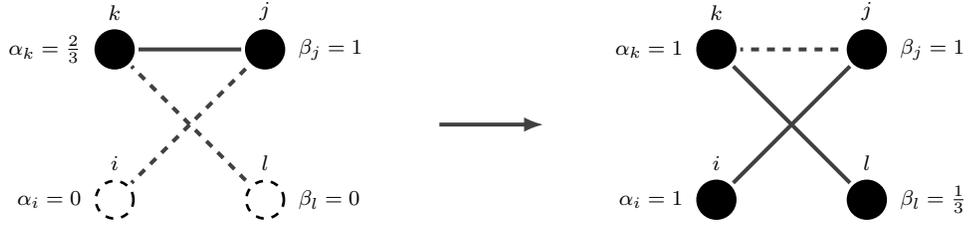


Figura 2.8:  $l$  llega y se empareja usando un camino aumentante  $ijkl$ .



Figura 2.9: Al terminar la ejecución, si  $j$  se emparejó directamente a  $i$ ,  $ij$  queda en el matching final y todos los vecinos de  $j$  fueron emparejados, se intercambian los valores de  $\alpha_i$  y  $\beta_j$ .

Veamos ahora que el vector resultante es factible al finalizar. Sea una arista  $ij \in E$ , examinaremos cada uno de los posible pares de valores para  $(\alpha_i, \beta_j)$  que tienen una suma menor a 1.

- $(0, 0)$ : Solo ocurre si hay 2 vecinos sin emparejar, lo que no puede ocurrir pues  $\mathcal{A}$  es greedy y por lo tanto resulta en un matching maximal.
- $(0, \frac{2}{3})$ :  $\beta_j = \frac{2}{3}$  implica que todos los vecinos de  $j$  están emparejados por lo que  $\alpha_i$  no puede ser 0.
- $(0, \frac{1}{3})$ :  $\beta_j = \frac{1}{3}$  implica que  $j$  usó un camino de largo 3 al llegar. Como  $\mathcal{A}$  es SAP, todos los vecinos de  $j$  deben haber estado ocupados por lo que  $\alpha_i$  no puede ser 0.
- $(\frac{2}{3}, 0)$ : Sea  $l$  la pareja actual de  $i$ . Como  $\alpha_i = \frac{2}{3}$   $l$  debe haberse emparejado directamente con  $i$  al llegar, y  $l$  debe tener un vecino disponible, digamos  $k$ . Además  $j$  debe llegar después que  $l$  pues de lo contrario  $i$  hubiese estado disponible y  $j$  se hubiese emparejado. Luego al llegar  $j$  tiene el camino  $kjil$  disponible, por lo que  $j$  debe ser asignado y  $\beta_j$  no puede ser 0.

Luego  $\alpha_i + \beta_j \geq 1$  y el vector dual es  $D_G$ -factible.  $\square$

A continuación extenderemos este resultado al caso  $s = 1$  y  $t \in \mathbb{N}$  libre, aunque primero necesitaremos un resultado general para los algoritmos greedy y SAP de todos los problema de la familia OMP-3- $(s, t)$ .

**Proposición 2.14** Sean  $s, t \in \mathbb{N} \cup \{\infty\}$ . Sea  $\mathcal{A}$  un algoritmo greedy y SAP para OMP-3- $(s, t)$ ,  $G$  una instancia y  $m$  el matching resultante en  $G$ . Sea además una arista  $kj \in m$ .

Si al terminar la ejecución de  $\mathcal{A}$ ,  $k$  ni  $j$  están agotados y existe  $i \in N(j)$  que no fue emparejado, entonces todos los vecinos de  $k$  fueron emparejados. En otras palabras, si  $k$  ni  $j$  están agotados, la arista  $kj$  **no** puede ser la arista central de un 3-camino  $m$ -aumentante.

**Observación:** La proposición 2.14 es más general que la proposición 2.5 pues si  $s = t =$

$\infty$ , ningún vértice quedará agotado y por lo tanto no habrán caminos aumentantes de largo 3.

DEMOSTRACIÓN. La demostración será similar a la de la proposición 2.5. Supongamos que al terminar la ejecución de  $\mathcal{A}$  existen

- Un vértice  $i \in L$  sin emparejar.
- Una arista  $kj \in m$  con  $i \in N(j)$ , y con  $k$  y  $j$  sin agotar.

Supongamos ahora, buscando una contradicción, que existe además un vértice sin emparejar  $l \in N(k)$ .

Consideremos el instante en que  $\mathcal{A}$  recibe  $l \in R$ . Como  $l \in R$  no fue emparejado, sabemos que en ese instante

1.  $k \in N(l)$  estaba emparejado pues de lo contrario  $\mathcal{A}$  podría haber emparejado  $j$ .
2. la arista  $kj$  no estaba en el matching construido hasta el momento pues de lo contrario, como  $k$  y  $j$  no fueron agotados,  $\mathcal{A}$  podría haber emparejado  $j$  usando el camino  $ijkl$

Digamos que en este instante  $k$  estaba emparejado a  $x \in R \setminus \{j\}$ . Como  $l$  no se pudo emparejar, en este instante  $x$  estaba agotado o todos sus vecinos estaban emparejados. De cualquier forma la arista  $kx$  no puede salir de matching, lo que contradice que en un tiempo posterior  $k$  debe estar emparejado a  $j$ .

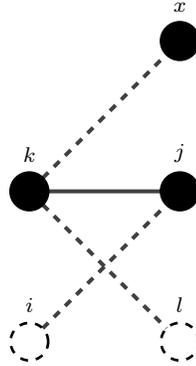


Figura 2.10: Estado al terminar la ejecución de  $\mathcal{A}$ .

□

Usando la proposición 2.14 daremos uno de los resultados principales de este trabajo.

**Teorema 2.15** *Sea  $t \in \mathbb{N} \cup \{\infty\}$ . Sea  $\mathcal{A}$  un algoritmo greedy y SAP para OMP-3-(1, t). Entonces  $\mathcal{A}$  es  $\frac{2t+1}{3t+2}$ -competitivo ( $\frac{2}{3}$ -competitivo si  $t = \infty$ ).*

DEMOSTRACIÓN. Sea  $t \in \mathbb{N}$ . Sea  $F$  (por determinar) la tasa que buscamos alcanzar, y sea  $\lambda$  tal que  $\frac{1}{F} = 1 + \lambda$ . Supondremos  $\frac{1}{2} \leq \lambda \leq 1$  y por lo tanto  $\frac{1}{2} \leq F \leq \frac{2}{3}$ . Agregaremos una variable adicional  $r_j \geq 0$  para cada  $j \in R$  que cumple el rol de *reservar* valor dual disponible que no

sea usado por las demás variables. La asignación dual mantendrá las siguientes invariantes a lo largo de la ejecución del algoritmo.

1. Si  $i \in L$  es emparejado, entonces  $\alpha_i \geq \lambda$ .
2. Si  $j \in R$  es emparejado, entonces  $\beta_j \geq 1 - \lambda$ . Notar que  $\lambda \geq 1 - \lambda$ .
3. Si una arista  $ij$  está en el matching y puede ser usada para una aumentación (es decir sus extremos **no** están agotados), entonces  $\alpha_i = \lambda$  y  $\beta_j = 1$  (hasta el último paso).
4. Si una arista  $ij$  está en el matching y cualquiera de sus 2 extremos se agota, entonces  $\alpha_i = 1$ .

Asignaremos las variables duales como sigue.

- Todas las variables se inicializan en 0.
- Si  $j \in R$  se empareja a  $i \in L$  directamente,  $\alpha_i = \lambda$  y  $\beta_j = 1$  como en la Figura 2.11.

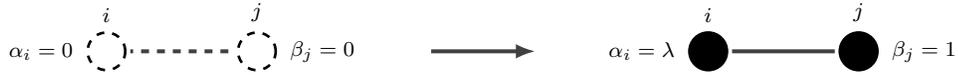


Figura 2.11:  $j$  llega y se empareja directamente a  $i$ .

- Si se utiliza un camino  $ijkl$ , entonces  $\alpha_i = \lambda$ ,  $\beta_j = 1$ ,  $\alpha_k = 1$  y  $\beta_l = 1 - \lambda$  como en la Figura 2.12. Como la arista  $kj$  estaba disponible para una aumentación, ésta tenía un valor dual total de  $1 + \lambda$ , y por lo tanto esta asignación solo aumenta el valor dual en  $2 - \lambda = 1 + (1 - \lambda)$  en vez del máximo permitido de  $1 + \lambda$ . Agregamos esta diferencia a  $r_j$ , que aumenta en  $2\lambda - 1$ .

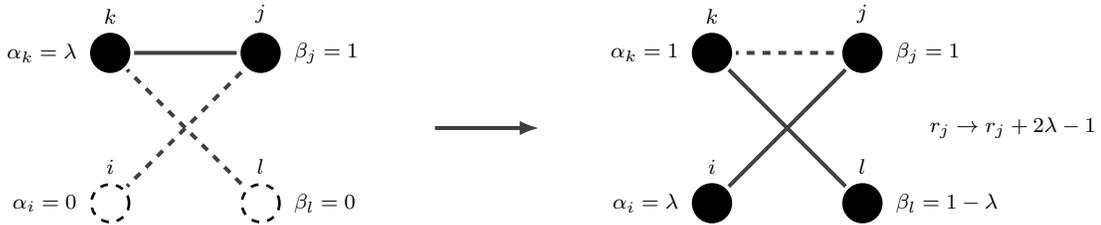


Figura 2.12:  $l$  llega y se empareja usando un camino aumentante  $ijkl$ .

- Si al usar un camino  $ijkl$  el vértice  $j$  es agotado (es decir fue reasignado por su  $t$ -ésima vez), entonces se aumenta el valor de  $\alpha_i$  en  $1 - \lambda$  para llegar a 1, como en la Figura 2.13. El valor adicional necesario se descuenta de  $r_j$ , por lo que deberemos fijar  $\lambda$  tal que  $r$  haya acumulado lo suficiente tras los  $t$  cambios.
- Al terminar, se revisa cada arista  $ij$  en el matching final cuyos extremos no hayan sido agotados, y por lo tanto  $\alpha_i = \lambda$  y  $\beta_j = 1$ . Si todos los vecinos de  $j$  fueron emparejados, entonces se actualizan los valores a  $\alpha_i = 1$  y  $\beta_j = \lambda$  como en la Figura 2.14.

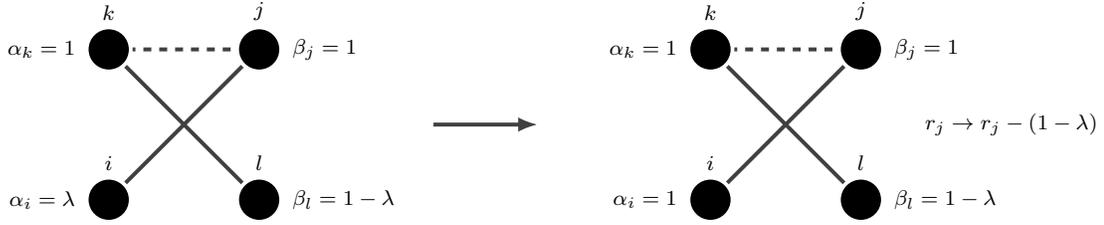


Figura 2.13: Se utiliza un camino aumentante  $ijkl$  que resulta en agotar los  $t$  cambios permitidos de  $j$ .



Figura 2.14: Al terminar la ejecución, si  $ij$  queda en el matching, sus extremos no están congelados y todos los vecinos de  $j$  están emparejados, entonces se intercambian los valores de  $\alpha_i$  y  $\beta_j$ .

Sea una arista  $ij \in E$ , verificaremos caso por caso que  $\alpha_i + \beta_j \geq 1$ .

**Caso 1:** Ambos extremos  $i$  y  $j$  son emparejados, concluimos por 1. y 2.

**Caso 2:** El extremo  $i \in L$  no está emparejado, así que  $\alpha_i = 0$ .

- $\beta_j \neq 0$  pues  $\mathcal{A}$  es greedy y por lo tanto resulta en un matching maximal.
- $\beta_j \neq 1 - \lambda$  pues significaría que  $j$  usó un camino aumentante al llegar, pero  $i$  estaba disponible y  $\mathcal{A}$  es SAP.
- $\beta_j \neq \lambda$  pues quedó un vecino de  $j$  sin emparejar.

Luego  $\beta_j = 1$  y  $\alpha_i + \beta_j = 1$ .

**Caso 3:** El extremo  $j \in R$  no está emparejado, por lo que  $\beta_j = 0$ . Como el matching es maximal  $i$  debe estar emparejado, sea  $x \in R$  la pareja final de  $i$ . Si  $i$  o  $x$  están agotados, por 4. tenemos que  $\alpha_i = 1$ . De lo contrario, la arista  $ix$  está disponible para realizar un intercambio e  $i$  tiene un vecino libre en  $j$ . Por la proposición 2.14  $x$  no puede tener vecinos sin emparejar. Es decir la arista  $ix$  tuvo sus valores invertidos en el último paso de la asignación y  $\alpha_i = 1$ . De todas formas  $\alpha_i + \beta_j = 1$ .

Por último, determinaremos el valor de  $\lambda$  necesario. Cada vez que un vértice  $j \in R$  sea reasignado,  $r_j$  aumenta en  $2\lambda - 1$ , y necesitamos que tras  $t$  cambios del mismo vértice  $j$  haya acumulado  $1 - \lambda$ . Fijaremos  $\lambda$  tal que  $t(2\lambda - 1) = 1 - \lambda$ , lo que resulta en  $\lambda = \frac{1+t}{2t+1}$  y  $F = \frac{2t+1}{3t+2}$ .

Para el caso  $t = \infty$  basta tomar desde el principio  $\lambda = \frac{1}{2}$ . Notamos que de esta forma todos los valores de  $r_j$  se mantienen en 0, pero como  $j \in R$  nunca se agotará no es necesario acumular valor dual en ellos.

□

**Observación:** Un algoritmo *greedy* y SAP para OMP-3-(1,  $\infty$ ) alcanza una tasa com-

petitiva de  $\frac{2}{3}$ , idéntica a la que un algoritmo *greedy* en OMP-3- $(\infty, \infty)$  y, como veremos a continuación, el mejor desempeño que cualquier algoritmo determinista para un problema de la familia OMP-3- $(s, t)$  podría alcanzar.

# Capítulo 3

## Cotas superiores

En el inicio de la sección anterior vimos que ningún algoritmo determinista para el problema de emparejamiento en línea original puede superar una tasa competitiva de  $\frac{1}{2}$ , en otras palabras, para cualquier algoritmo determinista  $\mathcal{A}$  para OMP existe una instancia que contiene un matching de tamaño 2, pero en la que  $\mathcal{A}$  solo construye un matching de tamaño 1.

En esta sección daremos resultados de este estilo para algunos de los problemas estudiados anteriormente. Primero veremos el problema OMP-3- $(\infty, \infty)$ .

**Proposición 3.1** *Ningún algoritmo determinista para OMP-3 puede superar una tasa competitiva de  $\frac{2}{3}$ .*

**Corolario 3.2** *Sean  $s, t \in \mathbb{N} \cup \{\infty\}$ . Ningún algoritmo determinista para OMP-3- $(s, t)$  puede superar una tasa competitiva de  $\frac{2}{3}$ .*

DEMOSTRACIÓN (PROPOSICIÓN). Consideremos la siguiente familia de 4 grafos bipartitos de orden 6.

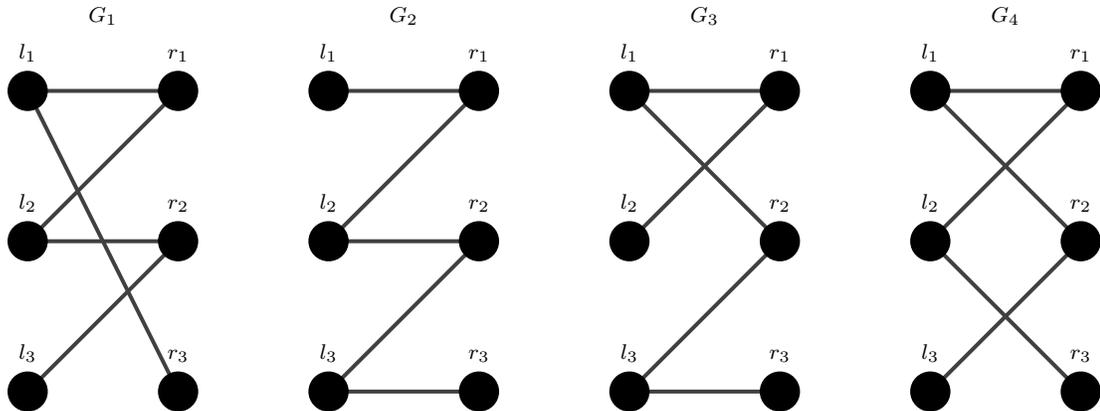


Figura 3.1: Cuatro grafos bipartitos de orden 6.

Se puede verificar que los cuatro grafos contienen un matching de tamaño 3. Consideremos además una transformación  $\phi$  que intercambia las etiquetas de los vértices  $l_1$  y  $l_2$ . Observamos que  $\phi$  convierte  $G_1$  en  $G_4$ ,  $G_4$  en  $G_1$ ,  $G_2$  en  $G_3$  y  $G_3$  en  $G_2$ . Es decir  $\phi$  aplicada a la familia  $\{G_1, G_2, G_3, G_4\}$  resulta en la misma familia. Sea ahora  $\mathcal{A}$  un algoritmo determinista para OMP-3 y consideremos la ejecución de este algoritmo en los 4 grafos anteriores. Si en alguna instancia  $\mathcal{A}$  es capaz de emparejar un vértice  $j \in R$  pero decide no hacerlo concluimos inmediatamente que no puede construir un matching perfecto en dicha instancia y por lo tanto no podrá superar una tasa competitiva de  $\frac{2}{3}$ , por lo que podemos suponer que  $\mathcal{A}$  es greedy.

En todas las instancias el primer vértice en llegar  $r_1 \in R$  tiene como vecinos a  $l_1$  y  $l_2 \in L$ . Como  $\mathcal{A}$  es determinista, debe tomar la misma decisión al recibir el primer vértice de cada grafo, y por la discusión anterior sobre la transformación  $\phi$  podemos suponer sin pérdida de generalidad que decide emparejarlo a  $l_1 \in L$ . Consideremos ahora, tras esta decisión, la llegada del segundo vértice  $r_2 \in R$  en  $G_3$  y  $G_4$ . El algoritmo tiene 2 opciones, emparejarlo directamente a  $l_3 \in L$ , resultando en el emparejamiento  $\{l_1r_1, l_3r_2\}$  o emparejarlo a  $l_2 \in L$  cambiando  $r_1 \in R$  a  $l_2 \in L$ , resultando en el emparejamiento  $\{l_1r_2, l_2r_1\}$ . En el primer caso no se puede lograr un matching perfecto en  $G_3$  y en el segundo no se puede en  $G_4$ . Luego ningún algoritmo determinista para OMP-3 puede lograr un matching perfecto en los 4 grafos a la vez, y por lo tanto ninguno puede superar una tasa competitiva de  $\frac{2}{3}$ .  $\square$

En la demostración anterior dimos explícitamente una lista de instancias y argumentamos que para cualquier algoritmo determinista existe al menos una instancia en la lista donde construye un matching de tamaño 2 o menor. Este método se vuelve inviable para otros problemas debido a que la lista de grafos necesaria y el orden de los grafos que la forman crecen rápidamente. Otra forma de lograr el mismo resultado evitando dichas complicaciones es, dado un algoritmo determinista  $\mathcal{A}$ , construir paso a paso de manera *online* un grafo basándose en las decisiones tomadas por  $\mathcal{A}$ , tal que el algoritmo tenga un *mal* desempeño.

A continuación se da una demostración alternativa de la Proposición 3.1 usando este método.

DEMOSTRACIÓN (PROPOSICIÓN 3.1). Dado  $\mathcal{A}$  un algoritmo para OMP-3, construiremos una instancia con un matching de tamaño 3 en la cual  $\mathcal{A}$  logra a lo más un matching de tamaño 2. Como buscamos que  $\mathcal{A}$  fallé en emparejar al menos 1 vértice, nuevamente podemos suponer que es greedy.

La instancia se construye como sigue.

- $L = \{l_1, l_2, l_3\}$ .
- Al recibir  $r_1$ ,  $N(r_1) = \{l_1, l_2\}$ . Sin pérdida de generalidad, re-etiquetando vértices si es necesario, suponemos que  $\mathcal{A}$  empareja  $r_1$  a  $l_2$ .
- Al recibir  $r_2$ ,  $N(r_2) = \{l_2, l_3\}$ . Sea  $u \in L$  el vértice de  $L$  que  $\mathcal{A}$  agrega al emparejamiento.
- Al recibir  $r_3$ ,  $N(r_3) = \{u\}$ .

Como se ve en la figura 3.2, en todos los casos  $\mathcal{A}$  será incapaz de emparejar a  $r_3$ , por lo que no puede superar una tasa de  $\frac{2}{3}$ . Observamos que si consideramos todos los grafos posibles de generar según las diferentes decisiones que puede tomar  $\mathcal{A}$ , obtenemos la misma familia de 4 grafos que en la demostración original.

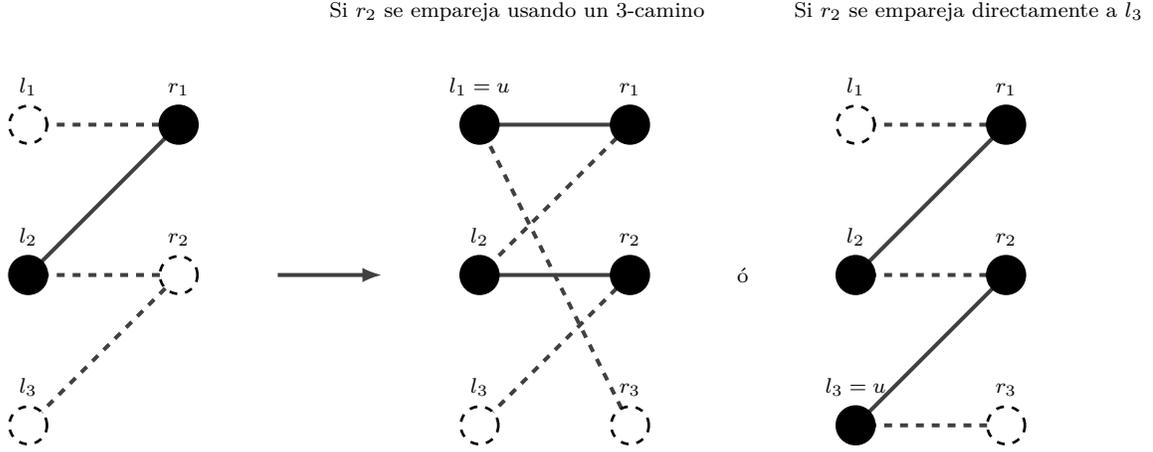


Figura 3.2: Construcción de instancia en la que  $\mathcal{A}$  logra solo un matching de tamaño 2.

□

A continuación usamos este método para dar una cota superior al desempeño de algoritmos deterministas para OMP- $K$ .

**Proposición 3.3** Sea  $K = 2n - 1$  con  $n \in \mathbb{N}$ . Ningún algoritmo determinista para OMP- $K$  puede superar una tasa competitiva de  $\frac{n}{n+1}$ .

DEMOSTRACIÓN. Sea  $\mathcal{A}$  un algoritmo determinista para OMP- $K$ . Construiremos una instancia con un emparejamiento perfecto de tamaño  $n + 1$  en la cual  $\mathcal{A}$  sea a lo más  $\frac{n}{n+1}$ -competitivo. Al igual que antes supondremos que  $\mathcal{A}$  es *greedy* pues de lo contrario pues de lo contrario su tasa competitiva no puede superar  $\frac{n}{n+1}$ .

- $L = \{l_1, l_2, \dots, l_n, l_{n+1}\}$ .
- Al recibir  $r_1$ ,  $N(r_1) = \{l_1, l_2\}$ . Sin pérdida de generalidad, re-etiquetando vértices si es necesario, suponemos que  $\mathcal{A}$  empareja  $r_1$  a  $l_2$ . Definimos  $u_2 = l_2$ .
- Para  $k = 2, \dots, n$ , al recibir  $r_k$  fijamos  $N(r_k) = \{l_{k+1}, u_k\}$ . Definimos  $u_{k+1}$  como el vértice de  $L$  que  $\mathcal{A}$  agregue al emparejamiento al emparejar  $r_k$  (el extremo en  $L$  del camino aumentante que se utilice).
- Al recibir  $r_{n+1}$ ,  $N(r_{n+1}) = \{u_{n+1}\}$ .

Para  $k = 1, \dots, n$ , sea  $G_k$  el grafo construido tras recibir el vértice  $r_k \in R$ . Notamos que para cada  $k = 1, \dots, n$ , el grafo  $G_k$  consiste de las siguientes componentes conexas.

- Un camino de largo  $2k$  que cubre todos los vértices  $\{l_1, \dots, l_{k+1}, r_1, \dots, r_k\}$ .

- Los vértices  $l_i$  con  $i \geq k + 2$ , cada uno un vértice aislado.

Luego al recibir  $r_{n+1}$ , el grafo resultante es un camino que cubre todos los vértices del grafo, y el matching construido hasta el momento es de tamaño  $n$  por lo que este es en realidad un camino aumentante. Como el grafo es de orden  $2n + 2$ , este camino es de largo  $2n + 1$  y  $\mathcal{A}$  no será capaz de emparejar  $r_{n+1}$ . Luego  $\mathcal{A}$  construye un matching de tamaño  $n$  pero el grafo resultante contenía un matching perfecto de tamaño  $n + 1$ , con lo que se concluye la demostración.  $\square$

Para terminar, daremos una cota superior para el problema OMP-3-(1, 1).

**Proposición 3.4** *Ningún algoritmo determinista para OMP-3-(1, 1) puede superar una tasa competitiva de  $\frac{3}{5}$ .*

DEMOSTRACIÓN. Recordemos que en esta versión del problema de emparejamiento en línea, cada vértice puede cambiar de pareja a lo más 1 vez.

Sea  $\mathcal{A}$  un algoritmo determinista para OMP-3-(1, 1). Ahora buscamos construir un grafo bipartito con un matching perfecto de tamaño 5 en donde  $\mathcal{A}$  no logre emparejar 2 vértices, es decir que sea a lo más  $\frac{3}{5}$ -competitivo. Ya no podemos suponer que  $\mathcal{A}$  es greedy pues podría dejar pasar un vértice sin emparejar y aun así construir un matching de tamaño 4, pero por simplicidad primero nos restringiremos a algoritmos greedy de todas formas.

La instancia se construye como sigue.

- $L = \{l_1, l_2, l_3, l_4, l_5\}$
- Al recibir  $r_1$ , fijamos  $N(r_1) = L$ . Sin pérdida de generalidad  $\mathcal{A}$  empareja  $r_1$  a  $l_1$ .
- Al recibir  $r_2$ , fijamos  $N(r_2) = L \setminus \{l_1\}$ . Sin pérdida de generalidad  $\mathcal{A}$  empareja  $r_2$  a  $l_2$ .
- Al recibir  $r_3$ , fijamos  $N(r_3) = \{l_1, l_2\}$ . Ambos vecinos están emparejados, por lo que  $\mathcal{A}$  debe elegir un 3-camino para emparejar  $r_3$ . Sea  $u$  el extremo en  $L$  de este camino y  $v$  el vértice interior  $L$ .
- Al recibir  $r_4$ ,  $N(r_4) = u$ .
- Al recibir  $r_5$ ,  $N(r_5) = v$ .

Los vértices  $r_4$  y  $r_5$  no pueden ser emparejados pues en el caso  $s = t = 1$  al usar un 3-camino, los 4 vértices involucrados no pueden volver a cambiar su pareja en la ejecución. En la figura 3.3 se muestra un ejemplo de la instancia generada cuando  $r_3$  se empareja usando el camino  $r_3l_2r_2l_3$  por lo que los vértices  $l_2$  y  $l_3$  no pueden ser re-asignados nuevamente. Se observa además que esta instancia contiene un matching perfecto y que por lo tanto  $\mathcal{A}$  es a lo más  $\frac{3}{5}$ -competitivo. Notar que por claridad dibujamos los vértices  $r_1$  y  $r_2$  a la izquierda de los de  $L$ .

Consideremos ahora un algoritmo determinista cualquiera  $\mathcal{A}$ , que no es necesariamente greedy. La construcción procede igual que la anterior hasta el momento en que se reciba un vértice  $r_i$  que es posible de emparejar, pero  $\mathcal{A}$  decide no hacerlo. Podemos suponer que

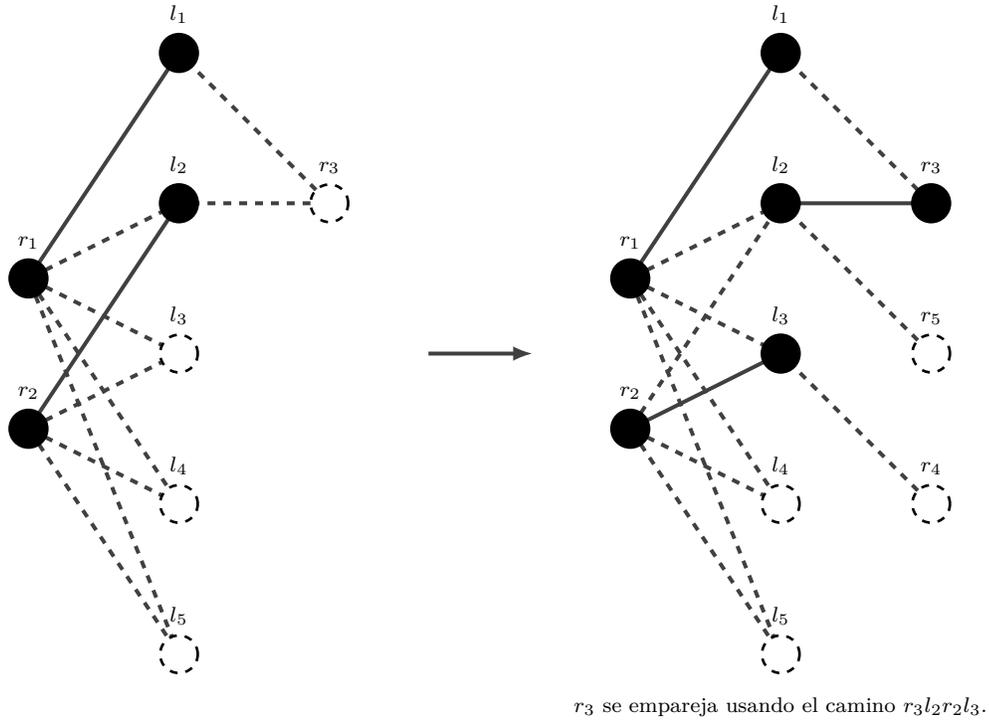


Figura 3.3: Instancia en la que  $\mathcal{A}$  para OMP-3-(1, 1) logra un matching de tamaño 3.

tal vértice existe pues de lo contrario concluimos con la misma instancia anterior. Además notamos que necesariamente  $i \in \{1, 2, 3\}$ . Desde este punto suponemos que  $\mathcal{A}$  emparejará todo vértice que sea posible de emparejar pues de lo contrario construirá un matching de a lo más tamaño 3, y es fácil terminar la instancia para asegurarse que exista un matching perfecto (por ejemplo para todo vértice restante  $j \in R$  asignar  $N(j) = L$ ).

Para concluir la construcción, asignamos a  $r_{i+1}$  los mismos vecinos que  $r_i$  y continuamos con la asignación original con  $r_{i+1}$  como el nuevo  $r_i$ , que por la discusión anterior sí será emparejado. De esta forma  $r_4$  será emparejado usando un 3-camino, y  $r_5$  tendrá como único vecino el extremo en  $L$  de dicho camino por lo que no podrá ser emparejado, resultando en un matching de tamaño 3. Se puede verificar además que para todo valor de  $i$ , esta construcción resulta en un grafo que contiene un matching perfecto.

□

# Capítulo 4

## Pesos en el lado *offline*

En esta sección estudiaremos brevemente una variante del problema OMP-3-(1,1) en la cual cada vértice  $i$  del lado *offline*  $L$  tiene asociado un peso no-negativo  $w(i) \geq 0$ . El emparejamiento se puede construir de la misma forma que en OMP-3-(1,1), usando caminos aumentantes de largo a lo más 3 y un cambio permitido por vértice, y el objetivo es maximizar el peso total del subconjunto de vértices de  $L$  que son emparejados.

Abusaremos de notación y cada vez que  $M \subseteq E$  es un matching, usaremos la convención  $M: V \rightarrow V \cup \{\perp\}$ , donde  $M(u) = v$  si  $uv \in M$ , y  $M(v) = \perp$  si  $M$  no cubre a  $v$ .

**Definición 4.1** (Algoritmo GREEDY $_q$ ) *Dado un parámetro  $q \geq 1$ , el algoritmo GREEDY $_q$  para el problema OMP3-(1,1) con pesos en  $L$  es el siguiente.*

---

**Algoritmo 1:** GREEDY $_q$  con parámetro  $q \geq 1$ .

---

**Input:**  $L$  lado *offline*,  $w$  función de peso en  $L$   
**Output:** Un matching  $M$

- 1  $M(v) \leftarrow \perp, \forall v \in L \cup R;$
- 2  $\mu(v) \leftarrow 2, \forall v \in L \cup R$  (este número disminuirá cuando  $v$  sea emparejado o cuando cambie su pareja);
- 3 **forall**  $j \in R$ , cuando llega **do**
- 4     Definir (fijar  $F(j) = \perp$  o  $S(j) = \perp$  si no existen)
- 5      $F(j)$  : vecino  $i \in N(j)$ , libre ( $\mu(i) = 2$ ), de mayor peso.
- 6      $S(j)$  : vecino  $i \in N(j)$ , ocupado tal que el nodo  $k = M(i)$  tiene un vecino  $l$  libre ( $jikl$  es un camino  $M$ -aumentante) y además,  $\mu(i) = \mu(k) = 1$ , tal que  $w(i)$  es máximo dentro de los que cumple esta condición. Dentro de todas las posibilidades para  $l$  elegimos el de mayor peso.
- 7     **if**  $w(S(j)) \geq q \cdot w(F(j))$  **then**
- 8         fijar  $M(j) = S(j)$  (y actualizar  $\mu(j) = 0, \mu(i) = 0, \mu(k) = 0, \mu(l) = 0$  (se puede fijar todo a 0).
- 9     **else**
- 10         fijar  $M(j) = F(j)$  (y actualizar  $\mu(j) = 1, \mu(i) = 1$ )
- 11     **end**
- 12 **end**

---

**Proposición 4.2** *Eligiendo  $q = \sqrt{2}+1$ , el algoritmo anterior es  $\frac{1+2\sqrt{2}}{7} \approx 0,546918$ -competitivo.*

DEMOSTRACIÓN. Para aplicar el esquema primal-dual a este algoritmo, deberemos plantear la relajación LP del problema de emparejamiento de peso máximo con pesos en  $L$ , que llamaremos  $P_{G,w}$ .

$$\begin{aligned}
 (P_{G,w}) \quad & \max_{x_e, e \in E} \sum_{ij \in E} w(i)x_{ij} \\
 \text{s.a.} \quad & \forall v \in L \cup R \quad \sum_{u \in N(v)} x_{uv} \leq 1 \\
 & \forall e \in E \quad x_e \geq 0
 \end{aligned} \tag{4.1}$$

Su dual  $D_G$  es el siguiente.

$$\begin{aligned}
 (D_{G,w}) \quad & \min_{\alpha_i, \beta_j, i \in L, j \in R} \sum_{i \in L} \alpha_i + \sum_{j \in R} \beta_j \\
 \text{s.a.} \quad & \forall ij \in E \quad \alpha_i + \beta_j \geq w(i) \\
 & \forall i \in L, j \in R \quad \alpha_i, \beta_j \geq 0
 \end{aligned} \tag{4.2}$$

Al emparejar un nuevo vértice  $i \in L$  el valor primal aumenta en  $w(i)$  y por lo tanto, si queremos demostrar una tasa competitiva de  $F$ , tenemos permitido incrementar el valor dual en  $\frac{w(i)}{F}$ . Además ahora buscamos que para toda arista  $ij \in E$ ,  $\alpha_i + \beta_j \geq w(i)$ .

Sea ahora  $\lambda$  tal que  $\frac{1}{F} = 1 + 2\lambda$ . Supongamos  $\frac{2}{3} \leq 2\lambda \leq 1$ . La asignación dual es como sigue.

- Todas las variables se inicializan en 0.
- Si al llegar  $j$  se empareja directamente a  $i := F(j)$ , entonces  $\beta_j = w(i)$  y  $\alpha_i = 2\lambda w(i)$  como en la Figura 4.1



Figura 4.1:  $j$  llega y se empareja directamente con  $i$ .

- Si al llegar  $l$  se aumenta el matching con un camino  $ijkl$  (es decir  $k = S(l)$ ), se fijan  $\alpha_i = w(i)$ ,  $\alpha_k = w(k)$ ,  $\beta_j = 2\lambda w(i) + \lambda w(k)$  y  $\beta_l = \lambda w(k)$  como en la Figura 4.2.
- Al terminar, buscar todos los vértices  $j \in R$  que sigan emparejados a  $i = F(j)$ , y que por lo tanto  $\alpha_i = 2\lambda w(i)$  y  $\beta_j = w(i)$ . Si todos los vecinos de  $j$  fueron emparejados, entonces estos valores se invierten a  $\alpha_i = w(i)$  y  $\beta_j = 2\lambda w(i)$  como en la Figura 4.3.

Cada vez que un nuevo vértice de  $L$  es emparejado, ya sea directamente o con un camino de largo 3, el valor total del vector dual aumenta en  $w(i)(1 + 2\lambda)$ . Sea ahora  $ij \in E$  debemos verificar que al finalizar  $\alpha_i + \beta_j \geq w(i)$ . Lo haremos caso por caso.

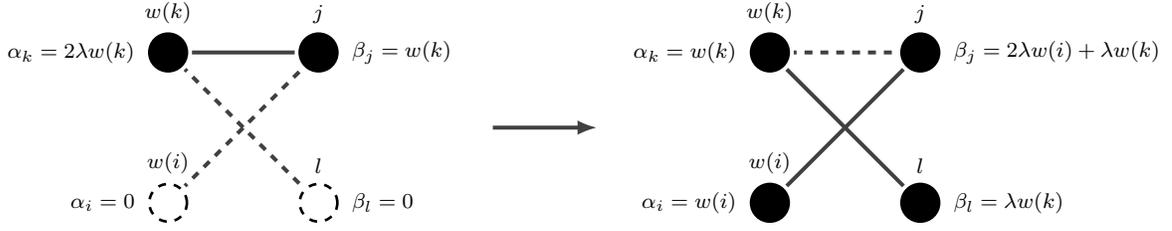


Figura 4.2:  $l$  llega y se empareja usando un camino aumentante  $ijk l$ .



Figura 4.3: Al terminar la ejecución, si  $j$  se emparejó directamente a  $i$ ,  $ij$  queda en el matching final y todos los vecinos de  $j$  fueron emparejados, se intercambian los valores de  $\alpha_i$  y  $\beta_j$ .

1.  $i$  no se empareja.  $j$  se empareja con asignación directa a  $F(j)$  y permanece así hasta el final. Como  $i$  estaba libre,  $w(F(j)) \geq w(i)$ . Además en este caso  $j$  terminó con un vecino libre ( $i$ ) por lo que **no** cambió su valor en el último paso, y luego  $\beta_j = w(F(j)) \geq w(i)$ .
2.  $i$  no se empareja.  $j$  se empareja con  $F(j)$  y después es re-asignado a otro vértice, digamos  $k \in L$ . Nuevamente  $i$  estaba libre por lo que  $w(F(j)) \geq w(i)$ , y además  $w(k) \geq w(i)$  pues  $j$  se reasignó a  $k$  en vez de  $i$ . Luego  $\beta_j \geq \lambda w(F(j)) + 2\lambda w(k) \geq 3\lambda w(i)$ .
3.  $i$  no se empareja.  $j$  se empareja a  $S(j)$  con un 3-camino. Como  $j$  eligió un 3-camino,  $w(S(j)) \geq q \cdot w(F(j)) \geq q \cdot w(i)$ . Luego  $\beta_j = \lambda w(S(j)) \geq q\lambda w(i)$ .
4.  $i$  se empareja y  $\alpha_i = w(i)$ . Inmediatamente  $\alpha_i + \beta_j \geq w(i)$ .
5.  $i$  se empareja y  $\alpha_i = 2\lambda w(i)$ . Es decir  $i$  fue elegido directamente por un vértice, digamos  $x \in R$ , y al terminar la ejecución la arista  $ix$  quedó dentro del matching y el vértice  $x$  quedó con algún vecino libre.
  - Si  $j$  llega antes que  $x$ ,  $i$  estaba libre y por lo tanto  $j$  se empareja a un vértice (ya sea  $F(j)$  o  $S(j)$ ) de peso al menos  $w(F(j)) \geq w(i)$ . En particular  $\beta_j \geq \lambda w(i)$  y  $\alpha_i + \beta_j \geq 3\lambda w(i)$ .
  - Si  $j$  llega después que  $x$ , entonces puede emparejarse con un 3-camino con la arista  $ix$  (pues  $x$  tiene un vecino libre). Luego  $w(S(j)) \geq w(i)$ . Si  $j$  se empareja con un 3-camino,  $\beta_j \geq \lambda w(i)$ . Si  $j$  se empareja directamente a  $F(j)$ , entonces  $\beta_j \geq \lambda w(F(j))$ , y como  $j$  eligió un emparejamiento directo por sobre el 3-camino disponible,  $w(S(j)) < q \cdot w(F(j))$ . Luego  $\beta_j \geq \frac{\lambda}{q} w(i)$  y  $\alpha_i + \beta_j \geq \lambda(2 + \frac{1}{q})w(i)$ .

Por lo tanto, si imponemos que  $3\lambda \geq 1$ ,  $q\lambda \geq 1$  y  $\lambda(2 + \frac{1}{q}) \geq 1$  podemos concluir que  $\text{GREEDY}_q$  es  $\frac{1}{1+2\lambda}$ -competitivo.

Optimizando sobre  $q$  y  $\lambda$ , obtenemos los valores  $\lambda = \sqrt{2} - 1$  y  $q = 1 + \sqrt{2}$ , y por lo tanto  $F = \frac{2\sqrt{2}+1}{7} = 0,546\dots$

□

Para finalizar, discutiremos brevemente la versión de OMP-3-(1,1) con pesos en **ambos** lados,  $L$  y  $R$ . En esta versión el peso de un vértice  $j \in R$  es desconocido hasta que  $j$  es recibido, y el objetivo es maximizar el peso total de todos los vértices emparejados.

El principal obstáculo al intentar diseñar un algoritmo para este problema es el hecho que los vértices que aún no han llegado pueden tener un peso mucho mayor a los que han sido recibidos hasta el momento. Consideremos por ejemplo la misma familia de instancias de la figura 3.1 donde asignamos los pesos tales que  $w(l_1) = w(l_2) = w(l_3) = 0$ ,  $w(r_1) = w(r_2) = 1$  y  $w(r_3) = M$  con  $M \geq 0$ .

Si nuestro algoritmo decidiera no emparejar alguno de los 2 primeros vértices  $r_1$  o  $r_2$  con tal de asegurar que  $r_3$  sea emparejado, podemos fijar  $M = 0$  con lo que solo se logra un matching de peso 1 del máximo de 2.

Si por el otro lado emparejamos  $r_1$  y  $r_2$ , ya vimos que en al menos una de las cuatro instancias  $r_3$  no puede ser emparejado por lo que en esa instancia solo logramos un matching de peso 2 del máximo de  $2 + M$ . Si fijamos  $M = 2$  vemos que no superamos una tasa de  $\frac{1}{2}$ .

Así vemos que ningún algoritmo determinista para está versión de OMP-3-(1, 1) con pesos en ambos lados puede superar una tasa de  $\frac{1}{2}$ , y en realidad este resultado es válido para las versiones con pesos en ambos lados de cualquier problema OMP-3-( $s, t$ ) con  $s, t \in \mathbb{N} \cup \infty$  pues la familia de 4 instancias de la figura 3.1 funciona en el caso  $s = t = \infty$ .

# Capítulo 5

## Conclusión

Los resultados generales del trabajo son los siguientes. Para cualquier valor de  $K = 2n - 1$ , cualquier algoritmo *greedy* y SAP para OMP- $K$  es óptimo (desde el análisis competitivo) entre los algoritmos deterministas, logrando una tasa de  $\frac{n}{n+1}$ . En el caso particular de  $K = 3$  basta tomar un algoritmo *greedy*.

Respecto a los problemas con caminos aumentantes de largo 3 y límites de cambios, un algoritmo *greedy* y SAP para OMP-3-(1, 1) es óptimo entre los deterministas y logra una tasa competitiva de  $\frac{3}{5}$ . En el caso con  $s = 1$  y  $t$  libre el mismo algoritmo alcanza una tasa de  $\frac{2t+1}{3t+2}$ . En particular, un algoritmo *greedy* y SAP para OMP-3-(1,  $\infty$ ) logra el mismo desempeño que un algoritmo *greedy* en OMP-3-( $\infty$ ,  $\infty$ ).

Finalmente, en la versión de OMP-3-( $s, t$ ) con pesos en el lado *offline*, el algoritmo GREEDY $_q$  con el valor  $q = \sqrt{2} + 1$  alcanza una tasa competitiva de  $\frac{1+2\sqrt{2}}{7} \approx 0,546918$ .

A continuación se resumen los resultados obtenidos, incluyendo el desempeño del mejor algoritmo determinista y la menor cota superior encontrados para cada problema.

Solo se encontraron algoritmos óptimos para los problemas OMP- $K$  con  $K \in \mathbb{N}$ , OMP-3-(1, 1) y OMP-3-(1,  $\infty$ ). Todos los demás problemas de la familia OMP- $K$ -( $s, t$ ) quedan con resultados por explorar. Se conjetura que de la misma forma en que es posible lograr el mismo desempeño en OMP-3-(1,  $\infty$ ) que en OMP-3-( $\infty$ ,  $\infty$ ), se debe poder lograr el mismo desempeño en OMP-3-(1,  $t$ ) que en OMP-3-( $s, t$ ). En otras palabras que, al menos para  $K = 3$ ,

Tabla 5.1: Resumen de resultados obtenidos para algoritmos deterministas

Problema	Desempeño alcanzado	Cota superior encontrada
OMP- $K$ , $K = 2n - 1$	$\frac{n}{n+1}$	$\frac{n}{n+1}$
OMP-3-(1, 1)	$\frac{3}{5}$	$\frac{3}{5}$
OMP-3-(1, $t$ )	$\frac{2t+1}{3t+2}$	$\frac{2t+1}{3t+2}$
OMP-3-(1, $\infty$ )	$\frac{2}{3}$	$\frac{2}{3}$
OMP-3-( $s, t$ )	$\frac{2t+1}{3t+2}$	$\frac{2}{3}$
OMP- $K$ -( $s, t$ )	?	$\frac{n}{n+1}$

valores de  $s$  mayores que 1 no permiten aumentar la tasa competitiva alcanzada. Queda también completamente abierto el estudio de algoritmos aleatorios para estos problemas.

# Bibliográfia

- [1] Claude Berge. TWO THEOREMS IN GRAPH THEORY. *Proceedings of the National Academy of Sciences*, 43(9), 1957.
- [2] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of RANKING for online bipartite matching. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2013.
- [3] John E. Hopcroft and Richard M. Karp. An  $n^{5/2}$  Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2(4), 1973.
- [4] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. Optimal algorithm for on-line bipartite matching. 1990.
- [5] Dénes König. Gráfok és mátrixok. *Matematikai és Fizikai Lapok*, 38:116–119, 1931.
- [6] Douglas B West. Introduction To Graph Theory: Solution Manual (to Second Edition 2001). *Read*, (2001), 2005.