



El Trimestre Económico

ISSN: 0041-3011

trimestreeditor@fondodeculturaeconomic
a.com

Fondo de Cultura Económica
México

Oviedo, Jorge; Rubio Duca, Ana
Algoritmo de aceptación diferida matricial
El Trimestre Económico, vol. LXXV, 2008, pp. 191-202
Fondo de Cultura Económica
Distrito Federal, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=31340953011>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

ALGORITMO DE ACEPTACIÓN DIFERIDA MATRICIAL*

*Jorge Oviedo y Ana Rubio Duca***

RESUMEN

En este artículo damos una versión matricial del algoritmo de aceptación diferida para el modelo de asignación (*matching*) uno a uno. El algoritmo va modificando la matriz de preferencia de los agentes. Cuando el algoritmo se detiene se muestra que coincide con una asignación estable óptima de los agentes.

ABSTRACT

In this paper we give matrix version of the deferred-acceptance algorithm for the matching model one to one. The new algorithm modifies the matrix of the agents preference. When the algorithm stops, it is shown that coincides with an agent optimal-stable matching.

INTRODUCCIÓN

En este artículo se da una versión matricial del algoritmo de aceptación diferida para el modelo de asignación (*matching*) uno a uno.

Este modelo se usa para estudiar problemas de asignación bilateral, es decir cuando los agentes se dividen en dos conjuntos disjuntos, el conjunto de

* *Palabras clave:* modelo de asignación uno a uno, asignación estable, algoritmo de aceptación diferida, matriz de preferencia. *Clasificación JEL:* C78, D71, D78. Este artículo fue financiado en parte por UNSL, Proico 319502. Conicet PIP 6293. FONCYT-SECYT, PAV 120/2003. FONCYT, PICT 03-10814.

** Instituto de Matemática Aplicada San Luis, Departamento de Matemática, Universidad Nacional de San Luis y Conicet (correos electrónicos: joviedo@unsl.edu.ar y rubio-duc@unsl.edu.ar).

empresas y el conjunto de trabajadores. El problema consiste en asignar a cada empresa un trabajador. Este problema de asignación se puede generalizar si se le permite a cada empresa contratar a un conjunto de trabajadores y a cada trabajador trabajar para varias empresas. Este modelo se lo conoce como modelo de asignación muchos a muchos; también hay un modelo intermedio de asignación, conocido como muchos a uno. La generalización de los resultados en general no son inmediatas.

Cada empresa tiene preferencias respecto a los potenciales trabajadores y simétricamente los trabajadores tienen preferencias respecto a las empresas para las cuales quisieran trabajar. Una solución de este problema de asignación es una asignación, que establece que compañero le toca a cada agente. Una asignación es estable si a cada agente le toca un compañero aceptable y si no existe un par de agentes empresa-trabajador que ambos prefieran asociarse mejorando en sus preferencias al compañero que le tocaba en la solución propuesta. Gale y Shapley (1962) mostraron que en el modelo de asignación uno a uno el conjunto de asignaciones estables es no vacío. Roth y Sotomayor (1990) y Gusfield e Irving (1986) son una importante referencia para estudiar los modelos de asignación. El conjunto de asignaciones estables está bien estructurado ya que contiene dos asignaciones: la óptima de las empresas (denotada por $_F$) y la óptima de los trabajadores (denotada por $_W$). Las asignaciones $_F$ y $_W$ son consideradas unánimemente por todas las empresas como la mejor y la peor entre todas las asignaciones estables; simétricamente $_W$ y $_F$ son consideradas por los trabajadores como la mejor y la peor de las asignaciones estables. Blair (1988) mostró que el conjunto de asignaciones estables tiene estructura de *latices*.

Los algoritmos han desempeñado un papel central en la bibliografía de la asignación. El algoritmo NIMP (Programa de médicos internos nacionales) de 1951, y posteriores modificaciones NRMP (Programa de médicos residentes nacionales) de 1979, 1987 1999 se usan para asignar a hospitales médicos residentes en los Estados Unidos. El algoritmo de aceptación diferida de Gale y Shapley se usa en otros programas de asignación de distintas especialidades de la salud a hospitales. Además del algoritmo de aceptación diferida de Gale y Shapley, Adachi (2000) usó técnicas de punto fijo para construir asignaciones estables. Fleiner (2001) y Echenique y Oviedo (2004) extendieron de punto fijo al modelo de asignación muchos a uno para construir asignaciones estables.

McVitie y Wilson (1971) e Irving y Leather (1986) definieron un algorit-

mo para computar el conjunto de todas las asignaciones estables en el modelo de asignación uno a uno. Martínez, Massó, Neme y Oviedo (2004) extendieron el algoritmo al modelo de asignación muchos a muchos.

El algoritmo que presentamos aquí es una versión del algoritmo de aceptación diferida en forma matricial. Para esto construimos dos matrices, M^F y M^W . La primera contiene en cada fila la información del orden de preferencias de las empresas, asignándole un 1 al trabajador más preferido, 2 al siguiente en su orden de preferencia y así continúa; la segunda matriz, contiene en forma simétrica en cada columna la información del orden de preferencias de los trabajadores, asigna un 1 a la empresa más preferida, 2 a la que sigue y así continúa. El algoritmo va modificando la matriz M^F : primero ubica todos los 1, es decir, cada empresa ubica al trabajador más preferido; si en alguna columna hay más de un 1, ese trabajador calcula, usando M^W , el mínimo, o sea, el trabajador elige la oferta más preferida. Redefinimos la matriz M^F asignándole a la fila (empresa) donde hay un 1 y columna (trabajador) que no es mínimo, es decir, para el trabajador esa empresa no es la más preferida. En los otros elementos de la fila le restamos 1, es decir, el trabajador que era segundo, en el orden de preferencia original, ahora tendrá un 1, es decir, es el trabajador más preferido; el tercero en el orden de preferencias original ahora será el 2, y así continúa. Se ubican nuevamente los 1 de la matriz modificada M^F y se repite el proceso anterior. El algoritmo se detiene cuando no podemos modificar más M^F . Ubicando los 1 de M^F tenemos qué trabajador (si es que hay uno) le toca a cada empresa. Mostramos que esta asignación coincide con la óptima $_F$ de las empresas.

Este artículo está organizado como sigue. En la sección I damos las definiciones y notaciones preliminares. La sección II contiene el algoritmo y el resultado principal que muestra que cuando el algoritmo *se detiene*, los 1 de M^F nos dan la asignación estable óptima $_F$ de las empresas.

I. DEFINICIONES BÁSICAS

Hay dos conjuntos finitos y disjuntos de agentes F y W , $F = \{f_1, f_2, \dots, f_n\}$ es el conjunto de empresas y $W = \{w_1, w_2, \dots, w_p\}$ es el conjunto de trabajadores. Cada agente $g \in F \cup W$ tiene una relación de preferencias $P(g)$ estricta, transitiva y completa respecto a los agentes del otro conjunto unido a él; por ejemplo:

$$w_1 P(f) w_3 P(f) w_5 P(f) f P(f) w_4 P(f) w_2$$

dice que la empresa f tiene a w_1 como el trabajador más preferido, luego sigue en orden w_3, w_5, f, w_4 y w_2 . Los trabajadores w son tales que $wP(f)f$ se dice que son aceptables. $fP(f)w_4$ significa que este trabajador no es aceptable para la empresa f . Tenemos que los trabajadores w_1, w_3 y w_5 , son aceptables para la empresa f , mientras que w_4 y w_2 son inaceptables para f . Para expresar el orden anterior de manera concisa, debido a que los agentes no están obligados a contratar o ser contratados por un agente no aceptable, denotaremos en el orden de preferencias sólo a los agentes aceptables, es decir, el orden anterior se denota por:

$$P(f) \quad w_1, w_3, w_5$$

Un perfil de preferencias es de $(n \quad p)$ -*upla* de relaciones de preferencias y lo representamos por $P = (P(f_1), \dots, P(f_n), P(w_1), \dots, P(w_p))$. R denota el orden débil asociado a P .

Un modelo de asignación uno a uno se denota por (F, W, P) .

Definición 1. Una asignación es una aplicación uno a uno de $F \rightarrow W$ en $F \rightarrow W$, en que para todo $f \in F$ y $w \in W$ se cumple que:

- i) $(f) \rightarrow f$ implica $(f) \rightarrow W$,
- ii) $(w) \rightarrow w$ implica $(w) \rightarrow F$,
- iii) $w \rightarrow (f)$ si y sólo si $f \rightarrow (w)$.

Una asignación representada por

$$\begin{array}{ccccc} f_1 & f_2 & f_3 & f_4 & w_5 \\ w_2 & w_4 & w_1 & w_3 & w_5 \end{array}, \text{ o por } \begin{array}{ccccc} w_1 & w_2 & w_3 & w_4 & w_5 \\ f_3 & f_1 & f_4 & f_2 & w_5 \end{array}$$

en este caso $(f_1) \rightarrow w_2, (f_2) \rightarrow w_4, (f_3) \rightarrow w_1, (f_4) \rightarrow w_3$ y $(w_5) \rightarrow w_5$.

La asignación es individualmente racional si para todo agente $g \in F \cup W$

$$(g)P(g)g$$

es decir que la asignación le asigna a cada agente un compañero aceptable.

El par de agentes (f, w) bloquea a la asignación si $(f) \rightarrow w$ y

$$wP(f)(f) \text{ y } fP(w)(w)$$

esto muestra que tanto f como w quieren romper la asignación para formar una nueva asignación emparejándose entre ellos.

Las asignaciones individualmente racionales que sobrevivan a blo-

queos por pares de agentes, desempeñan un papel fundamental en los modelos de asignación, ya que no podrán modificarse, porque en caso de hacerlo algún o algunos de los agentes intervinientes quedarán con un compañero menos preferido que los que le asignaba la asignación original, formalmente:

Definición 2. Una asignación es estable si es individualmente racional y no es bloqueado por ningún par de agentes.

Sea P un perfil de preferencias, y $S(P)$ denota el conjunto de asignaciones estables.

El ejemplo siguiente muestra las definiciones dadas hasta aquí:

Ejemplo 1. Sea el siguiente modelo de asignación (F, W, P) , en que $F = \{f_1, f_2, f_3, f_4\}$, $W = \{w_1, w_2, w_3\}$,

$$\begin{array}{ll} P(f_1) & w_2, w_1, w_3 \\ P(f_2) & w_1, w_3, w_2 \\ P(f_3) & w_1, w_2, w_3 \\ P(f_4) & w_2, w_3, w_1 \end{array} \quad \begin{array}{ll} P(w_1) & f_1, f_3, f_2 \\ P(w_2) & f_3, f_1, f_2 \\ P(w_3) & f_1, f_3, f_2 \end{array}$$

Consideremos las asignaciones siguientes:

$$^1 \begin{array}{cccc} w_1 & w_2 & w_3 & f_1 \\ f_4 & f_2 & f_3 & f_1 \end{array}, \quad ^2 \begin{array}{cccc} w_1 & w_2 & w_3 & f_4 \\ f_1 & f_2 & f_3 & f_4 \end{array}, \quad ^3 \begin{array}{cccc} w_1 & w_2 & w_3 & f_4 \\ f_3 & f_1 & f_2 & f_4 \end{array}$$

La asignación 1 no es individualmente racional, porque $w_1 P(w_1) f_4$ $^1(w_1)$. 2 es individualmente racional pero es inestable ya que (f_1, w_2) es un par bloqueante. 3 es una asignación estable.

El orden de preferencia $P(g)$ de un agente, se puede extender para que pueda comparar las asignaciones. Sea $g \in F \cup W$, decimos que

$$P(g) \leq h \text{ si } (g)P(g) \leq (g)h$$

Decimos que

$$P(F) \leq h \text{ si } \begin{array}{l} (f)R(f) \leq (f)h \text{ para cada } f \in F \\ (f)P(f) \leq (f)h \text{ para algún } f \in F \end{array}$$

y

$$R(F) \leq h \text{ si } \begin{array}{l} = \\ \text{o} \\ P(F) \end{array}$$

De manera análoga se define $P(W)$ y $R(W)$. En el ejemplo anterior tenemos que $^3 P(F) \leq ^2$.

Gale y Shapley (1962) mostraron que el conjunto de asignaciones esta-

bles es distinto del vacío. También mostraron que existe la asignación estable (F, W) que es la mejor a las asignaciones estables para las empresas (trabajadores). Además el óptimo de las asignaciones estables para una parte del mercado es el peor para otra parte, es decir que cada $S(P)$ se cumple que

$${}_F R(F) \cap R(F) \cap W \cap Y \cap W R(W) \cap R(W) \cap F$$

El algoritmo de aceptación diferida definido por Gale y Shapley (1962) construye (F, W) , dependiendo quien hace las ofertas. En cualquier paso del algoritmo, en el cual las empresas hacen las ofertas, una empresa se propone al mejor trabajador de su orden de preferencia que no la ha rechazado aún, mientras que un trabajador acepta, provisionalmente, a la mejor empresa aceptable que le hizo una oferta (si la hubiera) unido con la empresa provisionalmente aceptada en un paso anterior. El algoritmo se detiene en cualquier paso en el cual todas las ofertas son aceptadas; la asignación provisional que se forma con las propuestas aceptadas se torna definitiva. El resultado del algoritmo es la asignación estable (F, W) . Simétricamente si los trabajadores hacen las ofertas el resultado del algoritmo es la asignación estable (F, W) .

Ejemplo 1 (continuación). Tenemos que $(F, W) = (3, Y)$

$$\begin{array}{ccccc} & w_1 & w_2 & w_3 & f_4 \\ W & f_1 & f_3 & f_2 & f_4 \end{array}$$

II. ALGORITMO DE ACEPTACIÓN DIFERIDA MATRICIAL

Este algoritmo usa notación matricial. Las filas de la matriz representan a las empresas y las columnas a los trabajadores, así que fijaremos un orden para los agentes. Cuando no haya problemas denotamos con i la fila de la matriz que representa a la empresa f_i ; simétricamente la columna j representa al trabajador w_j . Dado el modelo de asignación (F, W, P) , definimos las siguientes matrices de preferencias M^F y M^W , sean $f_i \in F$ y $w_j \in W$

$$m_{ij}^F: \begin{array}{cc} k & 1 \\ w_j P(f_i) f_i & y \quad |\{w_j : w_j P(f_i) w_j\}| \\ f_i P(f_i) w_j & k \end{array}$$

y

$$m_{ij}^W: \begin{array}{cc} k & 1 \\ f_i P(w_j) w_j & y \quad |\{f_i : f_i P(w_j) f_i\}| \\ w_j P(w_j) f_i & k \end{array}$$

Tenemos que el trabajador w_j es aceptable para la empresa f_i , si m_{ij}^F y que la empresa f_i es aceptable para el trabajador w_j , cuando m_{ij}^W . La empresa f_i y el trabajador w_j son mutuamente aceptables cuando m_{ij}^F y m_{ij}^W .

Ejemplo 2. Construyamos de las matrices de preferencias M^F y M^W para $F = \{f_1, f_2, f_3, f_4, f_5\}$, $W = \{w_1, w_2, w_3, w_4\}$ y los ordenes de preferencia

$$\begin{array}{ll} P(f_1) & w_1, w_2, w_3, w_4 \\ P(f_2) & w_4, w_2, w_3, w_1 \\ P(f_3) & w_4, w_3, w_1, w_2 \\ P(f_4) & w_1, w_4, w_3, w_2 \\ P(f_5) & w_1, w_2, w_4 \end{array} \quad \begin{array}{ll} P(w_1) & f_2, f_3, f_1, f_4, f_5 \\ P(w_2) & f_3, f_1, f_2, f_4, f_5 \\ P(w_3) & f_5, f_4, f_1, f_2, f_3 \\ P(w_4) & f_1, f_4, f_5, f_2, f_3 \end{array}$$

Como $w_1 P(f_1) f_1$ y $|\{w_j : w_j P(f_1) w_1\}| = |\emptyset| = 0 < k$, entonces $m_{11}^F = 1$, similarmente, $m_{12}^F = 2$ pues $w_2 R(f_1) f_1$ y $|\{w_j : w_j P(f_1) w_2\}| = |\{w_1\}| = k - 1$, y $m_{13}^F = 3$ pues $w_3 R(f_1) f_1$ y $|\{w_j : w_j P(f_1) w_3\}| = |\{w_1, w_2\}| = k - 2$. Notemos que $m_{53}^F = 3$ pues $f_5 P(f_3) w_3$. Obtenemos las siguientes matrices de preferencias

$$M^F = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 4 & 2 & 3 & 1 \\ 3 & 4 & 2 & 1 \\ 1 & 4 & 3 & 2 \\ 1 & 2 & & 3 \end{array} \quad M^W = \begin{array}{cccc} 3 & 2 & 3 & 1 \\ 1 & 3 & 4 & 4 \\ 2 & 1 & 5 & 5 \\ 4 & 4 & 2 & 2 \\ 5 & 5 & 1 & 3 \end{array}$$

Algoritmo. Entrada: $M^{F,0}$, M^F , y M^W matrices de preferencias de los agentes. Salida: Matriz de a lo más un uno por columna.

i) Si para cada j , $|\{m_{ij}^{F,t} : m_{ij}^{F,t} = 1\}| = 1$, paro // se ha determinado la matriz.

ii) Si no, defino¹

$$m_{ij}^{F,t+1} = \begin{cases} j & \text{si } j \text{ y } m_{ij}^W = \min\{m_{lj}^W : m_{lj}^{F,t} = 1\} \\ m_{ij}^{F,t} & \text{si } j \text{ y } m_{ij}^W = \min\{m_{lj}^W : m_{lj}^{F,t} = 1\} \\ m_{ij}^{F,t} & \text{en otro caso} \end{cases}$$

Defino $t = t + 1$. Regresa a 1.

El algoritmo trabaja de la siguiente manera. Dada la matriz de preferencias de las empresas M^F , cuenta cuantos 1 tiene en cada columna, si

¹ $\min\{m_{lj}^W : m_{lj}^{F,t} = 1\}$: , si no existe l, j tal que $m_{lj}^{F,t} = 1$

para toda trabajador w_j hay a lo más un 1, el algoritmo se detiene, pues se encontró la matriz de salida. Si esto no ocurre se construye una nueva matriz de preferencias para las empresas, de la siguiente manera: sea el trabajador w_j , tal que existe una única empresa f_i , con $m_{ij}^{F,t} = 1$, entonces la f_i de $M^{F,t-1}$ no se modifica, es decir, $M_{i,j}^{F,t-1} = M_{i,j}^{F,t}$. Si para el trabajador w_j existen al menos dos empresas f_i y f_i , tales que $m_{ij}^{F,t} = m_{ij}^{F,t-1}$, sin pérdida de generalidad supongamos que $m_{ij}^W = \min\{m_{lj}^W : m_{lj}^{F,t} = 1\}$, entonces en este caso la fila i de $M^{F,t-1}$ no se modifica, es decir, $M_{i,j}^{F,t-1} = M_{i,j}^{F,t}$, mientras que la fila i se modificará por $M_{i,j}^{F,t-1} - m_{ij}^{F,t} + m_{ij}^W$ (notemos que para cada $t = t-1$, $m_{ij}^{F,t} = m_{ij}^{F,t-1}$) y $m_{ij}^{F,t-1} = m_{ij}^{F,t} - 1$ para todo $w_j = w_j$. Luego hace $t = t-1$, y vuelve a 1, a contar cuantos 1 tiene por columna.

Si el algoritmo no para es porque existe w_j , tal que $|\{m_{ij}^{F,t-1} : m_{ij}^{F,t-1} = 1\}| = 1$, entonces pasamos a 2, en el que se modificarán ciertas filas en donde la columna w_j sea tal que $m_{ij}^W = \min\{m_{lj}^W : m_{lj}^{F,t} = 1\}$. Note que en modificación cambia un 1 por un 0. Por finitud de las entradas el algoritmo debe parar, es decir existe t tal para cada j se cumple que:

$$|\{m_{ij}^{F,\bar{t}} : m_{ij}^{F,\bar{t}} = 1\}| = 1$$

Supongamos que el algoritmo se detiene en t , entonces definimos una aplicación $\bar{\cdot}$ de $F \times W$ en $f \times w$ por:

$$\begin{aligned} \bar{\cdot}(f_i) &= w_j \text{ si } m_{ij}^{F,\bar{t}} = 1 \text{ y } m_{ij}^W \\ &= f_i \text{ si } i \neq j : m_{ij}^{F,\bar{t}} \\ \bar{\cdot}(w_j) &= f_i \text{ si } m_{ij}^{F,\bar{t}} = 1 \text{ y } m_{ij}^W \\ &= w_j \text{ si } i \neq j : m_{ij}^{F,\bar{t}} \end{aligned}$$

Notemos que $\bar{\cdot}(f_i) = f_i$ es cuando en la fila i de $M^{F,\bar{t}}$ todos los elementos son 0, pues si hubiese un elemento menor que 1 debe existir un trabajador w_j tal que $m_{ij}^{F,\bar{t}} = 1$, ya que por construcción de $M^{F,t-1}$ el algoritmo le resta 1 en cada posición de la fila o se cambia un 1 por 0.

Por otra parte, $\bar{\cdot}(w_j) = w_j$, cuando la columna j de $M^{F,\bar{t}}$, no posee ningún 1, es decir el trabajador w_j no recibió ninguna propuesta de las empresas aceptables para él.

El ejemplo siguiente muestra el algoritmo para el ejemplo 2.

Ejemplo 2 (continuación). Datos $M^{F,0} = M^F$ y M^W .

$$M^{F,0} = \begin{matrix} & 1 & 2 & 3 & 4 \\ \begin{matrix} 1 \\ 4 \\ 3 \\ 1 \\ 1 \end{matrix} & 2 & 3 & 4 \\ & 1 & 2 & 3 \end{matrix} \quad M^W = \begin{matrix} & 3 & 2 & 3 & 1 \\ \begin{matrix} 1 \\ 2 \\ 4 \\ 5 \end{matrix} & 3 & 4 & 4 \\ & 1 & 5 & 5 \\ & 4 & 2 & 2 \\ & 5 & 1 & 3 \end{matrix}$$

i) El algoritmo no para ya que en la columna 1 y 4 hay más de un 1, y ii) dada $M^{F,0}$ y M^W , construimos $M^{F,1}$

$$M^{F,1} = \begin{matrix} & 1 & 2 & 3 & 4 \\ \begin{matrix} 4 \\ 2 \end{matrix} & 2 & 3 & 1 \\ & 3 & 2 & 1 \\ & 1 & & 3 \end{matrix}$$

notemos que las filas 3, 4, y 5 se modifican ya que

$$\min\{m_{24}^W, m_{34}^W\} = \min\{4, 5\} = 4 = m_{24}^W \quad (1)$$

luego $m_{34}^{F,1}$, y para $j = 4$, $m_{34}^{F,1} = m_{34}^{F,0} - 1$. También (1) implica que para todo $j = 1, 2, 3, 4$, $m_{2,j}^{F,1} = m_{2,j}^{F,0}$. Como

$$\min\{m_{11}^W, m_{41}^W, m_{51}^W\} = \min\{3, 4, 5\} = 3 = m_{11}^W$$

la fila 1 de la matriz $M^{F,1}$, no se modifica mientras que si lo hacen las filas 4 y 5.

Hacemos $t = 1$: i) el algoritmo no para ya que en la columna 4 hay dos 1, y ii) dada $M^{F,1}$ y M^W , construimos $M^{F,2}$

$$M^{F,2} = \begin{matrix} & 1 & 2 & 3 & 4 \\ \begin{matrix} 3 \\ 2 \end{matrix} & 1 & 2 \\ & 3 & 1 \\ & 3 & 2 & 1 \\ & 1 & & 2 \end{matrix}$$

nótese que se modifica la fila 2 ya que

$$\min\{m_{i4}^W, m_{i4}^{F,2} - 1\} = \min\{m_{24}^W, m_{44}^W\} = m_{44}^W$$

Hacemos $t = 2$: i) el algoritmo no para ya que la columna 2 tiene dos 1, y ii) dada $M^{F,2}$ y M^W , construimos $M^{F,3}$

$$M^{F,3} = \begin{matrix} & 1 & 2 & 3 & 4 \\ \begin{matrix} 3 \\ 2 \end{matrix} & 1 & 2 \\ & 3 & 1 \\ & 3 & 2 & 1 \\ & & & 1 \end{matrix}$$

Hacemos $t = 3$: i) el algoritmo no para ya que la columna 4 tiene dos 1, y ii) dada $M^{F,3}$ y M^W , construimos $M^{F,4}$

$$M^{F,4} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 3 \\ 2 \\ 3 \end{matrix} \end{matrix} \begin{matrix} 2 \\ 1 \\ 2 \end{matrix} \begin{matrix} 4 \\ 1 \\ 1 \end{matrix}$$

Hacemos $t = 4$: el algoritmo se detiene, ya en cada columna hay a lo más un 1. Tenemos que $t = 4$.

La asignación que se obtiene a partir de $M^{F,4}$, es:

$$\begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ & w_1 & w_2 & w_3 & w_4 & f_5 \end{matrix}$$

Presentamos ahora el resultado principal

Teorema 1. $\bar{\cdot}$ es una asignación estable y $\bar{\cdot} \in F$.

Demostración: Supongamos que el algoritmo para en la t . Mostremos que $\bar{\cdot}$ es una asignación. De la definición de $\bar{\cdot}$, se cumple que si $\bar{\cdot}(f) = f$, entonces $\bar{\cdot}(f) \in W$ y $\bar{\cdot}(w) = w$, entonces $\bar{\cdot}(w) \in F$. También se cumple que si $m_{ij}^{F,\bar{t}} = 1$, implica que $\bar{\cdot}(f_i) = w_j$ y $\bar{\cdot}(w_j) = f_i$, por tanto es una asignación.

Veamos que $\bar{\cdot}$ es individualmente racional. Supongamos que $\bar{\cdot}(f_i) = w_j$, entonces $m_{ij}^{F,\bar{t}} = 1$ y $m_{ij}^W = 0$, por tanto por la definición de M^W tenemos que $f_i P(w_j) w_j$, el algoritmo implica que si $m_{ij}^{F,\bar{t}} = 1$, entonces $m_{ij}^{F,0} = m_{ij}^F$ y por la definición de M^F , $w_j P(f_i) f_i$.

Mostraremos que $\bar{\cdot}$ no es bloqueado por ningún par de agentes. Supongamos que existe un par de agentes $(f_i, w_j) \in F \times W$ tal que $\bar{\cdot}(f_i) = w_j$ y

$$w_j P(f_i) \bar{\cdot}(f_i) = w_j \quad (2)$$

es decir que $m_{ij}^F = m_{ij}^F$, y como $m_{ij}^{F,\bar{t}} = 1$, entonces $m_{ij}^{F,\bar{t}} = 1$, por tanto existe \hat{t} tal que $m_{ij}^{F,\hat{t}} = 1$, por el algoritmo tenemos que

$$m_{ij}^W = \min \{m_{ij}^W : m_{ij}^{F,t} = 1, \text{ para } t = 1, \dots, t\} = m_{ij}^W$$

entonces $m_{ij}^W = m_{ij}^W$ (nótese que $\bar{\cdot}(w_j) = f_i$), por tanto $f_i P(w_j) f_i$, es decir que (f_i, w_j) no puede bloquear a $\bar{\cdot}$.

Veamos que $\bar{\cdot} \in F$. Como $\bar{\cdot} \in S(P)$ tenemos que $\bar{\cdot} \in R(F) \cap R(F) \cap W$. Supongamos que $\bar{\cdot} \notin F$, sin pérdida de generalidad sean $f_1, \dots, f_{\bar{i}}$, tales que $\bar{\cdot}(f_i) = \bar{\cdot}(f_i)$. Podemos suponer que existen $w_1, \dots, w_{\bar{i}}$ tales que

$$\begin{matrix} w_1 & \bar{\cdot}(f_1) & P(f_1) & \bar{\cdot}(f_1) & w_2 \\ w_2 & \bar{\cdot}(f_2) & P(f_2) & \bar{\cdot}(f_2) & w_3 \\ & \vdots & \vdots & \vdots & \vdots \\ w_{\bar{i}} & \bar{\cdot}(f_{\bar{i}}) & P(f_{\bar{i}}) & \bar{\cdot}(f_{\bar{i}}) & w_1 \end{matrix}$$

Para cada $i = 1, \dots, \bar{i}$, sean t_i, l_i tales que

$$\begin{aligned} m_{ii}^{F, t_i} &= 1 \text{ y } m_{ii}^{F, t_i - 1} \\ m_{l_i i}^{F, t_i} &= 1 \text{ y } m_{l_i i}^{F, t_i - 1} = 1 \end{aligned} \quad (3)$$

Tanto t_i como l_i existen, ya que al aplicar el algoritmo se cumple que $m_{i, i-1}^{F, \bar{i}} = 1, m_{ii}^F = m_{i, i-1}^F$, por tanto en algún paso t_i existe l_i tal que las condiciones (3) son verdaderas. Supongamos que

$$t_1 = \min\{t_i : m_{ii}^{F, t_i} = 1, m_{ii}^{F, t_i - 1} = 0, m_{l_i i}^{F, t_i} = 1 \text{ y } m_{l_i i}^{F, t_i - 1} = 0\}$$

entonces

$$\begin{aligned} m_{11}^{F, t_1} &= 1 \text{ y } m_{11}^{F, t_1 - 1} \\ m_{l_1 1}^{F, t_1} &= 1 \text{ y } m_{l_1 1}^{F, t_1 - 1} = 1 \end{aligned} \quad (4)$$

notemos que

$$m_{l_1 1}^W = m_{11}^W \text{ o } f_{l_1} P(w_1) f_1 \quad (5)$$

si ocurre que $w_1 P(f_{l_1}) w_{l_1}$, entonces (f_{l_1}, w_1) bloquearían a f_1 por tanto se cumple que

$$w_{l_1} P(f_{l_1}) w_1 \quad (6)$$

Supongamos que $f_1 P(f_{l_1}) \neg(f_{l_1}), m_{l_1 1}^{F, t_1} = 1$, entonces $w_1 P(f_{l_1}) \neg(f_{l_1}) f_1 P(f_{l_1})$ y la condición (5) implica que (f_{l_1}, w_1) bloquean a f_1 . Luego, $f_1 P(f_{l_1}) \neg(f_{l_1})$, es decir que $f_1 P(f_{l_1}) w_{l_1}$. Entonces tenemos que t_{l_1} satisface las condiciones (3), además $t_{l_1} = t_1$, ya que por (6) tenemos que $m_{l_1 l_1}^F = m_{l_1 1}^F$. Esto contradice la elección minimal de t_1 . ■

REFERENCIAS BIBLIOGRÁFICAS

- Adachi, H. (2000), "On a Characterization of Stable Matchings", *Economic Letters*, 68, pp. 43-49.
- Echenique, F., y J. Oviedo (2004), "Core Many-to-one Matchings by Fixed Point Methods", *Journal of Economic Theory* 115, pp. 358-376.
- Fleiner, T. (2003), "A Fixed-Point Approach to Stable Matchings and Some Applications", *Mathematics of Operations Research*, 28, pp. 103-126.

- Gale, D., y L. Shapley (1962), "College Admissions and the Stability of Marriage", *American Mathematical Monthly* 69, pp. 9-15.
- Gusfield, D., y R. Irving (1989), *The Stable Marriage Problem: Structure and Algorithms*, Cambridge, MIT Press.
- Irving, R., y P. Leather (1986), "The Complexity of Counting Stable Marriages", *SIAM Journal of Computing* 15, pp. 655-667.
- Martínez, R., J. Massó, A. Neme y J. Oviedo (2004), "An Algorithm to Compute the Full Set of Many-to-Many Stable Matchings", *Mathematical Social Sciences*, 47, pp. 187-210.
- McVitie, D., y L. Wilson (1971), "The Stable Marriage Problem", *Communications of the ACM* 14, pp. 486-493.
- Roth, A., y M. Sotomayor (1990), *Two-sided Matching: A Study in Game-Theoretic Modeling and Analysis*, Cambridge, Cambridge University Press [Econometrica Society Monographs núm. 18].