



EL COLEGIO DE MÉXICO CENTRO DE ESTUDIOS ECONÓMICOS

MAESTRÍA EN ECONOMÍA

TRABAJO DE INVESTIGACIÓN PARA OBTENER EL GRADO DE MAESTRO EN
ECONOMÍA

**MERCADOS DE MATCHING:
EL CASO PARTICULAR DE LAS PAREJAS.
EN BUSCA DE UN NUEVO ALGORITMO**

FRANCISCO J. RODRÍGUEZ CARRASCO

PROMOCIÓN 2006-2008

ASESOR: DAVID RENÉ MICHEL CANTALA

ABRIL 2009



Agradecimientos.

Agradezco a Dios, a mis padres que siempre me han apoyado, a todos mis más que compañeros amigos de la maestría que de alguna u otra forma contribuyeron a la realización de esta tesis. Y un muy merecido agradecimiento a mi asesor el Dr. David Cantala sin cuyo apoyo la realización de esta tesis hubiera resultado imposible.

Resumen:

El mercado de estudiantes de medicina y hospitales provee un campo propicio para poner en practica algoritmos capaces de encontrar los matchings (emparejamientos) estables de dicho mercado. En este sentido se ha puesto en práctica un algoritmo que parecía funcionar adecuadamente en el mercado, sin embargo, trabajos recientes como el de Klaus Klijn y Masso (2003) muestran que en presencia de parejas del lado de los estudiantes no el algoritmo falla al no encontrar el matching estable.

Una vez descubierto el problema se abocan a la tarea de encontrar condiciones (sobre la estructura de las preferencias) que garantizan el adecuado funcionamiento del algoritmo. Si bien esto es un avance las condiciones dadas parecen restrictivas y no capturan adecuadamente las preferencias de las parejas por cercanía, por lo que en el trabajo de Cantala (2004) se sugiere un cambio sobre la estructura de las preferencias que capture mejor este deseo de las parejas por emparejarse en hospitales cercanos. Sin embargo, bajo estas nuevas preferencias el algoritmo actualmente empleado es ineficiente.

El presente es un trabajo pretende ya no dar condiciones bajo las cuales el algoritmo pueda funcionar, si no mas bien, se encuentra un algoritmo que resuelve el problema de matching en este mercado sin importar la estructura de las preferencias y sin importar la presencia o ausencia de parejas.

El algoritmo que se sugiere pertenece a la familia de algoritmos “Branch and Bound”, los cuales comúnmente son usados en problemas de maximización donde las variables son discretas o tenemos una mezcla de variables discretas y continuas. El algoritmo ha sido construido de tal forma que se puede garantizar que siempre encontrara los matchings estables siempre y cuando estos existan.

Índice

1. Introducción	5
2. El modelo	7
3. El problema	9
3.1 Existencia	9
3.2 Algoritmo de Aceptación Diferida (AD) vs. Algoritmo de Ramificación Con Parejas (RP)	10
4. El Algoritmo de Ramificación con Parejas (RP)	11
5. Prueba del teorema 3	19
6. Conclusiones	22
7. Bibliografía	24

1. Introducción

Cada año cerca de 20 mil estudiantes de las escuelas de medicina de los Estados Unidos se incorporan al mercado laboral a través de un mecanismo centralizado de matching, el Nacional Resident Matching Program (NRMP, a partir de aquí). El programa nace en la década de los 50's ante las constantes fallas de procesos descentralizados de organizar el mercado laboral entre graduados de escuelas de medicina y hospitales. Roth (1984) explica como este mecanismo en un principio utilizaba una variante del mecanismo de Aceptación Diferida desarrollado por Gale y Shapley (1962). Sin embargo, en los años 70's los estudiantes empezaron hacer uso del NRMP ya no de forma individual sino que ahora eran parejas de estudiantes las que buscaban que el programa las asignara en hospitales de una misma área geográfica. Es incuestionable que la incorporación de parejas al mercado laboral ha causado que el problema de matching se haga más complejo, ahora no solo hay que buscar una solución que sea aceptable para ambos lados del mercado, sino que, tenemos que lidiar con decisiones grupales en el lado de la demanda. Dado que el NRMP no estaba preparado para esta eventualidad los estudiantes comenzaron a utilizar cada vez menos el NRMP y comenzaron a negociar directamente con los hospitales, lo cual provoco que el mercado volviera a caer en la desorganización. Ante esta situación en 1998 el NRMP adopto un nuevo algoritmo que tenía como intención incorporar las preferencias de las parejas de manera mas adecuada. Y en un principio parecía que este nuevo algoritmo cumplía bien con su cometido, sin embargo trabajos recientes, por ejemplo el realizado por Klaus Klijn y Massó (2003), han probado que tal vez este nuevo algoritmo no este cumpliendo de manera adecuada para lo que fue diseñado.

Del trabajo realizado por Roth (1984) se sabe que en presencia de parejas no podemos garantizar que, en general, el conjunto de matchings estables no sea vacío. Klaus y Klijn (2003) dan condiciones bajo las cuales la existencia de un matching estable esta garantizada. La condición que presentan en su trabajo es referente a la estructura de las preferencias de las parejas, que nos dice que si las preferencias de las parejas son "responsive" (se refieren a un tipo de preferencias en el cual la mejoría unilateral de un miembro de la pareja mejora a la pareja) la existencia de un matching estable esta garantizada. En otro trabajo Klaus, Klijn y Massó (2003) muestran que a pesar de que el matching estable existe, el nuevo algoritmo de 1998 del NRMP es incapaz de encontrarlo mientras que el algoritmo de aceptación diferida si lo logra. Sin embargo, su

ejemplo esta construido sobre preferencias responsive que no parecen capturar bien la idea de proximidad geográfica, que es en última instancia lo que parece preocupar más a las parejas. En el trabajo de Cantala (2004) se sugiere el uso de preferencias lexicograficas, las cuales capturan la idea de que las parejas tienen preferencias sobre las regiones en las cuales se encuentran los hospitales. En contraparte, bajo preferencias lexicograficas volvemos al caso en donde la existencia de un matching estable ya no esta garantizado.

Nuestro trabajo es en cierto sentido más amplio, ya que no definimos condiciones bajo las cuales se garantice la existencia de un matching estable, mas bien construimos un nuevo algoritmo el cual encontrara los matchings estables siempre y cuando estos existan. El algoritmo que se ha construido funciona bien bajo preferencias responsive así como también bajo la presencia de preferencias lexicograficas.

Para la construcción de este algoritmo lo que se ha hecho es adaptar un algoritmo de la familia de algoritmos “Branch and Bound”, los cuales comúnmente son usados en problemas de maximización donde las variables son discretas o tenemos una mezcla de variables discretas y continuas. El algoritmo ha sido construido de tal forma que se puede garantizar que siempre encontrara los matchings estables siempre y cuando estos existan. Algo importante a mencionar es que, a diferencia del algoritmo de aceptación diferida nuestro algoritmo encuentra no solo uno sino todos los matchings estables, esto es importante ya que no existe un criterio general para la clasificación de los matchings es decir no podemos decidir que matching es mejor o peor, resultado que ya ha sido previamente establecido por Aldershof y Carducci (1996), por lo que resulta conveniente tener todos los matchings a nuestra disposición.

El resto de la tesis esta organizada como sigue, en la sección 2 presentamos el modelo formal e introducimos nuestro concepto de estabilidad para el caso que tenemos parejas en el mercado. En la sección 3 presentamos el problema que surge con la introducción de parejas en el mercado y discutimos algunas de las características del algoritmo de AD. En la sección 4 presentamos el algoritmo de Ramificación con Parejas (RP) así como un ejemplo de su funcionamiento. En la sección 5 probamos nuestro principal resultado que es que bajo el algoritmo RP se garantiza que los matchings estables son encontrados. Finalmente en la sección 6 presentamos nuestras conclusiones.

2. El modelo

Nuestro problema estar definido por los siguientes conjuntos:

- $H = \{h_1, h_2, \dots, h_L\}$ Conjunto de hospitales.
- $S = \{s_1, s_2, \dots, s_M\}$ Conjunto de estudiantes solteros
- $S_c = \{c_1, c_2, \dots, c_N\}$ Conjunto de estudiantes con pareja.

De tal forma que $M + N = L$

Los estudiantes del conjunto S_c forman parejas, donde una pareja representativa esta dada por $P_i = (c_l, c_k)$ donde $l \neq k$ y $\{l, k\} = 1, \dots, N$. Estas parejas forman el conjunto de parejas $P = \left\{ P_1, P_2, \dots, P_{\frac{N}{2}} \right\}$. Cada hospital solo tiene una plaza disponible.

Preferencias de los hospitales. Cada hospital $h \in H$ tiene una relación de preferencias estricta, transitiva y completa \succ_h sobre $S \cup S_c \cup \phi$. Un ejemplo de esto será:

$$\left. \begin{array}{l} \succ_h \\ s_1 \\ c_1 \\ \phi \\ s_2 \end{array} \right\} \text{ Es decir, el hospital } h \text{ prefiere al estudiante } s_1 \text{ al estudiante } c_1 \text{ y prefiere quedarse sin estudiante asignado } \phi \text{ ha tener al estudiante } s_2 \text{ . Entonces definimos } \succ_H \text{ como los perfiles de preferencias de los hospitales.}$$

Preferencias de los estudiantes solteros. De manera similar, los estudiantes $s \in S$ tienen una relación de preferencias estricta, transitiva y completa \succ_s sobre $H \cup \phi$. Un ejemplo de esto será:

$$\left. \begin{array}{l} \succ_s \\ h_1 \\ h_2 \\ \phi \\ h_3 \end{array} \right\} \text{ Es decir, el estudiante } s \text{ prefiere al hospital } h_1 \text{ al hospital } h_2 \text{ y prefiere quedarse sin ser asignado } \phi \text{ ha tener que ser asignado al hospital } h_3 \text{ . Entonces definimos } \succ_S \text{ como los perfiles de preferencias de los estudiantes solteros}$$

Preferencias de las parejas. Cada miembro de una pareja $c \in p$ tiene una relación de preferencias estricta, transitiva y completa \succ_c sobre $(H \cup \phi)^2$. Un ejemplo de esto será:

$$\left. \begin{array}{l} \succ_{c'} \\ h_1, h_2 \\ h_3, h_4 \\ \phi \end{array} \quad \begin{array}{l} \succ_c \\ h_1, h_4 \\ h_3, h_2 \\ \phi \end{array} \right\} \text{ Es decir, el estudiante } c \text{ prefiere al hospital } h_1 \text{ y el estudiante } c', \text{ ambos miembros de la pareja } p, \text{ prefieren al hospital } h_3 \text{ y prefieren quedarse sin ser asignado } \phi \text{ ha tener que ser asignado a cualquier otro hospital. Como se ve no imponemos condiciones sobre la estructura de preferencias de las parejas. Definimos entonces } \succ_p \text{ como los perfiles de preferencias de las parejas}$$

Entonces un mercado estará definido por $\langle H, S, S_c, \succ_H, \succ_S, \succ_P \rangle$.

El problema de matching consiste en asignar cada estudiante a cada hospital teniendo en consideración las preferencias de ambos lados del mercado. Formalmente definimos un matching μ como un mapeo que va de $H \cup S \cup S_c$ a $H \cup S \cup S_c \cup \phi$, tal que para todo hospital $h \in H$, $s \in S$ y $c \in S_c$: (a) ya sea $|\mu(h)| = 1$ y $\mu(h) \in S \cup S_c$, o $\mu(h) = \phi$; (b) ya sea $|\mu(s)| = 1$ y $\mu(s) \in H$, o $\mu(s) = \phi$; (c) $|\mu(c, c')| \leq 2$ y $\mu(c, c') \in (H \cup \phi)^2$, o $\mu(c, c') = \phi$.

El matching que buscamos tiene que ser estable, por lo que es necesario definir nuestro concepto de estabilidad en este mercado. Dado un matching μ decimos que es estable si:

1. Es individualmente racional si:
 - a) Para todo hospital $h \in H$: $\mu(h) \succ_h \phi$, es decir, cada hospital prefiere estrictamente a su actual asignación a quedarse sin asignación
 - b) Para todo estudiante $s \in S$: $\mu(s) \succ_s \phi$, es decir, cada estudiante prefiere estrictamente a su actual asignación a quedarse sin asignación
 - c) Para toda pareja $p \in P$: $\mu(c, c') \succ_{c, c'} \phi$, es decir, cada pareja prefiere estrictamente su actual asignación a quedarse sin asignación
2. No hay pares bloqueadores. Definimos un par bloqueador como:
 - a) Entre estudiantes solteros y hospitales: es un par (h, s) donde $h \in H$ y $s \in S$ tal que $h \succ_s \mu(s)$ y $s \succ_h \mu(h)$
 - b) Entre parejas y hospitales: es una pareja de estudiantes y un par de hospitales (h, h', c, c') donde $(h, h') \in H$ y $(c, c') \in S_c$ tal que:
 - i. $h \succ_c \mu(c)$ y $c \succ_h \mu(h)$ Es decir solo un miembro de la pareja bloquea, en vista de esto el otro miembro renuncia a su plaza
 - ii. $(h, h') \succ_c \mu(c, c')$ $(h, h') \succ_{c'} \mu(c, c')$ y $c \succ_h \mu(h)$ y $c \succ_{h'} \mu(h')$ es decir ambos miembros de la pareja bloquean al mismo tiempo.

Definición 1. *Un matching μ es estable si es individualmente racional (tanto para los hospitales como para los estudiantes solteros y con pareja) y no existen pares bloqueadores.*

3. El problema

3.1 Existencia

La presencia de parejas en el mercado viene a complicar el problema de matching, en primer lugar al tener parejas en el mercado ya no podemos garantizar que el conjunto de matchings estables sea vacío. Roth (1984) ya ha planteado este resultado:

Teorema 1. *En un mercado donde algunos agentes son pareja, el conjunto de matchings estables puede ser vacío.*

El resultado de Roth da espacio a la búsqueda de condiciones bajo las cuales se pueda garantizar que el conjunto de matchings estables sea no vacío. Klaus Klijn y Massó (2003) dan condiciones sobre la estructura de preferencias que garantizan que el conjunto de estables sea no vacío:

Definición 2. *La pareja $P_i = (c, c')$ tiene preferencias responsive si existen unas preferencias individuales \succ_c y $\succ_{c'}$ tal que para todo $h_p, h_q, h_r \in H$, [$h_p \succ_c h_r$ implica $(h_p, h_q) \succ_{P_i} (h_r, h_q)$] y [$h_p \succ_{c'} h_r$ implica $(h_q, h_p) \succ_{P_i} (h_q, h_r)$]. Si estas preferencias \succ_c y $\succ_{c'}$ existen, entonces son únicas.*

Lo que estas preferencias implican es que cada miembro de una pareja es en principio un estudiante soltero y al unirse en pareja conservan sus preferencias, otra implicación es que mejoras unilaterales de un miembro de la pareja mejora a la pareja. Bajo estas preferencias Klaus y Klijn (2003) encuentran que se puede garantizar la existencia de un matching estable:

Teorema 2. *Sea $\langle \succ_H, \succ_P \rangle$ el mercado donde las parejas tienen preferencias responsive. Entonces, cualquier matching que es estable en el mercado con solo solteros asociado a este mercado con parejas $\langle \succ_H, \succ_{S(P)} \rangle$ es también estable para $\langle \succ_H, \succ_P \rangle$. En particular, existe un matching estable para $\langle \succ_H, \succ_P \rangle$*

Este teorema aprovecha el hecho de que las preferencias responsive nos permiten descomponer a las parejas en individuos, lo que nos lleva a tener el mismo mercado

matching con solteros que teníamos antes, y sabemos que cuando tenemos solo estudiantes solteros podemos garantizar la existencia de matching estables.

Klaus Klijn y Massó (2003) dan en este trabajo un ejemplo, usando sus preferencias responsive, en el cual el nuevo algoritmo del NRMP no encuentre el matching estable que sabemos existe dado el teorema 2. El ejemplo es el siguiente:

h_1	h_2	h_3	h_4	$\{s_1, s_2\}$	$\{s_3, s_4\}$
s_2	s_2	s_2	s_2	h_1h_2	h_2h_3
s_3	s_3	s_3	s_3	h_1h_3	h_2h_4
s_1	s_1	s_1	s_1	h_1h_4	h_2h_1
s_4	s_4	s_4	s_4	h_2h_1	h_1h_3
				h_2h_3	h_1h_4
				h_2h_4	h_1h_2
				h_3h_1	h_3h_4
				h_3h_2	h_3h_2
				h_3h_4	h_3h_1
				h_4h_1	h_4h_3
				h_4h_2	h_4h_2
				h_4h_3	h_4h_1

Bajo este ejemplo, el NRMP no encuentra el matching estable existente que es $\mu(H) = \{z_2, z_3, z_1, z_4\}$. Las iteraciones que se siguen con el nuevo algoritmo del NRMP son las siguientes: el algoritmo empieza con el matching vacío $\mu^0(H) = \{\phi, \phi, \phi, \phi\}$ y hace ciclos sobre matchings inestables $\mu^I(H) = \{\phi, s_3, s_4, \phi\}$, $\mu^{II}(H) = \{s_1, s_2, \phi, \phi\}$, $\mu^{III}(H) = \{s_3, \phi, s_4, \phi\}$, $\mu^{IV}(H) = \{s_2, s_1, \phi, \phi\}$ y finalmente de regreso a $\mu^I(H)$. Con este ejemplo queda establecido que el NRMP no captura bien las preferencias de las parejas. Aun que esta condición sobre las preferencias garantiza la existencia de estabilidad, esta estructura de preferencias no parece capturar bien la distancia geográfica que es en el último caso lo más importante para las parejas este problema se analiza en el trabajo de Cantala (2004). Es por eso también que nosotros hemos decidido no imponer restricciones a las preferencias de las parejas.

3.2 Algoritmo de Aceptación Diferida (AD) vs. Algoritmo de Ramificación Con Parejas (RP).

En el ejemplo anterior el matching estable puede ser encontrado con el AD, entonces, una buena pregunta es ¿por que necesitamos un nuevo algoritmo? Algo que debemos tener presente es que el AD si bien a probado ser eficaz cuando las preferencias son responsive, sin embargo, en general no sabemos como responde ante diferentes estructuras de preferencias, y dado que es complicado distinguir a que estructura de preferencias nos enfrentamos lo mas conveniente es contar con un mecanismo que no nos limite a solo cierto tipo de estructura de preferencias, ya que la otra soluciones seria encontrar un mecanismo para identificar cuando podemos usar el algoritmo de AD pero eso parece mas difícil, además aun cuando podamos identificar cuando usarlo ¿que pasaría con los casos cuando no lo podamos usar? En este sentido contar con un algoritmo que nos permita prescindir de definir estructuras de preferencias es mejor que tener un algoritmo que solo sirva para un contado numero de estructura de preferencias. Otra ventaja del algoritmo RP es que nos permite encontrar todos los matchings estables. Algo que sabemos que usando el algoritmo AD es que si son los hospitales los que aceptan o rechazan las ofertas de los estudiantes llegaremos a un matching estable que es el menos preferido para los estudiantes y el más preferido para los hospitales, y viceversa. Es decir el resultado del AD dependerá de que lado del mercado haga las ofertas y que lado del mercado las acepte, mientras que nuestro algoritmo nos permite encontrar todos los matching estables lo cual resulta en una mejora ya que existiran casos en que para algunos estudiantes (o para algunos hospitales) un matching resultara mejor que otro, sin embargo, para el resto de los estudiantes (u hospitales) ese mismo matching resultara ser el peor, lo que tenemos es que no hay acuerdo entre estudiantes de que matching es mejor. Resulta entonces mejor tener todos los matchings disponibles.

4. El Algoritmo de Ramificación con Parejas (RP)

El algoritmo que aquí se presenta pertenece a la familia de algoritmos “Branch and Bound”, es decir es un algoritmo de “Acotación y Ramificación.” En este caso no existe una fase de acotación por lo que solo usamos la parte de la ramificación. Que como veremos resulta bastante conveniente ya que nos permite encontrar todos los matchings estables siempre y cuando estos existan.

Consideremos el siguiente mercado con parejas: $\langle H, S, S_c, \succ_H, \succ_S, \succ_P \rangle$

Para el algoritmo RP definimos el siguiente conjunto: $Z = \{z_1, z_2, \dots, z_{M+N}\}$ donde $z \in S \cup S_c$. Es decir Z es un conjunto unión de los conjuntos de estudiantes solteros y

con pareja, es decir, es un conjunto que agrupa en un solo conjunto tanto a los estudiantes solteros como a los estudiantes con pareja: donde los primeros M estudiantes son solteros y los siguientes N son miembros de alguna pareja.

Definimos también el conjunto $RP\langle H, S, S_c, \succ_H, \succ_S, \succ_P \rangle$ como el conjunto de soluciones obtenidas a través del algoritmo RP en el mercado definido por $\langle H, S, S_c, \succ_H, \succ_S, \succ_P \rangle$. Para todos los hospitales $h \in H$, sea $\beta_h = \{z \in Z \mid z \succ_h \phi\}$ es decir el conjunto de estudiantes individualmente racionales para el hospital h .

Para todos los estudiantes $z \in Z$ sea $\beta_z = \{h \in H \mid h \succ_z \phi\}$ es decir el conjunto de hospitales individualmente racionales para el estudiante z .

En cada iteración escogeremos solo estudiantes del conjunto β_h y cada estudiante solo será escogido si el hospital para el que es propuesto pertenece al conjunto β_z .

Paso 0.

Acomodamos a los estudiantes siguiendo un criterio de optimalidad reciproca:

- Un estudiante soltero y un hospital cumplen con optimalidad reciproca si y solo si: Para un estudiante z_i tenemos que $h_l \succ_{z_i} h_m \quad \forall m \neq l$ y para el hospital h_l tenemos que $z_i \succ_{h_l} z_j \quad \forall j \neq i$.
- Una pareja de estudiantes $\{z_i, z_j\}$ y una pareja de hospitales $\{h_l, h_k\}$ cumplen con optimalidad reciproca si y solo si: Para z_i tenemos que $(h_l, h_k) \succ_{z_i} (h_m, h_n)$ y para z_j tenemos que $(h_k, h_l) \succ_{z_j} (h_n, h_m) \quad \forall (h_n, h_m) \neq (h_l, h_k)$ y para el hospital h_l tenemos que $z_i \succ_{h_l} z_n \quad \forall n \neq j$ y para el hospital h_k tenemos que $z_j \succ_{h_k} z_n \quad \forall n \neq i$.

Los estudiantes y hospitales emparejados por este criterio son retirados de las listas de preferencias, de los estudiantes y hospitales remanentes. El proceso es iterativo por lo que una vez retirados dichos estudiantes y hospitales volvemos acomodar a los estudiantes que cumplen con el criterio de optimalidad reciproca.

El proceso se repite hasta que ya no hay estudiantes y hospitales que cumplan con el principio de optimalidad reciproca.

Paso Principal

Antes de continuar definimos el conjunto Ω que será el conjunto de todas las soluciones que hay que escudriñar, este conjunto tiene 2 características:

- Solo contiene soluciones de todos los matchings individualmente racionales (es decir solo contienen estudiantes asignados que se encuentran en β_h y los estudiantes solo están asignados a hospitales que se encuentran en β_z) y
- Los estudiantes y hospitales óptimamente recíprocos ya se encuentran asignados de manera fija.

En cada iteración tomaremos un conjunto de soluciones $\omega = \{z_1, \dots, z_i, \phi, \dots, \phi\}$ este conjunto nos representa que tenemos i estudiantes asignados en los primeros i hospitales y representa todas las soluciones que tienen esta asignación en sus primeros i hospitales; dicho conjunto es retirado del conjunto Ω es decir tenemos ahora $\Omega := \Omega \setminus \omega$, nuestra intención será decidir si dentro de este conjunto de soluciones se encuentra el matching estable que buscamos:

- Si el matching no se encuentra dentro de este conjunto de soluciones estas son retiradas permanentemente de Ω . Es decir son descartadas todas las soluciones que tienen la estructura de ω . Tomamos entonces otro grupo de soluciones $\omega_2 = \{z_2, \dots, z_i, \phi, \dots, \phi\}$
- Si la solución puede encontrarse dentro de este conjunto en la siguiente iteración ramificamos, es decir, de ω tomamos un subgrupo ω' donde $\omega' = \{z_1, \dots, z_i, z_{i+1}, \dots, \phi\}$. Donde $z_{i+1} \in \beta_{h_{i+1}}$ y $h_{i+1} \in \beta_{z_{i+1}}$. Por tanto tendremos tantos subgrupos como estudiantes sin asignar tengamos en ω . En cada iteración subsiguiente estos subgrupos serán verificados siguiendo el mismo proceso. Añadimos estos subgrupos al conjunto de soluciones de tal forma que ahora tenemos $\Omega := \Omega \cup \{\omega', \omega', \dots\}$.

Continuaremos con este proceso hasta que $\Omega = \emptyset$ es decir ya no haya más soluciones que verificar.

Para decidir si dentro de ω se puede encontrar la solución buscada nos preguntamos: En el matching final estable ¿Pueden estar los estudiantes $\{z_1, \dots, z_i\}$ asignados a los hospitales $\{h_1, \dots, h_i\}$? La respuesta a esta pregunta dependerá de en que iteración del proceso nos encontremos:

Primer Iteración:

En la primera iteración escogemos el ω mas general, es decir escogemos el ω donde solo haya un estudiante asignado a un hospital (además de los estudiantes que ya se encuentran fijos debido al proceso de optimalidad reciproca). Tomamos $\omega^1 = \{z_1, \phi, \phi, \dots, \phi\}$, entonces para responder la pregunta en esta primera iteración solo tenemos que verificar si el estudiante $z_1 \in \beta_{h_1}$ y $h_1 \in \beta_{z_1}$ lo que resulta trivial por la forma en que hemos definido ω . Dado que la respuesta a la pregunta fue afirmativa ramificamos ahora tenemos el $\omega^2 = \{z_1, z_2, \phi, \dots, \phi\}$

Antes de continuar definimos el siguiente conjunto $\omega^i(h) = \{h_1, \dots, h_i\}$ que será el conjunto de hospitales que en la iteración i tienen un estudiante asignado del mismo modo definimos $\omega^i(z) = \{z_1, \dots, z_i\}$ como el conjunto de estudiantes que en la iteración i ya están asignados a algún hospital. Además definimos $\mu^i(h_j)$ como el matching del hospital h_j en la iteración i , y $\mu^i(z_j)$ como el matching del estudiante z_j en la iteración i .

I-esima Iteración:

Nos preguntamos: En el matching final estable ¿Pueden estar los estudiantes $\{z_1, \dots, z_i\}$ asignados a los hospitales $\{h_1, \dots, h_i\}$?. Para responder tenemos que verificar si el estudiante y el hospital en esta ramificación pueden bloquear. Para esto seguimos el siguiente proceso:

- Verificamos si dentro del conjunto $\omega^{i-1}(h)$ hay algún hospital con el que el estudiante asignado en esta ramificación pueda bloquear. En este caso tenemos que preguntarnos si ¿el estudiante z_i puede bloquear con algún hospital $h \in \omega^{i-1}(h)$?
- Verificamos si dentro del conjunto $\omega^{i-1}(z)$ hay algún estudiante con el que el hospital asignado en esta ramificación pueda bloquear. En este caso tenemos que preguntarnos si ¿el hospital h_i puede bloquear con algún estudiante $z \in \omega^{i-1}(z)$?

Para llevar a cabo estas verificaciones definimos los siguientes conjuntos:

Caso 1 Bloqueos en el caso de solteros

- a) $\rho(h_i) = \{z \succ_{h_i} \mu^i(h) \mid z \in \omega^{i-1}(z)\}$ Es decir el conjunto de estudiantes que ya han sido asignados en las iteraciones anteriores y que son estrictamente preferidos para el hospital h_i al que estamos asignando un estudiante en esta iteración. Entonces, $\forall z_x \in \rho(h_i) \cap S$ es decir para los estudiantes solteros con los que quiere bloquear construimos el conjunto $\rho(z) = \{h \succ_z \mu(z) \mid h \in \omega^i(h)\}$. Si $h_i \in \bigcup \rho(z)$ entonces hay un estudiante soltero $z \in \omega^{i-1}(z)$ con el que el hospital h_i puede bloquear. Ahora vemos que pasa cuando quiere bloquear con estudiantes miembros de una pareja, entonces $\forall z \in \rho(h_i) \cap S_c$ construimos el conjunto $\rho(z, z') = \{(h, h') \succ_z \mu(z) \mid h \in \omega^i(h)\}$ Si $h_i \in \bigcup_x \rho(z_x)$ y $\exists h' \in \omega^i(h)$ tal que: $(h, h') \succ_z (\mu(z), \mu(z'))$ y $(h', h) \succ_{z'} (\mu(z'), \mu(z))$ entonces hay un estudiante con pareja $z \in \omega^{i-1}(z)$ con el que el hospital h_i puede bloquear.
- b) $\rho(z_i) = \{h \succ_{z_i} \mu^i(z_i) \mid h \in \omega^{i-1}(h)\}$ Es decir el conjunto de hospitales que ya han sido asignados en las iteraciones anteriores y que son estrictamente preferidos para el estudiante z_i que es asignado en esta iteración. Entonces $\forall h \in \rho(z_i)$ construimos el conjunto $\rho(h) = \{z \succ_h \mu(h) \mid z \in \omega^i(z)\}$. Si $z_i \in \bigcup \rho(h)$ entonces hay un hospital $h \in \omega^{i-1}(h)$ con el que el estudiante z_i puede bloquear

Caso 2 de bloqueos con parejas

En este caso en la iteración I se ha acomodado a un estudiante del conjunto S_c .

- a) De manera similar al caso de los solteros formamos el conjunto $\rho(h_i) = \{z \succ_{h_i} \mu^i(h) \mid z \in \omega^{i-1}(z)\}$ es decir el conjunto de estudiantes que son más preferidos para el hospital h_i que su actual asignación. Entonces, $\forall z_x \in \rho(h_i) \cap S$ es decir para los estudiantes solteros con los que quiere bloquear construimos el conjunto $\rho(z) = \{h \succ_z \mu(z) \mid h \in \omega^i(h)\}$. Si $h_i \in \bigcup_x \rho(x, z)$ entonces hay un estudiante soltero $z \in \omega^{i-1}(z)$ con el que el hospital h_i puede bloquear. Ahora vemos que pasa cuando quiere bloquear con estudiantes miembros de una pareja, entonces $\forall z \in \rho(h_i) \cap S_c$ construimos el conjunto $\rho(z, z') = \{(h, h') \succ_z \mu(z) \mid h \in \omega^i(h)\}$ Si $h_i \in \bigcup_x \rho(z)$ y $\exists h' \in \omega^i(h)$

tal que: $(h, h') \succ_z (\mu(z), \mu(z'))$ y $(h', h) \succ_{z'} (\mu(z'), \mu(z))$ entonces hay un estudiante con pareja $z \in \omega^{i-1}(z)$ con el que el hospital h_i puede bloquear.

- b) $\rho(z_i) = \{h \succ_{z_i} \mu^i(z_i) \mid h \in \omega^{i-1}(h)\}$ es decir el conjunto de hospitales que ya han sido asignados en las iteraciones anteriores y que son estrictamente preferidos para el estudiante z_i que es asignado en esta iteración, z_i es miembro de la pareja (z_i, z_j) . Entonces $\forall h \in \rho(z_i) \exists h' \in \omega^{i-1}(h)$ tal que $(h, h') \succ_z (\mu(z), \mu(z'))$ y $(h', h) \succ_{z'} (\mu(z'), \mu(z))$ además $z_j \succ_{h'} \mu(h')$ también construimos el conjunto $\rho(h) = \{z \succ_h \mu(h) \mid z \in \omega^i(z)\}$. Si $z_i \in \bigcup \rho(h)$ entonces hay un hospital $h \in \omega^{i-1}(h)$ con el que el estudiante z_i puede bloquear

Si alguna de las dos condiciones a) y/o b) (ya sea para el caos 1 o el caso 2) anteriores se cumple retiramos todas las soluciones que tengan la estructura $\omega^i = \{z_1, \dots, z_i, \phi, \dots, \phi\}$ es decir ahora tenemos $\Omega := \Omega \setminus \omega^i$. Habiendo hecho esto tendremos que volver a la primera iteración con el conjunto ω más general es decir $\omega^1 = \{z_1, \phi, \phi, \dots, \phi\}$.

Si ninguna de las condiciones se cumple, ramificamos a $\omega^{i+1} = \{z_1, \dots, z_i, z_{i+1}, \phi, \dots, \phi\}$.

Cuando ya no queden estudiantes que asignar o todos los hospitales ya tengan algún estudiante asignado podemos decir que hemos encontrado un matching estable, sin embargo si $\Omega \neq \phi$ esto quiere decir que aun existen aun soluciones por verificar y por tanto es posible que mas matchings estables. Por lo el proceso habrá de repetirse hasta que $\Omega = \phi$.

Una vez que hemos definido el proceso del algoritmo podemos dar nuestro resultado principal. Primero definamos el conjunto $E(H, S, S_c, \succ_H, \succ_S, \succ_{S_c})$ como el conjunto de matchings estables en le mercado definido por $\langle H, S, S_c, \succ_H, \succ_S, \succ_P \rangle$.

Teorema 3.

$$\forall \langle H, S, S_c, \succ_H, \succ_S, \succ_P \rangle E(H, S, S_c, \succ_H, \succ_S, \succ_{S_c}) = RP \langle H, S, S_c, \succ_H, \succ_S, \succ_P \rangle$$

En palabras sencillas lo que este teorema implica es que con el algoritmo RP siempre encuentra el matching (o los matchings estables) siempre y cuando este exista, es decir, cuando el conjunto de soluciones estables sea no vacío.

Ejemplos:

Consideremos el siguiente mercado de matching:

$$H = \{h_1, h_2, h_3, h_4\}$$

$$S_c = \{s_1, s_2, s_3, s_4\}$$

$$P = \{P_1, P_2\} \text{ con } P_1 = \{s_1, s_2\} \text{ y } P_2 = \{s_3, s_4\}$$

Formamos el conjunto $Z = \{z_1, z_2, z_3, z_4\}$

El ejemplo que resolvemos es el que ya habíamos presentado en la sección 3 y que como vimos hace fallar al NRMP. A continuación se muestra las listas de preferencias de los hospitales y de las parejas:

h_1	h_2	h_3	h_4	$\{s_1, s_2\}$	$\{s_3, s_4\}$
s_2	s_2	s_2	s_2	$h_1 h_2$	$h_2 h_3$
s_3	s_3	s_3	s_3	$h_1 h_3$	$h_2 h_4$
s_1	s_1	s_1	s_1	$h_1 h_4$	$h_2 h_1$
s_4	s_4	s_4	s_4	$h_2 h_1$	$h_1 h_3$
				$h_2 h_3$	$h_1 h_4$
				$h_2 h_4$	$h_1 h_2$
				$h_3 h_1$	$h_3 h_4$
				$h_3 h_2$	$h_3 h_2$
				$h_3 h_4$	$h_3 h_1$
				$h_4 h_1$	$h_4 h_3$
				$h_4 h_2$	$h_4 h_2$
				$h_4 h_3$	$h_4 h_1$

Siguiendo nuestro algoritmo tenemos:

Paso 0. Ninguna pareja ni ningún hospital cumplen con optimalidad reciproca

Iteración 1¹.

Escogemos $\omega^1 = \{z_1, z_2, \phi, \phi\}$ como la pareja es aceptable para la pareja de hospitales $\{h_1, h_2\}$. Ramificamos a $\omega^2 = \{z_1, z_2, z_3, z_4\}$

Iteración 2

- Formamos el conjunto:

$$\rho(3,4, z, z) = \{(h_1, h_2), (h_1, h_4), (h_1, h_3), (h_2, h_1), (h_2, h_4), (h_2, h_3)\}$$

- Formamos el conjunto:

$$\rho(3,4, h, h) = \{(z_2, z_1)\}$$

a) Para las parejas de hospitales que forman parte del conjunto $\rho(3,4, z, z)$ encontramos el conjunto de estudiantes estrictamente mas preferidos.

$$\rho(1,2, h, h) = \{\phi\}$$

$$\rho(1,4, h, h) = \{(z_3, z_4), (z_1, z_2), (z_2, z_1)\}$$

$$\rho(1,3, h, h) = \{(z_3, z_4), (z_1, z_2), (z_2, z_1)\}$$

$$\rho(2,1, h, h) = \{\phi\}$$

$$\rho(2,4, h, h) = \{(z_2, z_1)\}$$

$$\rho(2,3, h, h) = \{\phi\}$$

Entonces $(z_3, z_4) \in \bigcup_{x,y} \rho(x, y, h, h)$

b) Haciendo lo mismo para el conjunto de estudiantes que forma parte del conjunto

$$\rho(3,4, h, h) = \{(z_2, z_1)\}$$

$$\rho(2,1, z, z) = \{\phi\}$$

Dado que la condición a) se cumple podemos decir que hay un par bloqueador y por tanto podemos descartar todas las soluciones con la forma $\omega^2 = \{z_1, z_2, z_3, z_4\}$

Iteración 3.

Escogemos $\omega^3 = \{z_1, \phi, z_2, \phi\}$ como la pareja es aceptable para la pareja de hospitales $\{h_2, h_4\}$. Ramificamos a $\omega^4 = \{z_2, z_3, z_2, z_4\}$.

¹ Los conjuntos ω han sido escogido *ad hoc*, lo que no hace diferencia ya que el algoritmo es exhaustivo y no se detiene hasta haber revisado todas las soluciones. Simplemente se escoge este orden por cuestiones de hacer un poco más breve el ejemplo. El resultado del algoritmo será el mismo independientemente de los ω que se nos ocurra utilizar.

Iteración 4

- Formamos el conjunto:

$$\rho(3,4, z, z) = \{(h_2, h_3)\}$$

- Formamos el conjunto:

$$\rho(2,4, h, h) = \{(z_2, z_1)\}$$

- a) Para las parejas de hospitales que forman parte del conjunto $\rho(3,4, z, z)$ encontramos el conjunto de estudiantes estrictamente mas preferidos.

$$\rho(2,3, h, h) = \{z_2, z_1\}$$

Entonces $(z_3, z_4) \notin \bigcup_{x,y} \rho(x, y, h, h)$

- b) Haciendo lo mismo para el conjunto de estudiantes que forma parte del conjunto

$$\rho(2,4, h, h) = \{(z_2, z_1)\}$$

$$\rho(2,1, z, z) = \{\emptyset\}$$

Entonces $(h_1, h_3) \notin \bigcup_{x,y} \rho(x, y, z, z)$

Dado que las condiciones no se cumplen el matching final es: $\omega^4 = \{z_2, z_3, z_1, z_4\}$

A destacar que este es el ejemplo planteado por Klaus Klijn y Massó en su paper del 2003 en el cual el actual algoritmo del NRMP no encontraba el matching estable.

5. Prueba Del Teorema 3

Consideremos el siguiente mercado:

$$H = \{h_1, h_2, \dots, h_L\}$$

$$S = \{s_1, s_2, \dots, s_M\}$$

$$S_c = \{c_1, c_2, \dots, c_N\}$$

De tal forma que $M + N = L$

$$P = \left\{ P_1, P_2, \dots, P_{\frac{N}{2}} \right\} \text{ Conjunto de parejas. Donde } P_n = (c_l, c_k)$$

Dado este mercado el problema de matching podemos garantizar que el algoritmo que se ha construido esta diseñado de forma tal que siempre encuentra el matching estable si

es que este existe. La prueba se realiza en 2 etapas: primero verificamos que nuestro criterio de optimalidad reciproca sea estable y segundo verificamos que todo la demás parte del algoritmo genere resultados estables

Optimalidad Reciproca y Estabilidad

Lo primero ha hacer notar es que el paso cero, en donde asignamos de manera fija hospitales y estudiantes siguiendo nuestro criterio de optimalidad reciproca, provoca que estos estudiantes al final del algoritmo quieran bloquear o que durante alguna iteración del algoritmo RP algún estudiante u hospital pueda bloquear con estos estudiantes y hospitales que aquí se han asignado. Estos bloqueos nunca ocurriran, la prueba de que esto no sucede se sigue directamente de nuestra definición de pares bloqueadores y de la definición de nuestro criterio de optimalidad reciproca. Supongamos que un estudiante asignado por medio de este criterio una vez obtenida nuestra solución (o soluciones) final quiere bloquear con algún hospital, esto no puede pasar ya que al ser asignado por nuestro criterio de optimalidad reciproca el estudiante se encuentra asignado a su mejor opción disponible, y dado que solo bloqueara con hospitales que sean mejor a su actual asignación esto nunca pasara. Lo mismo pasa para hospitales una vez asignados con nuestro criterio de optimalidad reciproca no podrán bloquear.

Ramificaciones y Estabilidad

Según hemos definido para nuestro algoritmo Ω será el conjunto de todas las soluciones que hay que escudriñar. Lo primero que podemos decir que Ω será un conjunto finito, esto claro por que el hecho de que el conjunto de estudiantes así como el conjunto de hospitales es finito y por tanto el conjunto de mathings posibles será finito. De lo anterior se desprende que nuestro algoritmo siempre termina. Esto es claro ya que lo que hacemos en cada iteración del algoritmo es escoger un conjunto de soluciones con cierta estructura, $\omega = \{z_1, \dots, z_i, \phi, \dots, \phi\}$, que pueden ser o no ser retiradas de Ω , dado que este es un conjunto finito después de cierto numero de iteraciones habremos descartado todas las soluciones que no forman parte del resultado de forma tal que $\Omega = \phi$.

En principio podrían presentar diferentes errores, se argumenta que estos errores no se pueden cometer siguiendo nuestro algoritmo.

Error 1. Una solución no fue escudriñada.

Dado que el algoritmo no se detiene hasta que $\Omega = \emptyset$ es imposible que queden soluciones sin escudriñar.

Error 2. Una solución que no debía ser descartada fue descartada.

Supongamos que tenemos la solución $\omega^i = \{z_1, \dots, z_i, \emptyset, \dots, \emptyset\}$, si esta por ser descartada tiene que ser el caso que alguna de las 2 condiciones se este cumpliendo (o ambas):

1. Dado el conjunto $\rho(z_i) = \{h \succ_{z_i} \mu^i(z) \mid h \in \omega^{i-1}(h)\}$. Entonces $\forall h_x \in \rho(i, z)$ construimos $\rho(x, h) = \{z \succ_{h_x} \mu^x(h) \mid z \in \omega^i(x)\}$. Y sabemos que $z_i \in \bigcup_x \rho(x, h)$.
2. Dado el conjunto $\rho(h_i) = \{z \succ_{h_i} \mu^i(h) \mid z \in \omega^{i-1}(z)\}$. Entonces $\forall z_x \in \rho(i, h)$ construimos $\rho(x, h) = \{z \succ_{h_x} \mu^x(h) \mid z \in \omega^i(x)\}$. Y sabemos que $z_i \in \bigcup_x \rho(x, h)$.

Si la solución no debía de ser descartada aun cuando estas condiciones se estén cumpliendo significa que las soluciones finales tendrán un par bloqueador, por lo cual no podría ser una solución de nuestro problema, por tanto si la solución es descartada es por que debió de ser descarta. Nunca se descartan soluciones que no deban serlo.

Error 3. Se acepta una solución que no debió de haber sido aceptada.

Supongamos que tenemos la solución $\omega^i = \{z_1, \dots, z_i, \emptyset, \dots, \emptyset\}$, y esta por ser aceptada tiene que ser el caso que alguna ninguna de las 2 condiciones se este cumpliendo:

1. Dado el conjunto $\rho(i, z) = \{h \succ_{z_i} \mu^i(z) \mid h \in \omega^{i-1}(h)\}$. Entonces $\forall h_x \in \rho(i, z)$ construimos $\rho(x, h) = \{z \succ_{h_x} \mu^x(h) \mid z \in \omega^i(x)\}$. Y sabemos que $z_i \in \bigcup_x \rho(x, h)$.
2. Dado el conjunto $\rho(i, h) = \{z \succ_{h_i} \mu^i(h) \mid z \in \omega^{i-1}(z)\}$. Entonces $\forall z_x \in \rho(i, h)$ construimos $\rho(x, h) = \{z \succ_{h_x} \mu^x(h) \mid z \in \omega^i(x)\}$. Y sabemos que $z_i \in \bigcup_x \rho(x, h)$.

Si la solución esta siendo aceptada es por que no hay pares bloqueadores, y por tanto el solución final será estable. El algoritmo nunca acepta soluciones que tengan pares bloqueadores por tanto si las soluciones son aceptadas es por que deben de ser aceptadas.

6. Conclusión

En este trabajo hemos elaborado un algoritmo que considera de manera adecuada las preferencias de las parejas para resolver el problema de matching. Nuestro algoritmo de “Acotación y Ramificación” nos permite encontrar todos los matchings estables cuando estos existan, lo cual es una mejora al algoritmo de aceptación diferida que solo nos permitía encontrar un matching estable. Este resultado es importante ya que no existe un criterio sobre que matchings deben ser más preferidos, es decir, no hay una convención sobre el ranking de los matchings.

Otra característica importante del algoritmo es que no fue necesario imponer condiciones sobre las preferencias de las parejas, caso contrario al trabajo realizado por Klaus Klijn y Massó (2003) en donde se hace uso de las preferencias responsive para encontrar los matchings estables y a partir de ellas se demuestra que el nuevo algoritmo del NRMP falla al tratar de obtener matchings estables en presencia de parejas. En este sentido nuestro trabajo es un buen complemento del trabajo de Klaus Klijn y Massó ya que mientras ellos encuentran la falla en el algoritmo del NRMP nosotros hemos encontrado un nuevo algoritmo que parece corregir este desperfecto.

Una de las propiedades más importantes del algoritmo RP es que nos da todos los matchings estables que el problema pudiera tener, para ver como funciona esto, y más que nada para ver por que es esto importante, consideremos el siguiente ejemplo:

$$H = \{h_1, h_2, h_3, h_4\}$$

$$S_c = \{c_1, c_2, c_3, c_4\} \text{ donde } P_1 = \{s_1, s_2\} \text{ y } P_2 = \{s_3, s_4\}$$

Las preferencias tanto de las parejas como los hospitales son:

h_1	h_2	h_3	h_4	(c_1c_2)	(c_3c_4)
c_1	c_3	c_1	c_3	h_1h_2	h_2h_1
c_4	c_2	c_4	c_2	h_3h_4	h_4h_3
ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

Usamos nuestro algoritmo RP:

Paso 0

No hay optimalidad reciproca

Iteración 1.

Escogemos $\omega^1 = \{c_1, c_2, \phi, \phi\}$ como la pareja es aceptable para la pareja de hospitales $\{h_1, h_2\}$. Ramificamos a $\omega^2 = \{z_1, z_2, z_3, z_4\}$

Iteración 2

- Formamos el conjunto:

$$\rho(3,4,c,c) = \{(h_1, h_2)\}$$

- Formamos el conjunto:

$$\rho(3,4,h,h) = \{(\phi)\}$$

- c) Para la pareja de hospitales que forman parte del conjunto $\rho(3,4,c,c)$ encontramos el conjunto de estudiantes estrictamente mas preferidos.

$$\rho(1,2,h,h) = \{\phi\}$$

Entonces $(c_3, c_4) \notin \bigcup_{x,y} \rho(x, y, h, h)$

- d) No hay conjunto que formar para los estudiantes mas preferidos

Dado que ni la condición a) ni b) se cumplen podemos decir que no hay un par bloqueador y por tanto podemos decir que $\omega^2 = \{c_1, c_2, c_3, c_4\}$ es un matching estable.

Sin embargo quedan soluciones por verificar.

Iteración 3

Escogemos $\omega^3 = \{c_3, c_4, \phi, \phi\}$ como la pareja es aceptable para la pareja de hospitales $\{h_1, h_2\}$. Ramificamos a $\omega^4 = \{c_3, c_4, c_1, c_2\}$

Iteración 4

- Formamos el conjunto:

$$\rho(1,2,c,c) = \{(h_1, h_2)\}$$

- Formamos el conjunto:

$$\rho(3,4,h,h) = \{(\phi)\}$$

- a) Para la pareja de hospitales que forman parte del conjunto $\rho(1,2,c,c)$ encontramos el conjunto de estudiantes estrictamente mas preferidos.

$$\rho(1,2,h,h) = \{\phi\}$$

Entonces $(c_1, c_2) \notin \bigcup_{x,y} \rho(x, y, h, h)$

- b) No hay conjunto que formar para los estudiantes mas preferidos

Dado que ni la condición a) ni b) se cumplen podemos decir que no hay un par bloqueador y por tanto podemos decir que $\omega^2 = \{c_3, c_4, c_1, c_2\}$ es un matching estable.

Aun quedan soluciones por verificar, sin embargo se puede verificar que estos son los únicos matchings estables, por lo que detendremos el algoritmo aquí.

Algo que podemos notar es que la pareja $P_1 = \{c_1, c_2\}$ prefieren el matching $\mu(H) = \omega^2 = \{c_1, c_2, c_3, c_4\}$ al matching $\mu'(H) = \omega^4 = \{c_3, c_4, c_1, c_2\}$ y para la pareja $P_2 = \{c_3, c_4\}$ es el matching $\mu'(H)$ el más preferido. Este ejemplo nos muestra que cuando tenemos mas de un matching estable no tenemos un criterio claro para decidir cual matching es mejor. Si tuviéramos un algoritmo que solo nos de uno de los matchings estables no sabríamos si este es mejor o el peor matching que podemos obtener, pero con nuestro algoritmo tenemos la oportunidad de ver todos los matchings estables y decidir que es lo que mas nos conviene.

Referencias.

Aldershof B., Carducci O. M.(1996) “Stable matchings with couples”, *Discrete Applied Mathematics*, 68, 203-207.

Cantala D. (2004) “Matching markets: The particular case of couples”, *Economic Bulletin*, Vol. 3, No 45 pp. 1-11

Klaus B., Klijn F. (2003) “Stable matching and preference of couples”, Mimeo Universitat Autònoma de Barcelona

Klaus B., Klijn F. y Massó J. (2003) “Some things couples always wanted to know about stable matchings (but were afraid to ask)”, Mimeo Universitat Autònoma de Barcelona

Roth A.E. (1984). “The evolution of the labor market for medical interns and residents: a case study in game theory”, *Journal of political Economy* 92. 991-1016