Inteligencia Artificial

Estado del Arte: Problema Stable Marriage

Patricio Ramirez Jofre

July 16, 2014

Evaluación

Resumen (5%):	____________
Introducción (5%):	____________
Definición del Problema (10%):	____________
Estado del Arte (35%):	____________
Modelo Matemático (20%):	____________
Conclusiones (20%):	____________
Bibliografía (5%):	____________
Nota Final (100%):	____________

Abstract

En este informe se representará y explicará el problema Stable Marriage, sus variaciones más importantes, que tipo de aproximaciones existen para la resolución de estas, que situaciones modela o resuelve y qué algoritmos son útiles. El Stable Marriage Problem (SM o SMP), consiste en un grupo de n hombres y otro grupo de n mujeres, los cuales cada uno de ellos tiene una lista de preferencia del sexo opuesto. El problema consiste en crear matrimonios estables (Stable), o sea, que no exista deseo por otra persona o divorcio. El problema SM puede resolverse en tiempo polinomial, pero las algunas variaciones de Stable Marriage with Incomplete lists and Ties (SMTI), pueden transformarse en un problema NP-Hard.

1 Introducción

El propósito de informe corresponde a la identificación, estudio y desarrollo del problema llamado Stable Marriage Problem, y sus variaciones existentes, tales como el "Stable Marriage with Ties" (SMT), "Stable Marriage with Incomplete Lists" (SMI) y una combinación de los anteriores "Stable Marriage with Ties and Incomplete Lists" (SMTI), el tipo de problema mas ambicioso de resolver. Para ello se presentarán una o algunas definiciones del problema, explicando su funcionamiento y algunas representaciones para las variaciones existentes, se verán distintos algoritmos, técnicas, heurísticas y enfoques utilizados desde la formulación de este problema en 1962 por David Gale y Lloyd Shapley [2], ademas se presentaran los avances mas importantes, que cosas se ha hechos respecto a este problema en la literatura y hacia donde avanza el estudio.

El Stable Marriage Problem, consiste en n hombres y n mujeres, donde cada uno de ellos tiene una lista de preferencia ordenada con los miembros del sexo opuesto, siendo el objetivo crear parejas estables, o sea, que tanto hombre como mujer no deseen divorciarse o separarse [2]. Con el pasar de los años en las distintas variaciones se ha refinado o modificado este problema agregando otras variables, conceptos y funciones que fortalecen el problema, se definen elementos como "Weakly Marriage" (Matrimonio Debil) y "Strong Marriage" (Matrimonio Fuerte), y estos basados en "Blocking Pairs" (Matrimonio donde Hombre y Mujer deseen estar con otra pareja).

SMP tiene muchas aplicaciones en la vida real, puede representar y solucionar distintas situaciones, como cierto tipo de problemas similares y muy relacionados con problemas de asignación, asignaciones Hospital-Residente, problemas de matching, etc. En consecuencia conocer las distintas de representaciones es de gran ayuda para resolver muchas situaciones del ámbito de la Informática, Economía, Matematicas, etc, siendo éste un problema muy interesante de estudiar. En el ámbito de Informática, si bien las versiones SM, SMI y SMT pueden ser resueltas en tiempos polinomiales tanto SMTI y la escala de datos para hacer matching son desafíos interesantes que pueden ser atacados por Algoritmos más ambiciosos.

Este informe está estructurado con tal de obtener un flujo de información sobre el problema Stable Marriage Problem, comenzaremos definiendo el problema y sus variaciones más conocidas, además de ciertas aproximaciones a ellas, comentare del Estado del Arte del problema, entregando ciertos conceptos y resultados obtenido para cada aproximación del problema, en la etapa final compartiré un modelo matemático y finalizando con conclusiones respecto a la información obtenida.

2 Definición del Problema

El problema Stable Marriage, fue definido por David Gale y Lloyd Shapley en 1962 [2], donde m hombres y n mujeres, crean listas de preferencias estrictamente ordenadas con todos los miembros del sexo opuesto, luego se define un *Matching*, correspondiente a un par ordenado hombre-mujer, que define que el hombre **m** y la mujer **w** estan emparejados, ademas se definen los *Blocking Pairs*, que ocurren para una pareja (m_i, w_i), el hombre m_i prefiere a una mujer w_j que a su actual pareja (o Matching), e la mujer w_j prefiere a m_i que a su pareja actual, de esto se define una pareja *estable*, como aquellos Matching que no permiten un *Blocking Pair* . Las variaciones mas conocidas son SM con listas incompletas, SM con Ties(Empates) y una combinación de los anteriores el SM with Incomplete Lists and Ties,

2.1 Stable Marriage with Incomplete Lists

Para este problema, muy parecido al problema original, pero se permite que a cada uno de los hombres y mujeres, pueden tener una lista incompleta de intergrantes del sexo opuesto. Es así como en esta representación, podemos definir que un hombre m es *aceptable* si aparece en la lista de una mujer w, e *inaceptable*, en el de que no aparezca en la lista de preferencias, esto se aplica contrariamente para las mujeres en las listas de preferencias de los hombres. Además cambia el concepto de *estabilidad*, entonces un Matching M en una instancia I de SMI se considera estable si no existe un par (hombre, mujer), tal que para que cada uno:

- Se encuentren sin pareja en M y encuentran a otro *aceptable* ó
- Prefieren a otra persona que a su pareja.

Esto implica para que un *matching estable* no es necesario que sean *matching* completo, esto quiere decir que puede existir un grupo para hombres y mujeres que no se encuentran en pareja dentro del *matching.* Esto implica que todos los *matching estables* para una instancia I de SMI, tienen el mismo tamaño, o sea involucra exactamente la misma cantidad de hombres y mujeres [8]

2.2 Stable Marriage with Ties

Otra variación importante de Stable Marriage Problem, surge cuando algún hombre ó mujer, requiere hacer una lista de preferencias estrictamente ordenada del sexo opuesto, tomando en cuenta que hay ciertos elementos del conjunto de preferencias que son indiferentes entre si, es ási que se definen las *ties* o empates, se define una instancia de SM, como SMT cuando se admiten *ties*, en consecuencia un *Matching* se considera *estable* cuando existe un par (hombre, mujer) que prefieren estar con otro que con su pareja, de estos criterios, se definen diferentes los conceptos de estabilidad:

- *Weak stability* para un Matching M, no existe un par (m,w), donde cada uno de ellos prefiere estrictamente estar con otro que con su pareja en M.
- *Strongly stability* para un matching M, cuando existe un par (m,w), donde el hombre m prefiere estrictamente a la mujer w y w prefiere estrictamente a m.
- *Super Stability*, donde no existe un par (m,w) tal que alguno de ellos prefieran a estar con otra persona o es indeferente entre ellas.[8, 1]

2.3 Incomplete Lists with Ties

Para esta variación del problema debemos tener en cuenta las dos variaciones anteriores, ya que el problema *Stable Marriage Problem with Incomplete List and Ties*, genera una representación de este modelo combinando ambas ideas, las listas incompletas de SMI, y la indiferencia o *ties* del SMT

$m_1 : w_3 w_2$	$w_1 : m_1 m_2$
$m_2 : (w_2, w_1) w_3$	$w_2 : m_1 (m_3 m_2)$
$m_3 : (w_2, w_1)$	$w_3 : (m_1 m_2)$

Table 1: Ejemplo SMTI

Los conceptos de *Weakly stable*, *Strongly Stable* y *Super Stable*, cambian en base a las nuevas condiciones del problema, para un Matching M, se tienen ási los siguientes definiciones de estabilidad:

- *Weakly stable* se define cuando, no existe un par (X,Y), donde cada uno de ellos se encuentra *aceptable* o prefiere estrictamente estar con otro que con su pareja en M.
- *Strongly stable* sucede cuando un par (X,Y) tal que (a) **X** no tiene pareja en M y encuentra a **Y** aceptable o prefiere estrictamente a **Y** que su pareja en M,(b) **Y** no tiene pareja en M y encuentra a **X** aceptable ó prefiere estrictamente **Y** que a su pareja en M ó **Y** es indiferente entre **X** y su pareja en M .
- *Super stable* si no existe una par (x,y) tal que cada uno de ellos no tiene una pareja en M y encuentra al otro aceptable, ó si prefiere estrictamente a su pareja en M ó es indiferente entre ellos.[9]

El problema SMTI es el mas ambicioso, ya que a diferencia de SMT, el largo entro los Matching posibles pueden ser de distintos tamaños, es asi que buscar un Matching Maximo(Un

Matching mas largo)se convierte en un problema NP-Hard. Es asi que algunos de los algoritmos que se hablaran en las siguientes secciones, buscan atacar la cardinalidad de los Matching generados

AGREGAR COMPLEJIDADES

2.4 Otras Variaciones

- **Stable Marriage Problem**: en este problema no se definen dos grupos de personas, si no un solo gran grupo de $2n$ integrantes donde cada uno de ellos crea una lista de preferencias para los $2n-1$ otros integrantes. Dadas estas condiciones este problema es muy similar a problema original de Stable Marriage Problem. [6].

- **Hospital/Resident Problem**: este problema representa asignación de pacientes a cada hosptial de la zona basado en su preferencia y la cuota del hospital. Se generan dos grupos, uno de residentes (R) y otro de hospitales (H) y para cada hospital una cuota asignada (C). Este problema se asemeja mas a el tipo SMI, considerando que un Hospital no se encuentra en el Matching si tiene menos residentes que la cuota.Este problema se puede reducir a un problema SM, definiendo que cada **C** instancias para un Hospital, una para cada posible Residente [6].

3 Estado del Arte

3.1 SM

El problema original *Stable Marriage Problem*, se presento en el año 1962, por David Gale y Lloyd Shapley en un articulo llamado *College Admission and the Stability of Marriage* del journal *American Mathematical Monthly* [2] en esa publicación presentan ciertos algoritmos para encontrar un Matching estable en tiempo $O(n^2)$, donde toma para cada hombre la mejor pareja posible, este caso es *man-optimal* o sea que favorece al hombre esencialmente, aunque se aplique con tal de buscar la mejor pareja para las mujeres, este crearia un resultado *woman-optimal*.

El algoritmo de Gale y Shapley siendo que encontraba matching estables pero poco favorables para alguno de grupos, ya sea los hombres o las mujeres, es por eso que se buscan metodos para obtener matching estables basados en otros aspectos del problema es por eso que que en 1989, Dan Gusfield and Robert W. Irving postulan metodos para encontrar matching estables[4], mayoritariamente basados en los posición de un hombre **m** o una mujer **w**, $p_w(m)$ corresponde a la posición de la mujer **w** para el hombre **m**, de esto se define

$$r(M) = \max_{(m,w)\in M} max(p_m(w), p_w(m)) \tag{1}$$

$$c(M) = \sum_{(m,w)\in M} p_m(w) + \sum_{(m,w)\in M} p_w(m) \tag{2}$$

$$d(M) = \sum_{(m,w)\in M} p_m(w) - \sum_{(m,w)\in M} p_w(m) \tag{3}$$

Para encontrar un *regret stable matching*, *eqalitarian stable matching* y *sex-equal stable matching* se debe encontrar los valore minimos para *regret cost* (1), *egalitarian cost* (2), y *sex-equalness* (3) respectivamente.[4]

3.2 SMI y SMT

En 1994 Robert W. Irving muestra que para un problema SMT se puede encontrar un matching *weakly stable* o *Super stable* en tiempo $O(n^2)$, para aquellos algoritmos que busquen estos tipos de Matching o la no existencia en caso del *Super Stability*, además presenta un algoritmo basado en un extension del algoritmo de Gale y Shapley, que puede obtener una solución *Strongly Stable* si existe una solución en un tiempo de ejecución $O(n^2)$[5]

3.3 SMTI

En 2002 Manlove prueba que MAX SMTI (Maxima Cardinalidad para los Matching en una instancia I de SMTI) es del tipo NP-HARD, incluso para algunos casos muy restringidos con Ties, además muestra que MAX SMTI es NP-completo, cuando las Ties solo existen en un lado (Hombres o Mujeres), ó si solo hay un Tie por lista y el largo de las tie es igual a 2, estos resultados se aplicando cuando se busca la Minima Cardinaliad de los Matching[8].

3.4 Algoritmos

La mayoria de los Algoritmos aca presentados son útiles para atacar el problema de MAX SMTI (Maxima Cardinalidad de una instancia I de SMTI), que como se mencionó algunos son problemas NP-Hard, los algoritmos mas ambiciosos y mas específicos según su definición

3.4.1 RandBrk

Un *Randomized algorithm*, propuesto por Halldórsson en el 2002, definiendo que para un caso, donde solo existen *ties* en la lista de mujeres, donde existe al menos un *tie* de un largo especifico de 2. RandBrk recibe una instancia de un problema SMTI y para un tie de la lista de preferencias de m, se quiebra con igual probabilidad (si existe una lista de w_1 y w_2, ambas tienen una probabilidad de 0.5 de quedar primero), de este resultado se crea una nueva instacia del tipo SMI, y estas se pueden resolver en tiempo polinomial, con algún algoritmo para ello (i.e Gale-Shapley Algorithm)[7].

3.4.2 LocalSearch

Este metodo fue propuesto en el 2010 por Mirco Gelain[3] y básicamente menciona:

- Para una instancia I de SMTI, comenzamos con un Matching M generado de manera Random o por generado por alguna heuristica
- En cada paso siguiente, nos movemos en el vecindario a un nuevo Matching M, el neighborhood de M ($N(M)$), corresponden a todos los Matchings M luego de remover un *Blocking Pair*.
- Luego de movernos en el Neighborhood y remover un *Blockin Pair* (m,w) desde M, las parejas de **m** y **w** (m',w') quedan solteros.
- Nos movemos a un Matching $M' \in N(M)$, tal que $f(M') \leq f(M'')$, siendo $f(M) = nbp(M) + ns(M)$, con $nbp(M)$ el número de blockings pair en M y $ns(M)$ el número de solteros en M.
- Si M es estable, su neighborhood/Vecindario esta vacio y se debe aplicar un restart (Random).

Durante todo el algoritmo, se mantiene las mejoras encontradas, notamos que si no se ha encontrado un Matching M estable, se utiliza el que tiene menor valor de la función de evaluación $f(M)$, y en caso de no encontrar, es aquel con la menor cantidad de solteros.

Para evitar el estancamiento en un mínimo local, hacemos un movimiento random con una probabilidad p, basada en un parámetro del algoritmo. El algoritmo finaliza cuando se crea un Matching perfecto (Matching estable sin solteros), o luego de una cantidad especifica de pasos.

3.4.3 Algoritmo Genetico

Una implementación de un algoritmo genetico para el problema SM consiste en una representación de los cromosomas como una lista (o arreglo) de hombres o mujeres que se encuentran en pareja, por ejemplo, el arreglo [2 , 3, 1], corresponde a la mujer 2 con el hombre 1, la mujer 3 con el hombre 1, etc. Para esta implementación son necesarios el arreglo y las listas de preferencias, estas pueden ser creadas de manera aleatoria o con alguna heurística de selección. A través de las iteraciones las listas de preferencias se mantienen constantes , sólo trabajando con el arreglo.

Los algoritmos genéticos buscan resultados a través de técnicas de cruzamiento y mutación

- El **cruzamiento** se crea tomando dos nodos padres, combinandolos de una manera especifica un nodo hijo, en base a ellos. La manera mas común es eligiendo una posición especifica, quebrar los nodos padres en esa posición y tomando las piezas restantes crear los nodos hijos uniendo las piezas complementarias.De todas maneras este metodo no es posible de utilizar en Stable Marriage Problem, ya que la mayoría de los resultados serían inconsistentes, o valores repetidos para un Matching, de esto, en 1999 Aldershof y Carducci utilizan un metodo llamado *cyclic crossover*[**?**].
  - Cyclic crossover consiste que de cromosoma padre usara un valor x_1 en una posición específica y lo ponemos en la misma posición en el nodo Hijo.
  - Luego, vemos en la misma posición de x_1 en el segundo cromosoma padre, tomamos el valor correspondiente (llamado gen opuesto) y lo ponemos en el nodo Hijo 1, en la posición correspondiente segun el nodo padre.
  - Ahora se elige el valor x_2, pero esta vez según la posición en el primer nodo padre, ahora se elige el gen opuesto (en el segundo padre) y se pone en el nodo Hijo 1, según la posición que se encuentra en padre.
  - Se avanza con el mismo procedimiento hasta que encontremos un número que ya exista en el Nodo Hijo 1, Los valores utilizados se copian al Nodo Hijo 2 (Que hasta ahora esta vació)
  - finalmente los valores incompletos en ambos nodos hijos se rellenan con los valores del segundo nodo Padre
- El metodo de **mutación**, tiene las mismas dificultades que el cruzamiento. Para SM se generan dos funciones de mutación donde se elige una de ellas cada vez que se utilize la mutación. La primera función se hace un swap entre un cierto porcentaje de elementos del vector, ya que al igual que el poblema de cruzamiento cambiar solo un valor implicaria un valor duplicado, y asi mantener la matching aceptable. La otra función que es posible utilizar, es seleccionando todos pares inestables (blocking pairs) y elegir un valor random entre ellos y cambiar ese elemento con otro valor random

4 Modelo Matemático

Un modelo matematico, formulado para programación lineal:[9]

Step	Padres	Hijos
1	Padre 1: [3, **2**, 5, 1, 4] Padre 2: [1, **3**, 4, 2, 5]	Child 1 : [, 2, , ,] Child 2 : []
2	Padre 1: [**3**, 2, 5, 1, 4] Padre 2: [**1**, 3, 4, 2, 5]	Child 1 : [3, 2, , ,] Child 2 : []
3	Parent 1: [3, 2, 5, **1**, 4] Parent 2: [1, 3, 4, **2**, 5]	Child 1 : [3, 2, , 1,] Child 2 : []
4	Parent 1: [3, 2, **5**, 1, **4**] Parent 2: [1, 3, 4, 2, 5]	Child 1 : [3,2, ,1,] Child 2 : [, , 5, , 4]
5	Parent 1: [3, 2, 5, 1, 4] Parent 2: [1, 3, 4, 2, 5]	**Child 1 : [3,2,4,1,5]** **Child 2 : [1,3,5,2,4]**

Table 2: Cyclic Crossover, desde la posición 2[1]

4.1 Variables

- $X_{i,j}$ corresponde al par m_i, w_j si esta un Matching estable.

- A corresponde un grupo de pares aceptados
- Espacio de Busqueda: $2^{(m*w)}$

4.2 Funcion Objetivo

$$max \sum_i \sum_j x_{i,j}$$

4.3 Restricciones

$$\forall w \sum_i x_{i,w} \leq 1$$

$$\forall m \sum_j x_{m,j} \leq 1$$

$$\forall (m,w) \in A : \sum_j x_{m,j} + \sum_i x_{i,w} - x_{m,w} \geq 1$$

$$\forall (m,w) \notin A = 0$$

$$\forall (m,w) \in 0,1$$

5 Descripción del algoritmo

Cómo fue implementando, interesa la implementación más que el algoritmo genérico, es decir, si se tiene que implementar SA, lo que se espera es que se explique en pseudo código la estructura general y en párrafo explicativo cada parte como fue implementada para su caso particular, si se utilizan operadores se debe explicar por que se utilizó ese operador, si fuera el caso de una técnica completa, si se utiliza recursión o no, etc. En este punto no se espera que se incluya código, eso va aparte.

6 Experimentos

Se necesita saber como experimentaron, como definieron parámetros, como los fueron modificando, cuales problemas se trataron, instancias, por que ocuparon esos problemas.

7 Resultados

Que fue lo que se logró con la experimentación, incluir tablas y parámetros, gráficos si fuera posible, lo más explicativo posible.

8 Conclusiones

De acuerdo a la introducción que se hizo, entregar afirmaciones RELEVANTES basadas en los experimentos y sus resultados.

9 Conclusiones

10 Bibliografía

References

[1] Ioannis Damianidis. The stable marriage problem - optimizing different criteria using genetic algorithms. Master's thesis, University of Boras, 2010.

[2] David Gale and Lloyd Shapley. College admission and the stability of marriage. *American Mathematical Monthly*, (69):9–15, 1962.

[3] Mirco Gelain and Maria Silvia Pini. *Local Search for Stable Marriage Problems with Ties and Incomplete Lists.* 2010.

[4] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms.* The MIT Press, 1989.

[5] Robert W. Irving. Stable marriage and indifference. *Discrete Applied Mathematics*, pages 261–272, 1994.

[6] Shuichi Miyazaki Kazuo Iwama. A survey of the stable marriage problem and its variants. *International Conference on Informatics Education and Research for Knowledge-Circulating Socity 2008*, pages 131–136, 2008.

[7] Shuichi Miyazaki Magnús M. Halldórsson, Kazuo Iwama. Randomized approximation of the stable marriage problem. *Theoretical Computer Science*, (3):439–465, 2004.

[8] David Manlove, Robert Irving, and Kazuo Iwama. Hard variants of stable marriage. *Theoretical Computer Science*, pages 261–279, 2002.

[9] Andrej Podhradský. Stable marriage problem algorithms. Master's thesis, Masaryk University, 2010.