Casamentos Estáveis com Casais Proibidos

Guilherme Dias da Fonseca

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto de Matemática da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Informática.

Apresentado por:

____
Guilherme Dias da Fonseca

Aprovado por:

____
Prof. Celina Miraglia Herrera de Figueiredo, D.Sc.
(Presidente)

____
Prof. Jayme Luiz Szwarcfiter, Ph.D.

____
Prof. Vânia Maria Félix Dias, M.Sc.

RIO DE JANEIRO, RJ - BRASIL
JULHO DE 2000

RESUMO

Casamentos Estáveis com Casais Proibidos

Guilherme Dias da Fonseca

Orientadora: Celina Miraglia Herrera de Figueiredo

O problema dos casamentos estáveis consiste em casar n homens com n mulheres de modo a obter um certo tipo de "estabilidade". O algoritmo de Gale-Shapley [GS] encontra um casamento estável particular em tempo $O(n^2)$. Gusfield [G] apresenta algoritmos para encontrar todos os S (número possivelmente exponencial em n) casamentos estáveis e todos os casais estáveis, em tempo $O(n^2+nS)$ e $O(n^2)$, respectivamente. Todos estes algoritmos tem tempo e espaço ótimos.

Uma extensão do problema onde se acrescenta um conjunto de casais forçados e outro de casais proibidos foi proposta e resolvida por Vânia Dias [D]. Apresentamos uma maneira eficiente de converter o problema com casais forçados e proibidos em um problema apenas com casais proibidos. Fornecemos então algoritmos ótimos para, nesta extensão, encontrar: um casamento estável, todos os casamentos estáveis e todos os casais estáveis.

ABSTRACT

Stable Marriages with Forbidden Pairs

Guilherme Dias da Fonseca

Supervisor: Celina Miraglia Herrera de Figueiredo

The stable marriage problem consists in matching n men to n women achieving a certain type of "stability". The Gale-Shapley algorithm [GS] finds a particular stable marriage in $O(n^2)$ time. Gusfield [G] gives algorithms for finding all S (a number possibly exponential in n) stable marriages and all stable pairs in $O(n^2+ns)$ and $O(n^2)$ time, respectively. These algorithms are all optimal.

An extension of the problem, where a set of forced pairs and a set of forbidden pairs are also given, has been proposed and solved by Vânia Dias [D]. We show how to reduce the problem with forced and forbidden pairs to a problem with forbidden pairs only. We give optimal algorithms which find, in this extension: a stable marriage, all stable marriages and all stable pairs.

ÍNDICE

Parte I - Introdução

Primeiro estudaremos a versão original do problema, onde não há conjuntos de casais forçados e proibidos. Homens e mulheres listam as pessoas do sexo oposto em ordem de preferência. Estudaremos o algoritmo de Gale-Shapley para encontrar um casamento estável. Apresentamos então a variação do problema onde temos também conjuntos de casais forçados e proibidos e mostramos uma redução que elimina o conjunto de casais forçados, alterando o conjunto de casais proibidos.

1) O Problema sem Casais Proibidos:

Tomemos n homens e n mulheres. Cada homem ordena as n mulheres em ordem de preferência. Cada mulher faz o mesmo com os homens.

Um **casamento** é um conjunto de n pares (homem,mulher) onde cada homem e cada mulher só aparece uma vez.

Diz-se que um casamento é **instável** se existe um casal *(A,a)* (não pertencente ao casamento) onde *A* prefira *a* à sua esposa e *a* prefira *A* ao seu marido. Neste caso, o casal *(A,a)* **bloqueia** o casamento. Caso contrário o casamentoé **estável**.

Vamos provar que sempre existe pelo menos um casamento estável, quaisquer que sejam as listas de preferências [GS].

2) Encontrando uma Solução:

Vejamos o algoritmo abaixo (algoritmo de Gale-Shapley) [GS] que inicia com todos os homens e mulheres livres:

```
Enquanto existir homem livre
{
  A <- Algum homem livre;
  a <- Primeira mulher na lista de preferência de A;
  Se a prefere A ao seu marido ou a está livre
  {
    A casa com a;
    Retira-se a da lista de preferências do seu ex-marido (se existir);
    Libera-se o ex-marido de a;
    /*O ex-marido de a foi rejeitado por divórcio*/
  }
  senão
  {
    Retira-se a da lista de A; /*A foi rejeitado por recusa*/
  }
}
```

Vejamos um exemplo de execução:

Entrada	
A: a b c d	a: C A B D
B: a b c d	b: C B D A
C: b a d c	c: A B C D
D: b d c a	d: B A C D

Execução	
Casamentos formados	**Iteração**
Aa	A propõe a, aceito
Aa	B propõe a, recusado
Aa Bb	B propõe b, aceito
Aa Cb	C propõe b, aceito, libera B
Aa Cb Bc	B propõe c, aceito
Aa Cb Bc	D propõe b, recusado
Aa Cb Bc Dd	D propõe d, aceito, FIM

Provaremos mais tarde que a solução encontrada é sempre a mesma, independente da ordem em que os homens fazem as propostas.

É fácil ver que o algoritmo sempre termina, que resulta em um casamento e que tem $O(n^2)$ interações. Para isto basta olhar para as listas de preferências dos homens e notar que a cada passo eliminamos uma mulher na lista de preferência do homem que fez a proposta.

Lema 2.1: Uma mulher só se separa se preferir o novo marido ao anterior, portanto a condição das mulheres só melhora ao longo do algoritmo.

Teorema 2.1: O casamento gerado pelo algoritmo é estável.

Prova: Suponha que um casal (A,a) bloqueie o casamento. Isto quer dizer que A prefere a à sua esposa. Então, A propôs a em algum momento. Se a estivesse casada com alguém melhor que A no momento da proposta, isto também seria verdade no final do algoritmo devido ao lema. Neste caso (A,a) não bloquearia o casamento. Se a não estivesse casada com alguém melhor que A no momento da proposta, a aceitaria A. Pelo lema, no fim do algoritmo, a estaria casada com A ou alguém melhor e (A,a) não bloquearia o casamento.

3) Casais Forçados e Proibidos:

Podemos acrescentar à entrada do problema um conjunto de casais forçados F e um conjunto de casais proibidos P. Nesta extensão, um **casamento estável** é um casamento sem bloqueadores que contenha todos os casais de F e não contenha nenhum casal de P. Claramente, podemos não ter casamento estável (por exemplo: considere a entrada onde P contém todos os n^2 casais possíveis). Mostraremos agora que podemos reduzir facilmente este problema com conjuntos F e P a um problema com conjunto F vazio e um novo conjunto P'.

A entrada do problema original era uma dupla (n,L), onde n era o número de homens e L a lista de preferências dos homens e mulheres. Agora temos (n,L,F,P), onde F é o conjunto de casais forçados e P o conjunto de casais proibidos.

Sejam L as listas de preferências dos n homens e das n mulheres, F e P os conjuntos de casais forçados e proibidos, respectivamente, definiremos $T(n,L,F,P)=(n,L,\varnothing,P')$, onde P' é construído da seguinte maneira:

Inicie com $P'=P$. Para cada casal (A,a) de F, acrescente (A,b) a P', para todo $b\neq a$.

Teorema 3.2: Um casamento é estável para $E=(n,L,F,P)$ se e só se ele é estável para $T(E)$.

Prova: Suponha que um casamento seja estável para $T(E)$, mas não para E. Então ele não tem algum casal $(A,a)\in F$. Mas este casamento tem que ter algum casal (A,b), com $b\neq a$. Como todo (A,b) com $b\neq a$ está em P', este casamento é instável para $T(E)$, uma contradição.

Suponha que um casamento seja estável para E, mas não para $T(E)$. Então este casamento tem algum casal (A,b) tal que $(A,b)\in P'$ e $(A,b)\notin P$. Mas (A,b) foi colocado em P' devido a algum casal forçado $(A,a) \in F$. Então o casamento não contém $(A,a) \in F$, não sendo estável para E, uma contradição.

Com esta redução, nos preocuparemos apenas com o problema dos **casamentos estáveis com casais proibidos** ($F = \emptyset$). Apresentaremos algoritmos para encontrar um casamento estável neste problema, todos os casais pertencentes a algum casamento estável e todos os casamentos estáveis. Para isto, precisaremos estudar a fundo a teoria dos casamentos estáveis. Note que, apesar da dificuldade teórica para o problema com casais proibidos ser significativamente maior, a complexidade dos algoritmos obtidos para os 3 problemas apresentados é a mesma dos algoritmos ótimos da versão sem casais forçados nem proibidos. Assim, como estamos trabalhando em uma extensão do problema original, os algoritmos que apresentamos também são ótimos.

Parte II - Espaço de Soluções

Estudaremos propriedades específicas da solução obtida pelo algoritmo de Gale e Shapley. O espaço das soluções será caracterizado como um reticulado distributivo, tanto para a versão original, sem casais proibidos, quanto para a versão com casais proibidos. Veremos também que o número de soluções estáveis pode ser exponencial.

1) Otimalidade da Solução Encontrada:

Até aqui estávamos falando em um homem preferir uma mulher a outra. Vamos estender esta definição a casamentos: Um homem **prefere** um casamento M a outro M' significa que este homem prefere sua esposa em M a sua esposa em M'. Vamos estabelecer uma notação resumida: A prefere a a b: $A(a<b)$. A prefere M a M': $A(M<M')$. Também vamos usar: $a(A<M)$ como a prefere A ao seu marido em M.

Definamos **solução ótima para os homens** como o casamento estável M tal que para todo homem A: $A(M<M')$, qualquer que seja M' estável. Também chamaremos esta solução de $\mathbf{M_0}$.

Em princípio não é óbvio nem mesmo que esta solução exista. Vamos mostrar que esta é a solução encontrada pelo algoritmo. Como a solução ótima para os homens é única por definição, a solução encontrada pelo algoritmo independe da ordem em que os homens propõem. Antes, mais algumas definições: **Casal estável** é qualquer casal que exista em algum casamento estável. **Maridos estáveis** de uma mulher são os homens que formam casais estáveis com esta mulher.

Temos dois tipos de rejeição no algoritmo: divórcio e recusa. Em ambos os casos uma mulher rejeita um homem para se casar com um homem melhor. Dizemos que ***a* recusa *A* por causa de *B***.

<u>Lema 4.2:</u> Ao longo do algoritmo nenhuma mulher rejeita um marido estável.

<u>Prova:</u> Vamos supor que até determinada iteração do algoritmo nenhuma mulher rejeitou um marido estável (inicialmente isto é válido por vacuidade). Suponhamos que a rejeita A, que é um marido estável, por causa de B. Então: $a(B<A)$. Como nenhuma mulher rejeitou um marido estável, a é a melhor esposa estável de B. Assim, qualquer casamento que contivesse (A,a) seria bloqueado por (B,a). Portanto (A,a) é um casal instável, o que é uma contradição.

<u>Teorema 4.3:</u> A solução encontrada pelo algoritmo é ótima para os homens.

<u>Prova:</u> Segue diretamente do lema. Os homens começam a propor a partir das mulheres preferidas. Se um homem é rejeitado, então em nenhum casamento estável ele se casa com a mulher que o rejeitou.

Se invertermos os papéis dos homens e das mulheres no algoritmo, então chegaremos na **solução ótima para as mulheres**.

2) Homens x Mulheres:

Definamos **solução péssima para os homens** como o casamento estável M tal que para todo homem A, $A(M'<M)$, qualquer que seja M' estável. Idem para a **solução péssima para as mulheres**.

<u>Teorema 5.4:</u> A solução ótima para os homens é péssima para as mulheres.

Prova: Seja M o casamento ótimo para os homens, com $(A,a)\in M$ e M' outro casamento estável qualquer. Suponha que $a(M<M')$. Como M é a solução ótima para os homens, $A(M<M')$. Então (A,a) bloqueia M', contrariando a estabilidade de M'.

A prova para o fato da solução ótima para as mulheres ser péssima para os homens é análoga.

Vamos provar uma versão mais forte do Teorema 5.4. Provaremos que o que é melhor para os homens é pior para as mulheres e vice-versa.

Teorema 5.5: Dados dois casamentos M e M', em qualquer casal de M mas não de M', um dos componentes do casal prefere M e o outro prefere M'.

Prova: Vamos dividir os homens e mulheres que têm parceiros diferentes em M e M' em quatro conjuntos disjuntos:

$X = \{A \mid A(M<M')\}$;

$X' = \{A \mid A(M'<M)\}$;

$Y = \{a \mid a(M<M')\}$;

$Y' = \{a \mid a(M'<M)\}$.

Se $(A,a)\in M$ e $A\in X$, então $a\in Y'$, pois se $a(M<M')$, então (A,a) bloquearia M'.

Se $(A,a)\in M$ e $a\in Y$, então $A\in X'$, pois se $A(M<M')$, então (A,a) bloquearia M'.

3) Operadores $\cap$ e $\cup$:

Dados dois casamentos estáveis M e M', vamos definir $M\cap M'$ (pronuncia-se M melhor M') como o conjunto de casais onde cada homem se casa com a preferida dentre suas esposas em M e M'. Não é obvio que $M\cap M'$ seja um casamento estável, ou mesmo que seja um casamento.

Teorema 6.6: Se M e M' são casamentos estáveis, então $M\cap M'$ também é.

Prova:

1) $M\cap M'$ é casamento:

Vamos supor que a tenha dois maridos (A e B) em $M\cap M'$. Digamos que $(A,a)\in M$ e $(B,a)\in M'$. Então $A(M<M')$ e $B(M'<M)$. Aplicando o Teorema 5.5 nos dois casos temos: $a(M'<M)$ e $a(M<M')$, o que é uma contradição.

2) $M\cap M'$ é estável:

Vamos supor que (A,a) bloqueie $M\cap M'$. Como $A(a<M\cap M')$ então $A(a<M)$ e $A(a<M')$. Como $a(A<M\cap M')$, então ou $a(A<M)$ ou $a(A<M')$. No primeiro caso (A,a) bloqueia M e no segundo bloqueia M', o que é absurdo, já que M e M' são estáveis.

Pelo Teorema 5.5, $M\cap M'$ também pode ser definido como cada mulher receber o pior marido dentre M e M'.

Vamos definir $M\cup M'$ (pronuncia-se M pior M') como cada mulher receber o melhor marido dentre M e M' (ou cada homem receber a pior esposa). A prova de que o resultado é um casamento estável é equivalente.

4) Reticulado Distributivo:

Dizemos que M **domina** M' (escrevemos $M \subseteq M'$) se todo homem em M tem uma esposa melhor ou igual a que tinha em M'. Dizemos que M **é dominado por** M' (escrevemos $M \supseteq M'$) se todo homem em M tem uma esposa pior ou igual a que tinha em M'.

Vamos agora mostrar uma outra maneira de armazenar um casamento estável que torna mais coerente o uso dos operadores $\cup$, $\cap$, $\subseteq$ e $\supseteq$.

Chamamos de $P(M)$ o conjunto dos pares $\{(A,a) \mid (A,a) \in M \text{ ou } A(a<M)\}$.

Agora os operadores podem ser aplicados diretamente nos conjuntos $P(M)$. É fácil provar que:

i) $P(M \cup M') = P(M) \cup P(M')$;

ii) $P(M \cap M') = P(M) \cap P(M')$;

iii) $M \subseteq M' \Leftrightarrow P(M) \subseteq P(M')$;

iv) $M \supseteq M' \Leftrightarrow P(M) \supseteq P(M')$.

Com estas propriedades, podemos converter o espaço de soluções de um problema de casamentos estáveis em um anel de conjuntos. Como um anel de conjuntos forma um reticulado distributivo, o conjunto de todos os casamentos estáveis também forma um reticulado distributivo.

Notemos que a solução ótima para os homens é a “interseção” de todas as soluções e a solução ótima para as mulheres a “união”. A primeira domina (está contida em) todas as soluções, enquanto a última é dominada por (contém) todas. Gusfield e Irving [GI] provam vários teoremas de casamentos estáveis no anel de conjuntos equivalente, o que simplifica bastante algumas provas. Aqui, não faremos isto porque consideramos mais importante tornar o leitor familiar com a argumentação e notação de casamentos estáveis.

Todas estas provas funcionam independentemente de haver ou não conjunto de casais proibidos, pois se M e M' não têm casais proibidos, $M \cap M'$ e $M \cup M'$ também não têm.

5) Número de Casamentos Estáveis:

Nós temos interesse em encontrar casamentos estáveis com propriedades específicas (além da otimalidade para homens ou mulheres). Uma possível abordagem seria listar todos os casamentos estáveis e verificar quais satisfazem uma determinada propriedade. Vamos mostrar que esta técnica não é eficiente, já que o número de casamentos estáveis pode crescer exponencialmente em função de n, o número de homens.

Knuth [K] apresentou o exemplo a seguir:

A_1: $a_1, b_1, \ldots$	a_1: $\ldots, A_1$
B_1: $b_1, a_1, \ldots$	b_1: $\ldots, B_1$
A_2: $a_2, b_2, \ldots$	a_2: $\ldots, A_2$
B_2: $b_2, a_2, \ldots$	b_2: $\ldots, B_2$
...	...
A_m: $a_m, b_m, \ldots$	a_m: $\ldots, A_m$
B_m: $b_m, a_m, \ldots$	b_m: $\ldots, B_m$

Como cada par de homens pode casar com sua primeira ou segunda escolha, pode-se obter $2^m = 2^{n/2}$ casamentos estáveis. Note que este não é o número máximo de casamentos estáveis, mas apenas um limite inferior para ele. A determinação do número máximo de casamentos estáveis para uma instância com n homens, assim como a lista de preferências que produz este número são problemas abertos. Vários limites inferiores para este número já foram encontrados.

Parte III - Estruturas Compactas e Algoritmos

Para encontrarmos eficientemente soluções com propriedades específicas, precisamos representar o espaço de soluções de forma compacta (de tamanho polinomial). Estudaremos aqui um algoritmo que lista todos os casais estáveis e veremos que este algoritmo também serve para construir uma representação compactado espaço de soluções.

1) Diferenças Mínimas:

Como o número de casamentos estáveis pode ser exponencial, precisamos de uma estrutura compacta (de tamanho polinomial e que possa ser construída em tempo polinomial) que represente todos os casamentos estáveis. Existem várias estruturas deste tipo, baseadas em casamentos estáveis irredutíveis ou em rotações. Estudaremos a segunda alternativa, que gera os algoritmos mais eficientes conhecidos para vários problemas.

Para os casamentos estáveis distintos M, M' e M'', dizemos que M' é **sucessor imediato** de M quando $M \subseteq M'$ e não existe M'' que satisfaça $M \subseteq M'' \subseteq M'$. Temos interesse em identificar o que muda entre dois casamentos estáveis em que um é sucessor imediato do outro, o que chamaremos de **diferença mínima**. Dizemos que M' é sucessor imediato de M se e só se existe uma diferença mínima que **aplicada** em M gera M'.

Uma primeira idéia é criar um par de conjuntos disjuntos de casais (P,Q) onde $P \subseteq M$ e $M'=(M \backslash P) \cup Q$ é sucessor imediato de M. Os homens (mulheres) que estão em algum par em P também estão em algum par em Q e vice-versa. Caso contrário, $(M \backslash P) \cup Q$ não seria um casamento.

Sabemos também que cada homem e mulher aparece só uma vez em um par de P e só uma vez em um par de Q. Assim, se fizermos o digrafo que tem como vértices os homens de P e como arestas os pares (A,B) onde $(A,b) \in P$ e $(B,b) \in Q$, para alguma mulher b, temos exatamente uma aresta saindo e uma aresta entrando em cada vértice. Portanto, este digrafo ou é um ciclo ou um conjunto de ciclos. De fato, este digrafo é sempre um único ciclo.

Nós vamos definir uma maneira de construir este ciclo, que chamaremos de rotação, e provaremos que uma rotação é uma diferença mínima. Estas diferenças mínimas são especialmente úteis para representar os casamentos estáveis porque apesar do número de casamentos estáveis poder ser polinomial, o número de diferenças mínimas é $O(n^2)$.

2) Rotações:

Vamos chamar de ***S(M,A)*** a primeira mulher na lista de A que prefira estritamente A ao seu marido em M. Vamos chamar de ***Next(M,A)*** o marido de $S(M,A)$ em M. Note que nem sempre existe $S(M,A)$.

Seja $\rho=((A_0,a_0),(A_1,a_1),...,(A_{r-1},a_{r-1}))$ uma lista ordenada de casais de um casamento estável M em que $A_{i+1}=Next(M,A_i)$ e $A_0=Next(M,A_{r-1})$. Chamamos ρ de uma **rotação de ordem *r* exposta em *M***. Digamos que $Next(M,Next(M,A))=A$, neste caso temos uma rotação de ordem 2. A partir daqui os índices nas rotações serão usados módulo r. Falarmos em A_0 ou A_r é a mesma coisa, por exemplo.

Exemplo:

A: a **b** c d - Next:B	a: D B C A
B: b c **d** a - Next:D	b: C A B D
C: c d **a** b - Next:A	c: A D C B
D: d **a** b c - Next:A	d: B D C A

M = {(A,a),(B,b),(C,c),(D,d)}, sublinhado

ρ = (A,a),(B,b),(D,d)

S(M,x) em **negrito**

Lema 10.3: Se $\rho=((A_0,a_0),(A_1,a_1),\ldots,(A_{r-1},a_{r-1}))$ é uma rotação e para algum i, a é uma mulher entre a_i e a_{i+1} na lista de preferência de A_i, então a não é uma esposa estável de A_i.

Prova: Pela definição, a está entre a_i, a esposa de A_i, e $S(M,A_i)$ na lista de A_i. Isto significa que $a(M<A_i)$. Como $a(M<A_i)$ e $A_i(M<a)$, (A_i,a) não é casal estável, pois violaria o Teorema 5.5.

Se ρ é uma rotação exposta em M, chamamos de $\boldsymbol{M/\rho}$ o emparelhamento em que cada homem A_i de ρ se casa com a_{i+1} e os outros homens de M não mudam de parceiras. Dizemos que M/ρ é obtido quando **aplicamos ρ em M**.

Lema 10.4: M/ρ é um casamento estável dominado por M.

Prova: Suponha que (A,a) bloqueie M/ρ. Todas as mulheres tem maridos melhores ou iguais em M/ρ e $a(A<M/\rho)$. Então $a(A<M)$. Então A pertence a ρ, senão (A,a) bloquearia M. Então a está estritamente antes de $S(M,A)$ na lista de A e prefere A a seu marido em M. Isto é uma contradição, pois neste caso $S(M,A)$ teria de ser a.

Lema 10.5: Se M não é a solução ótima para as mulheres, então existe pelo menos uma rotação em M.

Prova: Chamemos de M_z a solução ótima para as mulheres.

i) Se A tem esposas diferentes em M e M_z, então existe $S(M,A)$:

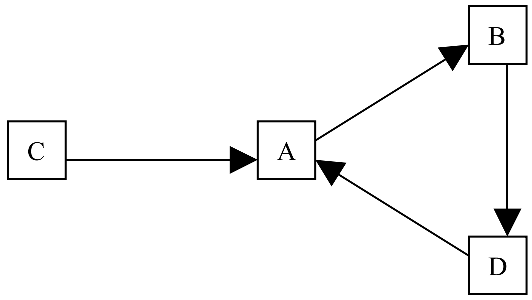
A esposa de A em M_z certamente prefere A a seu marido em M.

ii) Se A tem esposas diferentes em M e M_z, então $B=Next(M,A)$ tem esposas diferentes em M e M_z, portanto existindo $S(M,B)$:

Suponha que $Next(M,A)$ seja casado com $S(M,A)$ (sua esposa em M) em M_z. Então, $S(M,A)(A<M_z)$. Como A não está casado nem com a mesma esposa de M nem com $S(M,A)$, então $A(S(M,A)<M_z)$. Isto contradiz a estabilidade de M_z.

Agora a prova do Lema em si: Seja $\boldsymbol{H(M)}$ o digrafo em que os vértices são os homens com esposas diferentes em M e Mz e a aresta (A,B) existe se e só se $B=Next(M,A)$. Como de cada vértice sai exatamente uma aresta, este grafo tem que conter pelo menos um ciclo (rotação).

Exemplo: $H(M)$ do exemplo anterior:



Esta prova também fornece um algoritmo eficiente para achar uma rotação em um casamento estável M. Basta escolher qualquer homem com esposas diferentes em M e Mz e seguir com a função *Next* até passar duas vezes pelo mesmo vértice. Este algoritmo acha uma rotação em $O(n^2)$.

3) Rotações são Diferenças Mínimas:

Lema 11.6: Se M domina estritamente M' e ρ é exposta em M então ou todos os homens em ρ tem as mesmas parceiras em M e M' ou nenhum deles tem. No último caso, M/ρ domina M'.

Prova: Se $A_i \in \rho$ tem esposas diferentes em M e M' e $(A_i,a) \in M'$, então a ou é $S(M,A_i)$ ou está depois dela na lista de A_i. De qualquer modo, $S(M,A_i)$ não pode estar casada com A_{i+1}, seu marido em M.

Deste lema, junto com o Lema 10.3, segue diretamente:

Teorema 11.7: Para qualquer rotação ρ exposta em M, M/ρ é sucessor imediato de M.

Falta provarmos a outra direção:

Teorema 11.8: Se M' é sucessor imediato de M, então existe rotação ρ exposta em M de modo que $M'=M/\rho$.

Prova: Digamos que A tenha esposas diferentes em M e M'. Se começarmos a procurar uma rotação percorrendo $H(M)$ a partir de A ou acharemos uma rotação ρ que contém A ou acharemos uma rotação ρ que não contém A. No primeiro caso M/ρ será (o único) sucessor imediato de M em que A tem esposa diferente. No segundo caso, para A se casar com $S(M,A)$ ou alguma esposa pior, $Next(M,A)$ tem que fazer o mesmo. Aplicando este argumento sucessivamente, chegamos ao fato de que para piorarmos a esposa de A precisamos aplicar ρ em M. Neste caso M/ρ dominará qualquer casamento que seja dominado por M em que A tenha esposa diferente.

4) Encontrando Todos os Casais Estáveis:

Um problema natural em casamentos estáveis é a determinação de todos os **casais estáveis**, ou seja, todos os casais que estão presentes em algum casamento estável. As rotações nos permitem construir um algoritmo que acha todos os casais estáveis em $O(n^2)$. Este tempo é especialmente bom porque é possível provar que o tempo necessário para determinar se um único casal é estável também é $O(n^2)$.

```
M0 <- Solução ótima para os homens;
Listar todos os casais de M0;
i <- 0;
Enquanto existir rotação ρ exposta em Mi
{
  Mi+1 <- Mi/ρ;
  Listar todos os casais de Mi+1, mas não de Mi;
  i <- i+1;
}
```

O algoritmo acima é claramente $O(n^3)$. Veremos que ele pode ser modificado para rodar em $O(n^2)$. Primeiro vamos mostrar que ele funciona. Claramente todos os casais listados pelo algoritmo são estáveis. Vamos mostrar que todos os casais estáveis são listados pelo algoritmo.

Teorema 12.9: O algoritmo acima lista todos os casais estáveis.

Prova: Suponha que *(A,a)* seja um casal estável não listado. Como o algoritmo parte da solução ótima para os homens e chega na solução ótima para as mulheres, alguma rotação obtida pelo algoritmo levou *A* de uma esposa estritamente melhor que *a* para uma esposa estritamente pior que *a*. Pelo Lema 10.3, *(A,a)* é instável.

Vamos ver agora uma versão deste algoritmo que tem complexidade de tempo em $O(n^2)$:

```
S[A] <- 0, para todo homem A; (Significa A não está na pilha)
M0 <- Solução ótima para os homens;
Mz <- Solução ótima para as mulheres;
Listar todos os casais de M0;
i <- 0;
Enquanto existir homem A com esposas diferentes em Mi e Mz
{
  Empilhe (A);
  S[A] <- Mulher seguinte a esposa de A em Mi na lista de A;
  Enquanto a pilha não estiver vazia
  {
    A <- Homem no topo da pilha;
    Enquanto (S[A] preferir seu marido em Mi em relação a A)
      S[A] <- Mulher seguinte a S[A] na lista de A;
    B <- Marido de S[A]; /*Next(Mi,A)*/
    Se (S[B] = 0)
    {
      Empilhe B;
      S[B] <- Mulher seguinte a esposa de B em Mi na lista de B;
    }
    senão
    {
      Faça
      {
        A <- Homem no topo da pilha;
        Adicione (A,esposa de A em Mi) no início de ρ;
        S[A] <- 0;
        Remova um homem do topo da pilha;
      } enquanto (A≠B);
      Mi+1 <- Mi/ρ;
      Listar todos os casais de Mi+1, mas não de Mi;
      i <- i+1;
      Apague ρ;
    }
  }
}
```

A primeira vista este algoritmo pode parecer complicado, mas na verdade muda muito pouco em relação ao algoritmo anterior. Apenas evitamos cálculos desnecessários ao determinarmos $S(M,A)$ e $Next(M,A)$ para acharmos as rotações. Fazemos isto através da pilha, que obedece a seguinte regra em qualquer ponto do algoritmo: Se A está empilhado sobre B então $Next(M_i,B)=A$. Também utilizamos o fato que se $M_i \subseteq M_j$, então $S(M_j,A)$ não está antes de $S(M_i,A)$ na lista de A.

5) Encontrando Todas as Rotações:

Lema 13.7: Se a rotação $\rho=((A_0,a_0),(A_1,a_1),...,(A_r,a_r))$ está exposta em M e $A_j(M<M')$, onde A_j pertence a um dos casais de ρ, e M' é um casamento estável, então para $i=0..r$, $A_i(M<M')$.

Prova: Pelo Lema 10.3 $A_j(a_{j+1} \leq M')$. Se $(A_{j+1},a_{j+1}) \in M'$, então (A_j,a_{j+1}) bloqueia M' (contradição!). Se $A_{j+1}(M'<a_{j+1})$, então pelo Teorema 5.5 $a_{j+1}(M<M')$. Como $a_{j+1}(A_j<M)$, então $a_{j+1}(A_j<M')$. Assim, (A_j, a_{j+1}) bloqueia M' (contradição!). Então só nos resta que $A_{j+1}(M<M')$. Aplicando este argumento sucessivamente, está provado o lema.

Teorema 13.10: Um casal pertence a no máximo uma rotação.

Prova: Sejam ρ_1 e ρ_2 duas rotações, ρ_1 exposta em M. Vamos supor que $(A,a) \in \rho_1$, $(A,a) \in \rho_2$, $(B,b) \notin \rho_1$ e $(B,b) \in \rho_2$. Obviamente, $A(a<M/\rho_1)$. Pelo Lema 13.7, $B(b<M/\rho_1)$. Então, ao partirmos da solução ótima para os homens aplicando rotações até chegarmos a M/ρ_1, passamos por algum casamento M' (onde $M' \subseteq M$), que expõe uma rotação ρ_3 que contém (B,b). Caso contrário estaríamos contradizendo o Lema 10.3. Claramente, $(A,a) \notin \rho_3$. Analogamente, $B(b<M'/\rho_3)$ e $A(a<M'/\rho_3)$. Isto é absurdo porque $M' \subseteq M$.

Com este teorema, fica claro que os algoritmos apresentados para encontrar todos os casais estáveis também encontram todas as rotações. Basta listarmos cada rotação ρ obtida pelo algoritmo. Notamos também que o número máximo de rotações é $(n^2-n)/2$, já que temos no máximo n^2 casais estáveis, cada casal estável só aparece em no máximo uma rotação, cada rotação tem pelo menos 2 casais estáveis e os n casais estáveis da solução ótima para as mulheres não aparecem em nenhuma rotação.

Como cada casal pertence a no máximo uma rotação, dizemos que esta rotação separa o casal. Dizemos também que uma rotação $((A,a),(B,b),...)$ cria o casal (A,b). Analogamente, temos apenas uma rotação que cria o casal. A prova deste resultado é muito semelhante a do Teorema 13.10. Chamaremos de ***Cria(A,a)*** e ***Separa(A,a)*** estas duas rotações, que criam e separam o casal (A,a), respectivamente.

6) BreakMarriage:

Uma operação bastante útil para elaboração de algoritmos em casamentos estáveis é a operação $BreakMarriage(M,A)$, onde M é um casamento estável e A é um homem. Caso A tenha esposas diferentes em M e M_z, $BreakMarriage(M,A)$ retorna um casamento estável. Caso contrário, definimos que retorna 0. Esta operação foi originalmente definida em [MW] sem usar rotações. Aqui explicaremos esta operação através de rotações:

Em $H(M)$ (digrafo em que os vértices são os homens com esposas diferentes em M e Mz e arestas $(B,Next(M,B))$, para todo vértice B), partindo de A, seguimos a única aresta de saída de cada vértice até encontrarmos um ciclo. Aplicamos então a rotação representada por este ciclo.

Repetimos esta operação até aplicarmos uma rotação que altere a esposa de A, retornando o casamento obtido com a aplicação desta rotação.

Definimos então *BreakMarriage(M,A)* através do algoritmo abaixo, que é uma versão modificada do algoritmo que encontra todas as rotações:

```
BreakMarriage(M,A)
{
  S[A] <- 0, para todo homem A; (Significa A não está na pilha)
  M0 <- M;
  Mz <- Solução ótima para as mulheres;
  Se A tiver a mesma esposa em M e Mz
    Retorne 0;
  i <- 0;
  Empilhe (A);
  S[A] <- Mulher seguinte a esposa de A em Mi na lista de A;
  Enquanto a pilha não estiver vazia
  {
    A <- Homem no topo da pilha;
    Enquanto (S[A] preferir seu marido em Mi em relação a A)
      S[A] <- Mulher seguinte a S[A] na lista de A;
    B <- Marido de S[A]; /*Next(Mi,A)*/
    Se S[B] = 0
    {
      Empilhe B;
      S[B] <- Mulher seguinte a esposa de B em Mi na lista de B;
    }
    senão
    {
      Faça
      {
        A <- Homem no topo da pilha;
        Adicione (A,esposa de A em Mi) no início de ρ;
        S[A] <- 0;
        Remova um homem do topo da pilha;
      } enquanto (A≠B);
      Mi+1 <- Mi/ρ;
      i <- i+1;
      Apague ρ;
    }
  }
  Retorne Mi;
}
```

<u>Teorema 14.11:</u> Seja M um casamento estável e $(A,a)\in M$, *BreakMarriage(M,A)* retorna o melhor ($\cap$) dos casamentos estáveis que são dominados por M e não contêm (A,a), se existir.

<u>Prova:</u> Como *BreakMarriage* parte do casamento estável M aplicando rotações o casamento resultante é estável e dominado por M. Como a pilha começa com A, termina vazia, será aplicada uma rotação que contém (A,a). Então a solução não contém (A,a). Falta provarmos que o algoritmo encontra a melhor solução com estas propriedades. Ao aplicar a rotação *Separa(A,a)*, o algoritmo termina. Precisamos mostrar que todas as rotações que aplicamos antes de separarmos (A,a) realmente precisam ser aplicadas antes da rotação *Separa(A,a)* ser aplicada. Suponha que ρ foi a última rotação aplicada que não precisava ser aplicada. Logo antes de aplicarmos ρ, algum casal de ρ é $(Next(M_i,X),S(M_i,X))$ para um homem X que está na pilha e não pertencente a nenhum casal de ρ. Para aplicarmos a rotação que separa X de sua esposa (e é necessário fazermos isso, já que ρ é a última rotação desnecessária) temos que separar $(Next(M_i,X),S(M_i,X))$. Então é necessário aplicarmos ρ.

7) Encontrando uma Solução com Casais Proibidos:

Usando a operação *BreakMarriage* podemos enunciar o algoritmo que encontra a solução ótima para os homens na versão do problema com casais proibidos:

```
M <- Solução ótima para os homens ignorando casais proibidos;
Enquanto existir casal (A,a) proibido em M
{
  M <- BreakMarriage(M,A);

  Se M = 0
    Retorne "Não há solução";
}
retorne M;
```

Para mostrarmos que o algoritmo funciona vamos provar o seguinte lema:

Lema 15.8: Todo casamento estável M obtido ao longo da execução do algoritmo domina ou é igual a solução estável ótima para os homens (com casais proibidos), se existir.

Prova: Vamos chamar os casamentos M obtidos pelo algoritmo de $M_0,M_1,..,M_k$. M_0 satisfaz a condição por definição. Se M_i satisfaz a condição então, pelo Teorema 14.11, M_{i+1} também satisfaz, já que o homem utilizado no *BreakMarriage* formava um casal proibido em Mi.

Como todo casamento M do algoritmo é dominado pelo M da iteração anterior, fica claro pelo lema que chegaremos ao casamento estável.

Pode não ser claro que a complexidade de tempo do algoritmo é $O(n^2)$. Para isto devemos calcular M_z apenas uma vez no procedimento *BreakMarriage*. Neste caso o algoritmo fica muito semelhante ao algoritmo que encontra todos os casais estáveis, visto anteriormente. Ambos os algoritmos tem complexidade de tempo $O(n^2)$, pois estão sempre "andando" da esquerda para a direita na lista de preferências dos homens, através da aplicação sucessiva de rotações.

8) Digrafo das Rotações:

Seria eficiente se nós pudéssemos determinar uma estrutura que descrevesse de que maneiras podemos aplicar as rotações. Vejamos primeiro a seguinte estrutura, denominada $\Pi(L)$, para a lista de preferências L.

Os vértices do digrafo $\Pi(L)$ são as rotações para a lista L e existe aresta (ρ_1,ρ_2) se e só se ρ_1 precisa necessariamente ser aplicada antes de ρ_2, partindo de M_0. Este digrafo é claramente acíclico e transitivamente fechado, podendo ser considerado uma ordenação parcial das rotações. Chamamos de **subconjuntos fechados** de $\Pi(L)$ os subconjuntos dos vértices de $\Pi(L)$ tais que se ρ_2 pertence ao subconjunto e (ρ_1,ρ_2) é uma aresta de $\Pi(L)$, então ρ_1 também pertence ao subconjunto. A prova do teorema abaixo pode ser encontrada em [GI].

Teorema 16.12: Existe uma correspondência 1 para 1 entre os casamentos estáveis de L e os subconjuntos fechados de $\Pi(L)$.

Seja S um subconjunto fechado de $\Pi(L)$. Para encontrarmos o casamento estável correspondente devemos aplicar todas as rotações de S, partindo de M_0. A única condição na ordem que aplicamos as rotações é que uma rotação tem que estar exposta para ser aplicada, ou seja, temos que aplicar as rotações seguindo a ordenação parcial de $\Pi(L)$.

Normalmente, não é necessário construir explicitamente $\Pi(L)$, mas sim subgrafos de $\Pi(L)$ cujo fecho transitivo seja $\Pi(L)$. Estes subgrafos podem ser encontrados em $O(n^2)$, como veremos.

9) Construindo G(L) e ~G(L):

Definiremos o digrafo $G(L)$ cujo fecho transitivo é $\Pi(L)$. Os vértices de $G(L)$ são as rotações e as arestas podem ser de dois tipos:

Tipo 1: Se a rotação ρ cria o casal (A,a) e a rotação ρ' separa (A,a), então (ρ,ρ') é uma aresta do tipo 1.

Tipo 2: Se a rotação ρ leva uma mulher a de um marido estritamente pior que A para um marido estritamente melhor que A e a rotação ρ' leva A de uma esposa estritamente melhor que a para uma esposa estritamente pior que a, então (ρ,ρ') é uma aresta do tipo 2.

Lema 17.9: Se $(\rho,\rho')\in G(L)$, então $(\rho,\rho')\in \Pi(L)$.

Prova: Se (ρ,ρ') é uma aresta do tipo 1 em $G(L)$, claramente ρ tem que ser aplicada antes de ρ'. Então as arestas do tipo 1 de $G(L)$ pertencem a $\Pi(L)$. Se (ρ,ρ') é uma aresta do tipo 2 em $G(L)$, existe um casal (A,a) tal que ρ leva uma mulher a de um marido estritamente pior que A para um marido estritamente melhor que A e ρ' leva A de uma esposa estritamente melhor que a para uma esposa estritamente pior que a. Este casal (A,a) bloqueia qualquer casamento obtido a partir de M_0 com a aplicação de ρ', mas sem a aplicação de ρ, portanto ρ tem que ser aplicado antes de ρ'.

Lema 17.10: Se ρ é um predecessor imediato de ρ' em $\Pi(L)$, ou seja, $(\rho,\rho')\in \Pi(L)$ e não existe ρ'' tal que (ρ,ρ'') e (ρ'',ρ') pertençam a $\Pi(L)$, então $(\rho,\rho')\in G(L)$.

Prova: Vamos considerar o casamento estável M, correspondente ao subconjunto fechado de $\Pi(L)$ que contém toda rotação r tal que $(r,\rho)\in \Pi(L)$. Neste caso, M/ρ também é um casamento estável. Como ρ é um predecessor imediato de ρ', então $M/\rho/\rho'$ também é um casamento estável, mas M/ρ' não é um casamento estável.

Como $M/\rho/\rho'$ é um casamento estável e M/ρ' não é um casamento estável ocorre uma das seguintes condições: Ou existe um casal criado por ρ que é separado por ρ'; ou então existe um casal (A,a) tal que ρ leva uma mulher a de um marido estritamente pior que A para um marido estritamente melhor que A e ρ' leva A de uma esposa estritamente melhor que a para uma esposa estritamente pior que a. No primeiro caso (ρ,ρ') é uma aresta tipo 1 em $G(L)$ e no segundo caso (ρ,ρ') é uma aresta tipo 2 em $G(L)$.

Destes lemas segue diretamente:

Teorema 17.13: O fecho transitivo de $G(L)$ é $\Pi(L)$.

Podemos construir $G(L)$ em tempo $O(n^2)$ com o algoritmo abaixo. Para implementar este algoritmo é importante colocar que algumas arestas podem ser geradas mais de uma vez. Dependendo da estrutura de dados utilizada, é preciso remover estas arestas repetidas.

```
Para todo casal (A,a)
  L[A,a] <- 0;

Para cada rotação ρ
{
  Para cada (A,a)∈ρ
    L[A,a] <- (1,ρ);

  Para cada (A,a) tais que ρ move a de um marido pior que A para um melhor
    L[A,a] <- (2,ρ);

  Para cada (A,a) pertencente a solução ótima para as mulheres
    L[A,a] <- #;
}

Para todo homem A
{
  r <- 0;
  Para cada mulher a na ordem da lista de preferência de A
  {
    Se L[A,a] = #
      Saia imediatamente deste loop mais interno;

    Se L[A,a] = (1,ρ) (para um ρ qualquer)
    {
      Se r ≠ 0
        Adicione aresta (r,ρ); /* Tipo 1 */
      r <- ρ;
    }
    Senão se L[A,a] = (2,ρ) (para um ρ qualquer)
    {
      Se r ≠ 0
        Adicione aresta (ρ,r); /* Tipo 2 */
    }
  }
}
```

O digrafo *G(L)* já é bastante "esparso", já que o número de vértices é $O(n^2)$ e o número de arestas também. Para alguns algoritmos é necessário uma versão ainda mais "esparsa" de *G(L)*. Esta versão, que chamaremos de *~G(L)* tem ainda a restrição de que todos os vértices tenham grau de saída no máximo *n*, mantendo a condição do fecho transitivo de *~G(L)* ser *Π(L)*. Para encontrarmos *~G(L)*, basta modificarmos o algoritmo de modo que se encontrarmos mais de uma vez uma mesma rotação *ρ* na lista do mesmo homem, só consideraremos a primeira aparição de *ρ* e as aparições marcadas *(1,ρ)*, ou seja, desconsideraremos marcações do tipo *(2,ρ)* que vem depois de marcações do tipo *(1,ρ)* ou *(2,ρ)*.

Teorema 17.14: O fecho transitivo de *~G(L)* é *Π(L)*.

Prova: Basta provarmos que se uma aresta *(ρ,ρ')*, do tipo 2, está em *G(L)* mas não está em *~G(L)*, então existe um caminho de *ρ* para *ρ'* em *~G(L)*. Digamos que *(ρ,ρ')* foi criada quando percorríamos a lista de preferência do homem *A* no algoritmo. Como *(ρ,ρ')* não está em *~G(L)*, então existe uma marcação de *ρ* antes da marcação *(1,ρ')* na lista de *A*. Mas existe aresta *(ρ,ρ")* em *~G(L)*, criada porque *(1,ρ")* está antes de *(1,ρ')* na lista de *A*. Como existe um caminho via arestas do tipo 1 de *ρ"* para *ρ'* em *~G(L)*, então existe caminho de *ρ* para *ρ'* em *~G(L)*.

10) Encontrando Todos os Casamentos Estáveis:

Já sabemos construir *~G(L)* em tempo $O(n^2)$. Mostraremos agora como explorá-lo de modo a encontrar todos os seus subconjuntos fechados, usando o procedimento recursivo *TodosCasamentos(M,L,D)*. *M* é um casamento estável, *L* é uma lista de rotações e *D* é um vetor contendo originalmente o grau de entrada de cada rotação.

```
M0 <- Solução ótima para os homens;
R <- Rotações expostas em M0;
Para toda rotação ρ
  D[ρ] <- Grau de entrada de ρ em ~G(L);
Imprima M0;
TodosCasamentos(M0,R,D);


TodosCasamentos(M,R,D)
{
  Se R não está vazia
  {
    Remova uma rotação ρ de R;
    TodosCasamentos(M,R,D);
    M<-M/ρ;
    Imprima M;
    Para cada rotação π tal que (ρ,π)∈~G(L)
    {
      D[π]=D[π]-1;
      Se D[π]=0
        acrescente π a R;
    }
    TodosCasamentos(M,R,D);
  }
}
```

A prova detalhada deste algoritmo pode ser encontrada em [GI]. Note que, na primeira chamada recursiva, são impressos os casamentos correspondentes aos subconjuntos fechados que não contém ρ. Na segunda chamada recursiva são impressos os casamentos correspondentes aos subconjuntos fechados que contém ρ.

A complexidade do algoritmo é $O(n^2+nS)$, onde *S* é o número de casamentos estáveis. O algoritmo é claramente ótimo, pois $O(n^2)$ é o tempo necessário para achar mesmo uma única solução estável e $O(nS)$ é o tempo necessário para imprimir os *S* casamentos estáveis. É importante frisarmos que só podemos dizer que a complexidade do algoritmo é $O(n^2+nS)$ e não $O(n^2S)$ porque sabemos que o grau de saída dos vértices de *~G(L)* é menor ou igual a *n*. A complexidade de espaço do nosso algoritmo não é ótima, mas uma variação com complexidade de espaço $O(n^2)$ pode ser encontrada em [GI].

11) Encontrando Todos os Casais Estáveis com Casais Proibidos:

Na versão do problema com casais proibidos, além da lista de preferências *L*, temos o conjunto de casais proibidos *P*. Construímos o digrafo *G(L)*, ignorando os casais proibidos e, então, modificamos *G(L)*, apenas adicionando arestas, criando o digrafo ***G(L,P)***. Existe uma correspondência 1 para 1 entre os subconjuntos fechados de *G(L,P)* e os casamentos estáveis de *(L,P)*. O digrafo *G(L,P)* é construído adicionando as seguintes arestas em *G(L)*: Para cada casal proibido *(A,a)*, adiciona-se a aresta *(Separa(A,a),Cria(A,a))*.

Para tratarmos do caso em que não existe *Separa(A,a)* ou *Cria(A,a)*, mas *(A,a)* é um casal estável ignorando *P*, adicionaremos dois vértices especiais V_0 e V_z a *G(L,P)*. Definiremos

$Cria(A,a)=V_0$, se (A,a) pertence a solução ótima para os homens ignorando P e $Separa(A,a)=V_z$, se (A,a) pertence a solução ótima para as mulheres ignorando P. Não precisamos nos preocupar com os casais que já não eram estáveis mesmo sem considerarmos o conjunto P, pois estes continuam não sendo estáveis e podem ser removidos de P sem perdas. Os vértices V_0 e V_z podem ser encarados como rotações vazias e, se fossem adicionados à $G(L)$, formariam a única fonte e o único sumidouro de $G(L)$, respectivamente.

Relembramos a função que define a correspondência entre um subconjunto fechado S de $G(L)$ e um casamento estável M, que é: M é obtido pela aplicação de todas as rotações de S, partindo de M_0. Em $G(L,P)$ usaremos a mesma função, exceto por V_0 que será ignorado, já que é uma rotação vazia. Note que V_0 pertence a todos os subconjuntos fechados não vazios e, por causa disso, só estamos interessados nos subconjuntos fechados não vazios. Os subconjuntos fechados que contém V_z não podem corresponder a casamentos estáveis, pois criam um casal que não é destruído por nenhuma rotação. Assim, estamos interessados nos subconjuntos fechados que não contém V_z. Para evitar repetições, nos referiremos aos subconjuntos fechados de $G(L,P)$ que não são vazios e que não contém V_z apenas como subconjuntos fechados de $G(L,P)$.

Teorema 19.15: Existe uma correspondência 1 para 1 entre os subconjuntos fechados de $G(L,P)$ (que não são vazios e não contém V_z) e os casamentos estáveis de (L,P).

Prova: A função que define a correspondência é a mesma usada em $G(L)$, ignorando o vértice V_0. Como todo subconjunto fechado de $G(L,P)$ é um subconjunto fechado de $G(L)$ mais o vértice V_0, pois apenas adicionamos arestas, então todos os subconjuntos fechados de $G(L,P)$ correspondem a casamentos distintos sem bloqueadores. Precisamos mostrar que todo subconjunto fechado de $G(L,P)$ corresponde a um casamento sem casal proibido e que todo casamento estável (sem casal proibido nem bloqueadores) corresponde a um subconjunto fechado de $G(L,P)$.

Para a primeira afirmação, suponhamos que algum subconjunto fechado S corresponda a um casamento M com o casal proibido (A,a). Neste caso $Cria(A,a)\in S$ e $Separa(A,a)\notin S$. Como $(Separa(A,a),Cria(A,a))$ é uma aresta de $G(L,P)$, então S não é fechado.

Para a segunda afirmação, suponhamos que M seja um casamento estável para o qual não existe subconjunto fechado de $G(L,P)$ correspondente. Existe subconjunto fechado S de $G(L)$ correspondente a M. Neste caso, para alguma rotação $\rho\in S$ e $\rho'\notin S$, (ρ',ρ) é uma aresta de $G(L,P)$, mas não de $G(L)$. Então, para algum casal proibido (A,a), $\rho=Cria(A,a)$ e $\rho'=Separa(A,a)$. Neste caso, M contém o casal proibido (A,a), não sendo estável.

Agora que temos ciclos no digrafo, algumas rotações só podem ser aplicadas em conjunto (as da mesma componente fortemente conexa). Temos interesse então em estender o conceito de aplicar uma rotação para aplicar um conjunto de rotações. Uma **transformação** τ é um conjunto de triplas (A,a,b) que pode ser aplicado em um casamento estável M de modo a obter outro casamento estável M/τ da seguinte maneira: Para cada tripla (A,a,b) de τ, (A,a) deve pertencer a M e (A,b) é um casal de M/τ. Os outros casais de M/τ são os mesmos de M.

Uma transformação define uma permutação das esposas entre os homens do casamento. Uma transformação representa um conjunto de rotações que podem ser aplicadas consecutivamente. Se tivermos apenas uma rotação $((A_0,a_0),(A_1,a_1),\ldots,(A_{n-1},a_{n-1}),(A_n,a_n))$ no conjunto, a transformação correspondente será $\{(A_0,a_0,a_1),(A_1,a_1,a_2),\ldots,(A_{n-1},a_{n-1},a_n),(A_n,a_n,a_0)\}$. Mas, quando temos várias rotações, devemos aplicá-las sucessivamente de modo a construir a transformação correspondente. Dizemos que essas rotações **compõe** a transformação. Para fazermos isto usamos o algoritmo abaixo, que tem como entrada uma lista de rotações R, ordenada de modo que se ρ_1 precede ρ_2 em R, então ρ_1 precede ρ_2 em $G(L)$. Este algoritmo usa um vetor v com n posições, previamente inicializado com 0.

```
Transformação(R)
```

```
{
  Enquanto R não estiver vazia
  {
    Remover rotação ρ=(A0,a0),(A1,a1),...,(Ar,ar) do início de R;
    Para i de 0 até r
    {
      Se v[Ai] = 0
      {
        origem[Ai] <- ai;
        Acrescenta Ai na lista C;
        v[Ai] <- 1;
      }
      destino[Ai] <- ai+1;  /* Módulo r */
    }
  }

  Enquanto C não estiver vazia
  {
    remova A de C;
    acrescente (A,origem[A],destino[A]) a τ;
    v[A]<-0; /* Para restaurar o vetor */
  }
}
```

Uma rotação de $G(L,P)$ só pode ser aplicada se todas as rotações da componente fortemente conexa a qual ela pertence forem aplicadas. Podemos encontrar todas as componentes fortemente conexas de $G(L,P)$ em tempo linear no tamanha do digrafo, ou seja $O(n^2)$. Para cada componente fortemente conexa que não contém V_0 e não contém V_z, encontramos a transformação correspondente. Isto também pode ser feito em $O(n^2)$, com o algoritmo acima. De posse de todas as transformações, todos os casais estáveis são os casais (A,a) tais que (A,a,b) ou (A,b,a) pertença a uma transformação. Equivalentemente, os casais estáveis são os casais (A,a) tais que (A,a,b) pertença a uma transformação ou (A,a) pertença a solução ótima para as mulheres. Ou ainda, os casais estáveis são os casais (A,a) tais que (A,b,a) pertença a uma transformação ou (A,a) pertença a solução ótima para os homens.

12) Encontrando Todos os Casamentos Estáveis com Casais Proibidos:

Dizemos que uma transformação τ **cria** o casal (A,a) se $(A,b,a)\in\tau$, para alguma mulher b. E uma transformação τ **separa** (A,a) se $(A,a,b)\in\tau$, para alguma mulher b.

Para encontrarmos todos os casamentos estáveis com casais proibidos definiremos os digrafos acíclicos $\Omega(L,P)$, $H(L,P)$ e $\sim H(L,P)$, análogos a $\Pi(L)$, $G(L)$ e $\sim G(L)$, respectivamente. Os vértices de destes três digrafos são as transformações correspondentes as componentes fortemente conexas de $G(L,P)$ que não contém V_0 e não contém V_z. O digrafo $\Omega(L,P)$ é transitivamente fechado. Em $\Omega(L,P)$ temos as arestas (τ_1,τ_2) tais que (ρ_1,ρ_2) está em $\Pi(L)$ e ρ_1 compõe τ_1 e ρ_2 compõe τ_2. Do Teorema 19.15 segue diretamente:

<u>Teorema 20.16:</u> Existe uma correspondência 1 para 1 entre os casamentos estáveis de (L,P) e os subconjuntos fechados de $\Omega(L,P)$.

Como $\Pi(L)$, $\Omega(L,P)$ não pode ser construído em tempo $O(n^2)$, por ser muito denso. Para obtermos o digrafo $\sim H(L,P)$ que pode ser construído em tempo $O(n^2)$ e tem todos os vértices com grau de saída menores ou iguais a n, definiremos as arestas de $H(L,P)$ e $\sim H(L,P)$ de maneira análoga a utilizada para $G(L)$ e $\sim G(L)$. Temos então, em $H(L,P)$, dois tipos de aresta:

Tipo 1: Se a transformação τ cria o casal *(A,a)* e a transformação τ' separa *(A,a)*, então (τ, τ') é uma aresta do tipo 1.

Tipo 2: Se a transformação τ leva uma mulher *a* de um marido estritamente pior que *A* para um marido estritamente melhor que *A* e a transformação τ' leva *A* de uma esposa estritamente melhor que *a* para uma esposa estritamente pior que *a*, então (τ, τ') é uma aresta do tipo 2.

O algoritmo para construir *H(L,P)* é muito semelhante ao que constrói *G(L)*. Nos interessamos mais pelo digrafo *~H(L,P)*. Para encontrarmos as arestas de *~H(L,P)*, basta modificarmos o algoritmo de modo que se encontrarmos mais de uma vez uma mesma transformação τ na lista do mesmo homem, só consideraremos a primeira aparição de τ e as aparições marcadas $(1,\tau)$, ou seja, desconsideraremos marcações do tipo $(2,\tau)$ que vem depois de marcações do tipo $(1,\tau)$ ou $(2,\tau)$. O algoritmo é apresentado a seguir:

```
Para todo casal (A,a)
  L[A,a] <- 0;
Para cada transformação τ
{
  Para cada (A,a,_)∈τ
    L[A,a] <- (1,τ);

  Para cada (A,a) tais que τ move a de um marido pior que A para um melhor
    L[A,a] <- (2,τ);

  Mz <- Solução ótima para as mulheres considerando casais proibidos;
  Para cada (A,a)∈Mz
    L[A,a] <- #;

  p[τ] <- 0;
}
Para todo homem A
{
  t <- 0;
  Para cada mulher a na ordem da lista de preferência de A
  {
    Se L[A,a] = #
      Saia imediatamente deste loop mais interno;

    Se L[A,a] = (1,τ) (para um τ qualquer)
    {
      Se t ≠ 0
        Adicione aresta (t,τ)
      t <- τ;
      p[τ] <- 1;
    }
    Senão se L[A,a] = (2,τ) (para um τ qualquer)
    {
      Se t ≠ 0 e p[τ]=0
        Adicione aresta (τ,t)

      p[τ] <- 1;
    }
  }

  Faça p[τ] <- 0 para todo τ usado nesta iteração;
  /* Os outros p[τ] já eram iguais a 0 */
}
```

Lema 20.11: Se $(\tau, \tau') \in H(L,P)$, então $(\tau, \tau') \in \Omega(L,P)$.

Prova: Se (τ, τ') é uma aresta do tipo 1 em $H(L,P)$, claramente τ tem que ser aplicada antes de τ'. Então as arestas do tipo 1 de $H(L,P)$ pertencem a $\Omega(L,P)$. Se (τ, τ') é uma aresta do tipo 2 em $H(L,P)$, existe um casal (A,a) tal que τ leva uma mulher a de um marido estritamente pior que A para um marido estritamente melhor que A e τ' leva A de uma esposa estritamente melhor que a para uma esposa estritamente pior que a. Este casal (A,a) bloqueia qualquer casamento obtido a partir de M_0 com a aplicação de τ', mas sem a aplicação de τ, portanto τ tem que ser aplicado antes de τ'.

Lema 20.12: Se τ é um predecessor imediato de τ' em $\Omega(L,P)$, ou seja, $(\tau, \tau') \in \Omega(L,P)$ e não existe τ'' tal que (τ, τ'') e (τ'', τ') pertençam a $\Omega(L,P)$, então $(\tau, \tau') \in H(L,P)$.

Prova: Vamos considerar o casamento estável M, correspondente ao subconjunto fechado de $\Omega(L,P)$ que contém toda transformação t tal que $(t, \tau) \in \Pi(L)$. Neste caso, M/τ também é um casamento estável. Como τ é um predecessor imediato de τ', então $M/\tau/\tau'$ também é um casamento estável, mas M/τ' não é um casamento estável.

Como $M/\tau/\tau'$ é um casamento estável e M/τ' não é um casamento estável ocorre uma das seguintes condições: Ou existe um casal criado por τ que é separado por τ'; ou então existe um casal (A,a) tal que τ leva uma mulher a de um marido estritamente pior que A para um marido estritamente melhor que A e τ' leva A de uma esposa estritamente melhor que a para uma esposa estritamente pior que a. No primeiro caso (τ, τ') é uma aresta tipo 1 em $H(L,P)$ e no segundo caso (τ, τ') é uma aresta tipo 2 em $H(L,P)$.

Destes lemas segue diretamente:

Teorema 20.17: O fecho transitivo de $H(L,P)$ é $\Omega(L,P)$.

Falta então provarmos:

Teorema 20.18: O fecho transitivo de $\sim H(L,P)$ é $\Omega(L,P)$.

Prova: Basta provarmos que se uma aresta (τ, τ'), do tipo 2, está em $H(L,P)$ mas não está em $\sim H(L,P)$, então existe um caminho de τ para τ' em $\sim H(L,P)$. Digamos que (τ, τ') foi criada quando percorríamos a lista de preferência do homem A no algoritmo. Como (τ, τ') não está em $\sim H(L,P)$, então existe uma marcação de τ antes da marcação $(1, \tau')$ na lista de A. Mas existe aresta (τ, τ'') em $\sim G(L,P)$, criada porque $(1, \tau'')$ está antes de $(1, \tau')$ na lista de A. Como existe um caminho via arestas do tipo 1 de τ'' para τ' em $\sim H(L,P)$, então existe caminho de τ para τ' em $\sim H(L,P)$.

Podemos, então, construir $\sim H(L,P)$ em tempo $O(n^2)$ e explorá-lo em $O(n^2+nS)$ com um algoritmo equivalente ao da sessão (9Assim, encontramos todos os casamentos estáveis com casais proibidos em tempo $O(n^2+nS)$.

Conclusão

Ao acrescentarmos uma lista de casais forçados e outra de casais proibidos, o problema dos casamentos estáveis tornou-se bem mais complicado. A lista de casais forçados foi facilmente eliminada, mas, para resolvermos o problema com casais proibidos, precisamos estudar a fundo a teoria dos casamentos estáveis e seu espaço de soluções.

Na versão com casais proibidos, estudamos 3 problemas clássicos: encontrar um casamento estável (no caso, o ótimo para os homens), encontrar todos os casais estáveis e encontrar todos os casamentos estáveis. Destes problemas, o de encontrar um casamento estável e o de encontrar todos os casamentos estáveis já haviam sido estudados por Vânia Dias [D], que obteve algoritmos de complexidade $O(n^4)$ e $O(n^4+n^2S)$, respectivamente. Vânia Dias trabalhava diretamente com os casais forçados e com os casais proibidos.

Quando eu assistia a defesa da tese de Vânia Dias, achei que talvez fosse possível melhorar um pouco a complexidade destes algoritmos. Pensava em obter um algoritmo $O(n^4+nS)$ para encontrar todos os casamentos estáveis. O problema se mostrou mais complicado do que eu pensava, mas conseguimos um algoritmo $O(n^2)$ para encontrar um casamento estável. Depois chegamos no algoritmo $O(n^4+nS)$ e, finalmente, $O(n^2+nS)$, para encontrar todos os casamentos estáveis.

Foi surpreendente termos obtido algoritmos que resolvem os 3 problemas na mesma complexidade de tempo da versão original, onde não havia casais proibidos. Como a versão com casais proibidos é uma extensão da versão original e os algoritmos da versão original já foram provados ótimos, os algoritmos aqui fornecidos também são ótimos.

Bibliografia

[GI] Dan Gusfield, Robert W. Irving
The Stable Marriage Problem - Structure and Algorithms
1989, The MIT Press

[K] Donald E. Knuth
Mariages Stables
1976, Les Presses de l'Université de Montréal

[G] Dan Gusfield
Three Fast Algorithms for Four Problems in Stable Marriage
1987, SIAM Journal on Computing, 16:111-128

[IL] Robert W. Irving, Paul Leather
The Complexity of Counting Stable Marriages
1986, SIAM Journal on Computing, 15:655-667

[GS] David Gale e Lloyd S. Shapley
College Admissions and the Stability of Marriage
1962, American Mathematical Monthly, 69:9-15

[D] Vânia Maria Félix Dias
O Problema dos Casamentos Estáveis com Restrições de Pares
2000, Tese de Mestrado - COPPE - UFRJ

[MW] D.G. McVitie e L.B. Wilson
The Stable Marriage Problem
1971, Communications of the A.C.M., 14:486-490