Problemas de Emparelhamento (*matching*)

Ana Paula Tomás

LEIC - Desenho de Algoritmos
Universidade do Porto

Abril 2022

1. Outras Aplicações de Fluxo Máximo
   - Emparelhamentos em grafos bipartidos

2. O problema dos casamentos estáveis e variantes

Emparelhamento de cardinal máximo em grafos bipartidos

Grafos bipartidos e Emparelhamentos (*Matchings*)

Um grafo não dirigido $G = (V_1 \cup V_2, E)$ é bipartido sse $u \in V_1$ e $v \in V_2$, qualquer que seja o ramo $\langle u, v \rangle \in E$.

Um emparelhamento num grafo $G = (V, E)$ é um subconjunto M de E tal que quaisquer ramos em M são incidentes em vértices distintos de V (ou seja, não há dois ramos em M que partilhem algum extremo).

Problema: *Maximal Bipartite Matching Problem*

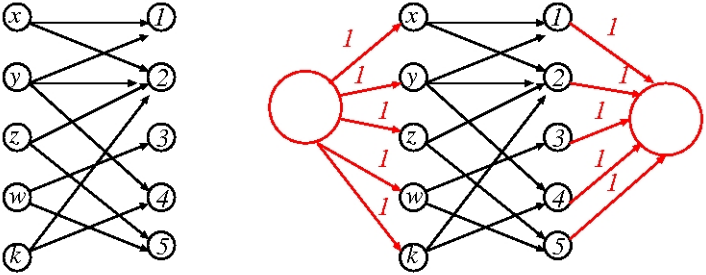
Dado um grafo bipartido $G = (V_1 \cup V_2, E)$ determinar um emparelhamento de cardinal máximo em G.

Exemplos de aplicação:

- **Atribuição de tarefas a pessoas**. Cada pessoa só desempenha no máximo uma tarefa e cada tarefa é atribuída no máximo a uma pessoa. As pessoas podem ter habilitações distintas. Maximizar o número de tarefas realizadas.
- **Propagação de restrições** em sistemas de programação por restrições (*constraint programming*): sendo $x_1, \ldots, x_n$ variáveis com $x_i \in D_i$, e D_i finito, para $1 \leq i \leq n$, existirá uma atribuição de valores às variáveis que satisfaça $x_i \neq x_j$ se $i \neq j$, para todo i, j?

Emparelhamento de cardinal máximo em grafos bipartidos

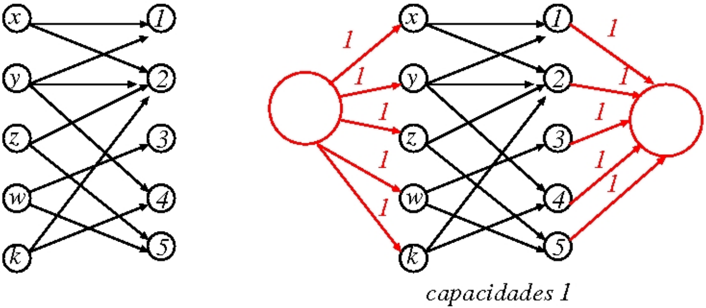
Pode-se reduzir instâncias de *"maximum matching"* a fluxo máximo.
Basta orientar os ramos, inserir origem s e destino t fictícios, acrescentar ramos de s para os nós de V_1 e dos nós de V_2 para t, e atribuir capacidades unitárias.



capacidades 1

Emparelhamento de cardinal máximo em grafos bipartidos

Pode-se reduzir instâncias de *"maximum matching"* a fluxo máximo.
Basta orientar os ramos, inserir origem s e destino t fictícios, acrescentar ramos de s para os nós de V_1 e dos nós de V_2 para t, e atribuir capacidades unitárias.

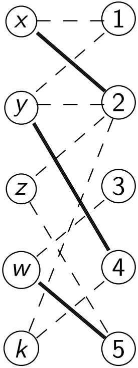


Um fluxo máximo na rede corresponde a um emparelhamento de cardinal máximo.

NB: **Complexidade** $O(mn)$, com $n = |V_1 \cup V_2|$. O número de iterações $\leq \min(|V_1|, |V_2|) \leq n/2$, pois emparelha mais 2 nós em cada iteração. Existem algoritmos específicos, por exemplo, o Algoritmo de Hopcroft-Karp com complexidade $O(m\sqrt{n})$.

Exemplo

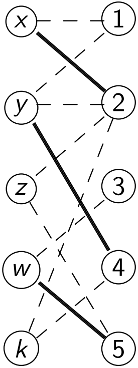
Considerar o emparelhamento $M = \{\langle x, 2\rangle, \langle y, 4\rangle, \langle w, 5\rangle\}$ no grafo bipartido representado (os restantes ramos estão a tracejado).



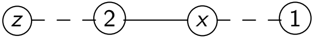
Os nós z, k, 3 e 1 estão **expostos** em M, i.e., não emparelhados. É um emparelhamento de cardinal máximo?

Exemplo

Considerar o emparelhamento $M = \{\langle x, 2\rangle, \langle y, 4\rangle, \langle w, 5\rangle\}$ no grafo bipartido representado (os restantes ramos estão a tracejado).



Os nós z, k, 3 e 1 estão **expostos** em M, i.e., não emparelhados. É um emparelhamento de cardinal máximo? Não, porque



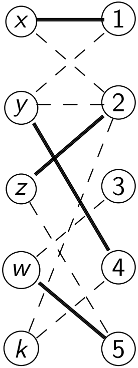
determina um caminho para aumento que, em *matching*, se designa por **caminho alternado para aumento**, por alternar entre ramos que não estão em M e que estão em M.

Na rede residual para M, os ramos a tracejado têm capacidade 1 e estão orientados da esquerda para a direita; os que estavam em M têm capacidade 1 e ficam orientados da direita para a esquerda. Após o aumento, o novo emparelhamento M' é

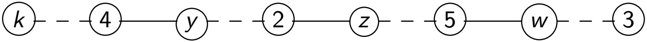
$$M' = M \oplus \{\langle z, 2\rangle, \langle x, 2\rangle, \langle x, 1\rangle\} = \{\langle x, 1\rangle, \langle z, 2\rangle, \langle y, 4\rangle, \langle w, 5\rangle\}$$

Exemplo (cont.)

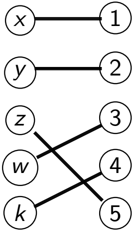
Será $M' = \{\langle x,1\rangle, \langle z,2\rangle, \langle y,4\rangle, \langle w,5\rangle\}$ máximo?



Os nós k e 3 estão expostos em M' e M' não é máximo porque, por BFS a partir de k, podemos encontrar o seguinte caminho para aumento de M':



Se retirarmos os ramos $\langle z,2\rangle$, $\langle y,4\rangle$ e $\langle w,5\rangle$ e acrescentarmos $\langle k,4\rangle$, $\langle y,2\rangle$, $\langle z,5\rangle$ e $\langle w,3\rangle$ obtemos um **emparelhamento perfeito**.



Outros exemplos de aplicações

Problemas que se resolvem por redução ao problema de fluxo máximo:

- Qual é o **número máximo de caminhos que não partilham ramos** num grafo dirigido G de um nó s para um nó t, sendo G, s e t dados?
- Qual é o **número mínimo de ramos que desconectam s de t** em G?
- **Atribuição de tarefas a pessoas**: cada pessoa tem competências para algumas tarefas e uma capacidade máxima. Maximizar o número de tarefas atribuídas. Cada tarefa só será desempenhada por uma pessoa no máximo.
- **Colocações de pessoas em postos de trabalho** (sem preferências): maximizar o número de pessoas colocadas, dadas as vagas em cada posto e as habilitações de cada pessoa (i.e., os postos que pode ocupar).

Como são as redes de fluxo? Origem e destino? Capacidades dos ramos?

Outras aplicações: https://www.cs.princeton.edu/courses/archive/spring13/cos423/lectures/07NetworkFlowII.pdf

---

A seguir, introduzimos um problema clássico de afetação com preferências mútuas.

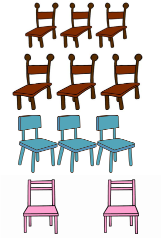
1. Outras Aplicações de Fluxo Máximo
   - Emparelhamentos em grafos bipartidos

2. O problema dos casamentos estáveis e variantes

Amores e Algoritmos

Sentar ou não sentar?

São de ideias quase fixas os habitantes do reino da Teimosia, preferindo ficar de pé do que sentar-se em cadeiras pelas quais não nutrem especial empatia. Só já muito exaustos se sentam nalguma que, não sendo das suas prediletas, também não rejeitam totalmente. Respeitam uma **hierarquia estrita**, *aguardando que os seus superiores se sentem ou insistam em ficar de pé*, para avançar então para um lugar da sua preferência, ainda desocupado. Onde se senta cada um?



Qualquer semelhança com o **problema de colocação de candidatos aos ensino superior** é pura coincidência!

Sentar ou não sentar?

- Candidatos estritamente ordenados.
- Lista de preferências única para todas as instituições. Coincide com a seriação dos candidatos.
- **Ideia do Algoritmo para colocação:**
  - Seguir a lista de seriação até ter analisado todos os candidatos.
  - Para colocar o próximo candidato, analisar as suas preferências por ordem decrescente de preferência, e colocar o candidato na primeira opção que ainda tem vagas.

O problema dos casamentos estáveis e variantes

A versão clássica do problema traduz-se da forma seguinte.

> STABLEMARRIAGE: *Seja $\mathcal{H} = \{h_1, \ldots, h_n\}$ um conjunto de n homens e seja $\mathcal{M} = \{m_1, \ldots, m_n\}$ um conjunto de n mulheres. Cada elemento ordenou todos os de sexo oposto por ordem de preferência estrita. Determinar um emparelhamento estável M, com $|M| = n$.*

The stable marriage problem (Casamentos estáveis)

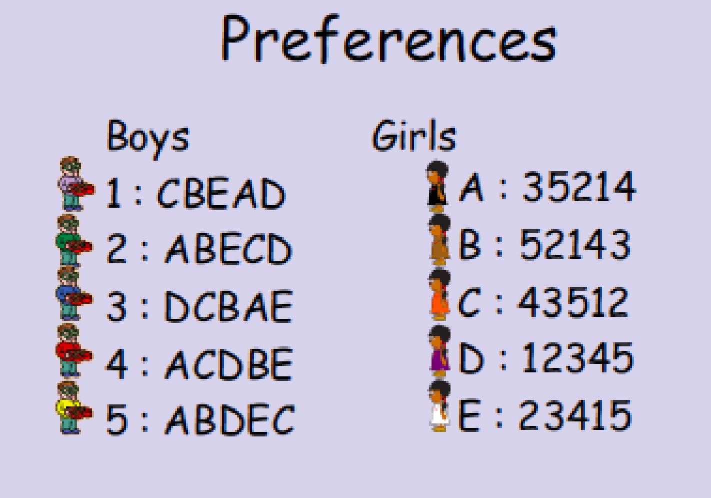
Preferences

Boys	Girls
1 : CBEAD	A : 35214
2 : ABECD	B : 52143
3 : DCBAE	C : 43512
4 : ACDBE	D : 12345
5 : ABDEC	E : 23415

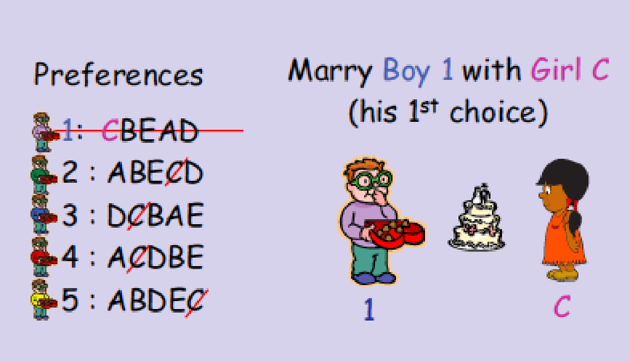
Figuras adaptadas de MIT 6.042J/18.062J, MIT OpenCourseWare, de Albert R. Meyer, 2013

`https://www.youtube.com/watch?v=RE5PmdGNgj0`

The stable marriage problem (Casamentos estáveis)



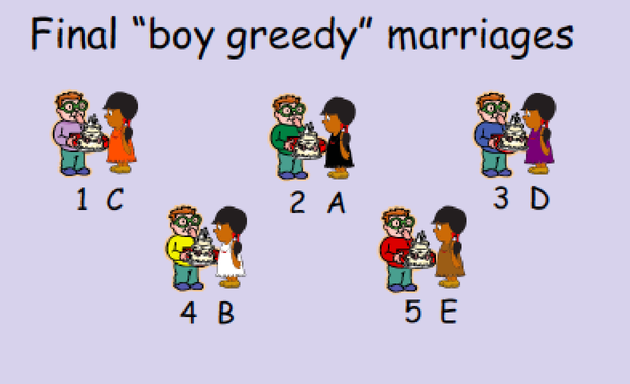
Abordagem *greedy* (gulosa)



The stable marriage problem (Casamentos estáveis)

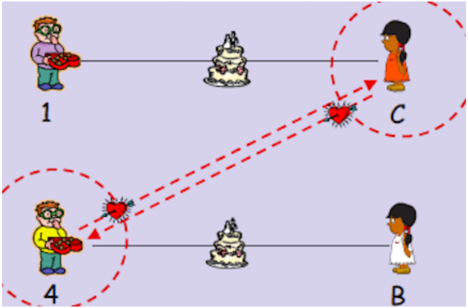
Abordagem *greedy* (gulosa)

Abordagem *greedy* (gulosa)



O que é **instabilidade?**

O par $(4, C)$ cria **instabilidade** porque 4 prefere C (a B) e C prefere 4 (a 1).



Não queremos **divórcio**. Como obter **um conjunto de casais estável**?

Greedy não resolve corretamente o problema!...

STABLEMARRIAGE: *Seja $\mathcal{H} = \{h_1, \ldots, h_n\}$ um conjunto de n homens e seja $\mathcal{M} = \{m_1, \ldots, m_n\}$ um conjunto de n mulheres. Cada elemento ordenou todos os de sexo oposto por ordem de preferência estrita. Determinar um* **emparelhamento estável** *M, com $|M| = n$.*

M é instável sse existir um par $(h, m) \notin M$ tal que h prefere m a $M(h)$ e m prefere h a $M(m)$, sendo $M(x)$ o par de x em M.

O **Algoritmo de Gale-Shapley** (1962) obtém **um emparelhamento estável**.

Algoritmo de Gale-Shapley para StableMarriage

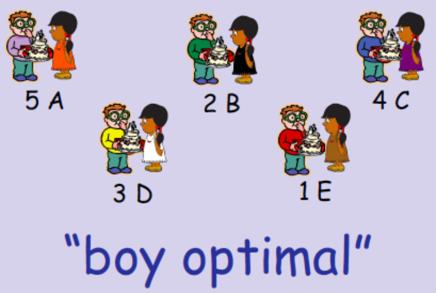
Gale e Shapley (1962) provaram que qualquer instância de StableMarriage admite um emparelhamento estável.

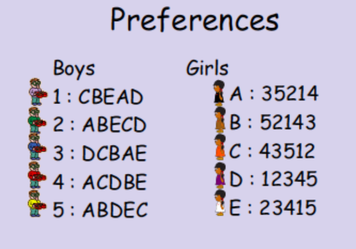
Algoritmo de Gale-Shapley

Considerar inicialmente que todas as pessoas estão livres.
Enquanto houver algum homem h livre fazer:
    seja m a primeira mulher na lista de h a quem este ainda não se propôs;
    se m estiver livre então
        emparelhar h e m (ficam noivos)
    senão
        se m preferir h ao seu noivo atual h' então
            emparelhar h e m (ficam noivos), voltando h' a estar livre
        senão
            m rejeita h e assim h continua livre.

O algoritmo de Gale-Shapley obtém um emparelhamento estável em **tempo $O(n^2)$**.
Foi provado que tal emparelhamento é **o melhor emparelhamento estável segundo os homens** e, se não for o único emparelhamento estável, é **o pior segundo as mulheres**.

The stable marriage problem (Casamentos estáveis)





Exemplo:

$$\begin{array}{lllll} h_1: & m_4, & m_2, & m_3, & m_1 \\ h_2: & m_2, & m_3, & m_4, & m_1 \\ h_3: & m_2, & m_3, & m_1, & m_4 \\ h_4: & m_1, & m_3, & m_2, & m_4 \end{array} \qquad \begin{array}{lllll} m_1: & h_4, & h_2, & h_1, & h_3 \\ m_2: & h_3, & h_1, & h_4, & h_2 \\ m_3: & h_2, & h_3, & h_1, & h_4 \\ m_4: & h_3, & h_4, & h_2, & h_1 \end{array}$$

$M = \{(h_1, m_4), (h_2, m_3), (h_3, m_2), (h_4, m_1)\}$ é o emparelhamento estável que se obtém por aplicação do algoritmo de Gale-Shapley a esta instância.

Não existe nenhum outro emparelhamento estável para esta instância.

Exemplo:

$$\begin{array}{llllll} h_1: & m_4, & m_2, & m_3, & m_1 \\ h_2: & m_2, & m_3, & m_4, & m_1 \\ h_3: & m_2, & m_3, & m_1, & m_4 \\ h_4: & m_1, & m_3, & m_2, & m_4 \end{array} \qquad \begin{array}{llllll} m_1: & h_4, & h_2, & h_1, & h_3 \\ m_2: & h_3, & h_1, & h_4, & h_2 \\ m_3: & h_2, & h_3, & h_1, & h_4 \\ m_4: & h_3, & h_4, & h_2, & h_1 \end{array}$$

$M = \{(h_1, m_4), (h_2, m_3), (h_3, m_2), (h_4, m_1)\}$ é o emparelhamento estável que se obtém por aplicação do algoritmo de Gale-Shapley a esta instância.

Não existe nenhum outro emparelhamento estável para esta instância.

Justificação: De acordo com a teoria sobre o problema StableMarriage, o emparelhamento que o algoritmo de Gale-Shapley produz é ótimo segundo os homens e péssimo segundo as mulheres. Se se **trocar o papel dos homens e das mulheres no algoritmo de Gale-Shapley**, obtém-se o emparelhamento ótimo segundo as mulheres e péssimo segundo os homens. Para esta instância esses dois emparelhamentos são iguais, o que quer dizer que M é o único emparelhamento estável que esta instância admite.

Outros casamentos

Candidatos e empresas

Colocar os **candidatos** c_1, c_2, c_3, e c_4 em quatro **postos de trabalho** p_1, p_2, p_3, p_4, numa empresa, dadas as preferências dos candidatos e da empresa (ambas por ordem decrescente). Cada posto é atribuído exatamente a um candidato.

$$\begin{array}{llllll} c_1: & p_1, & p_2, & p_3, & p_4 \\ c_2: & p_2, & p_3, & p_1, & p_4 \\ c_3: & p_3, & p_4, & p_2, & p_1 \\ c_4: & p_4, & p_1, & p_2, & p_3 \end{array} \qquad \begin{array}{llllll} p_1: & c_2, & c_3, & c_4, & c_1 \\ p_2: & c_3, & c_4, & c_1, & c_2 \\ p_3: & c_1, & c_4, & c_2, & c_3 \\ p_4: & c_2, & c_1, & c_3, & c_4 \end{array}$$

Qual é a melhor solução estável segundo os candidatos?

$$M^C = \{(c_1, p_1), (c_2, p_2), (c_3, p_3), (c_4, p_4)\}$$

Outros casamentos

Candidatos e empresas

Colocar os **candidatos** c_1, c_2, c_3, e c_4 em quatro **postos de trabalho** p_1, p_2, p_3, p_4, numa empresa, dadas as preferências dos candidatos e da empresa (ambas por ordem decrescente). Cada posto é atribuído exatamente a um candidato.

$$\begin{array}{llllll} c_1: & p_1, & p_2, & p_3, & p_4 \\ c_2: & p_2, & p_3, & p_1, & p_4 \\ c_3: & p_3, & p_4, & p_2, & p_1 \\ c_4: & p_4, & p_1, & p_2, & p_3 \end{array} \qquad \begin{array}{llllll} p_1: & c_2, & c_3, & c_4, & c_1 \\ p_2: & c_3, & c_4, & c_1, & c_2 \\ p_3: & c_1, & c_4, & c_2, & c_3 \\ p_4: & c_2, & c_1, & c_3, & c_4 \end{array}$$

Qual é a melhor solução estável segundo os candidatos?

$$M^C = \{(c_1, p_1), (c_2, p_2), (c_3, p_3), (c_4, p_4)\}$$

Mas, segundo a empresa, isto é, segundo os postos de trabalho, seria:

$$M^E = \{(c_1, p_3), (c_2, p_1), (c_3, p_4), (c_4, p_2)\}$$

Conclusão: M^C é o pior emparelhamento estável para a empresa e M^E é o pior emparelhamento estável para os candidatos.

Qual preferir? Existirá um mais "igualitário"?

The stable marriage problem (Casamentos estáveis)

O problema dos casamentos estáveis é um problema de emparelhamento em grafos bipartidos com preferências mútuas.
Foram estudadas diversas variantes com interesse prático.

- Para saber mais, consultar, por exemplo:
  - D. Gusfield and R. W. Irving. The stable marriage problem - structure and algorithms. MIT Press, Cambridge (1989);
  - D. Manlove. Algorithms of matchings under preferences. World Scientific (2013).

- **Prémio Nobel 2012 (Economia)** atribuído a A.Roth e L.Shapley *"for the theory of stable allocations and the practice of market design"*.

  (D. Gale tinha falecido em 2008)

Estrutura das soluções do (*"stable marriage problem"*)

Para saber mais (se tiver interesse) sobre o significado de:
"o emparelhamento que o ALGORITMO DE GALE-SHAPLEY *produz é ótimo para os homens* e *péssimo para as mulheres*".

- Isto quer dizer que, quando considerados todos **os emparelhamentos estáveis**, qualquer homem fica com a melhor companheira que poderia obter e qualquer mulher fica com o pior companheiro que poderia obter.

- Formalmente, o **conjunto** $\mathbb{M}$ **de todos os emparelhamentos estáveis** de uma instância de STABLEMARRIAGE constitui um **reticulado distributivo** se for ordenado por $\preceq$ assim: $M \preceq M'$, lendo-se M **domina** M' **(segundo os homens)**, sse todo homem tem ou a mesma companheira em M e M' ou uma companheira em M que prefere à que tem em M'.

- Prova-se que: M domina M' segundo os homens sse M' domina M segundo as mulheres.

Estrutura das soluções do (*"stable marriage problem"*)

Para saber mais (se tiver interesse):
Porque é que $(\mathbb{M}, \preceq)$ é um reticulado distributivo?

- Sendo M e M' emparelhamentos estáveis, tem-se:
  - é estável o emparelhamento $M \wedge M'$, em que cada homem h ficará com a mulher que **prefere** entre $M(h)$ e $M'(h)$;
  - é estável o emparelhamento $M \vee M'$, em que cada homem h ficará com a mulher de que **gosta menos** entre $M(h)$ e $M'(h)$.
- $(\mathbb{M}, \preceq)$ é um reticulado: $M \wedge M'$ é o **ínfimo** entre M e M'; $M \vee M'$ é o **supremo** entre M e M'.

  Distributivo porque $M_1 \vee (M_2 \wedge M_3) = (M_1 \vee M_2) \wedge (M_1 \vee M_3)$ e $M_1 \wedge (M_2 \vee M_3) = (M_1 \wedge M_2) \vee (M_1 \wedge M_3)$, para $M_1, M_2, M_3 \in \mathbb{M}$.
- O mínimo de $\mathbb{M}$ é o emparelhamento estável ótimo segundo os homens.
- O máximo de $\mathbb{M}$, se for distinto do mínimo, é o pior emparelhamento estável segundo os homens (mas o melhor para as mulheres).

Estrutura das soluções do (*"stable marriage problem"*)

Para saber mais (se tiver interesse):
A estrutura de reticulado permite obter alguns algoritmos eficientes.

Por exemplo, em D. Gusfield, Three Fast Algorithms for four Problems in Stable Marriage, *SIAM J. Computing*, 16(1):111-128, 1987. é explorada para:

- determinar de todos os **pares** estáveis em $O(n^2)$;
- enumerar com complexidade ótima (espaço-temporal) todos os emparelhamentos estáveis. Tempo $O(n^2 + n|\mathbb{M}|)$ e espaço $O(n^2)$;
  - Notar que $|\mathbb{M}|$ pode ser exponencial em n.
    R. Irving, P. Leather, The Complexity of Counting Stable Marriages, *SIAM J. Computing*, 15:655-667, 1986.
- construir em $O(n^2)$ o emparelhamento estável que **minimiza o nível de descontentamento máximo**, quando se considera simultaneamente todos os homens e mulheres.

Preferências incompletas mas estritamente ordenadas

StableMarriageWithIncompleteLists (SMI): as listas de preferências estão ordenadas estritamente mas pode haver pares inaceitáveis, ou seja, a lista de preferências de cada elemento pode ser um subconjunto próprio da lista de elementos do sexo oposto.

Preferências incompletas mas estritamente ordenadas

StableMarriageWithIncompleteLists (SMI): as listas de preferências estão ordenadas estritamente mas pode haver pares inaceitáveis, ou seja, a lista de preferências de cada elemento pode ser um subconjunto próprio da lista de elementos do sexo oposto.

Prova-se que:

Os elementos que ficam com par em algum emparelhamento estável, ficam com par em **todos** os emparelhamentos estáveis.

Preferências incompletas mas estritamente ordenadas

STABLEMARRIAGEWITHINCOMPLETELISTS (SMI): as listas de preferências estão ordenadas estritamente mas pode haver pares inaceitáveis, ou seja, a lista de preferências de cada elemento pode ser um subconjunto próprio da lista de elementos do sexo oposto.

Prova-se que:

Os elementos que ficam com par em algum emparelhamento estável, ficam com par em **todos** os emparelhamentos estáveis.

Este resultado é interessante porque, por exemplo, se a instância de SMI traduzir um **problema de colocação de candidatos em postos de trabalho** com preferências mútuas, e em que cada candidato só pode ficar num posto e cada posto só pode ser atribuído a um candidato, sabemos que independentemente do algoritmo aplicado para obter um emparelhamento estável, os candidatos que ficam colocados são sempre os mesmos e os postos que ficam livres também.

- Afetação do tipo **um-para-vários**: cada universidade (hospital) pode ter **várias vagas**, mas cada uma será preenchida no máximo por um candidato;
- Preferências possivelmente incompletas, mas estritamente ordenadas;
- **Extensão do Algoritmo de Gale-Shapley [1962]** para **colocar candidatos em universidades** (USA):

Colocar Alunos - Universidades e Internos - Hospitais

- Afetação do tipo **um-para-vários**: cada universidade (hospital) pode ter **várias vagas**, mas cada uma será preenchida no máximo por um candidato;
- Preferências possivelmente incompletas, mas estritamente ordenadas;
- **Extensão do Algoritmo de Gale-Shapley [1962]** para **colocar candidatos em universidades** (USA): os candidatos propõem-se sendo o resultado ótimo do ponto de vista dos candidatos (péssimo para as universidades).

Colocar Alunos - Universidades e Internos - Hospitais

- Afetação do tipo **um-para-vários**: cada universidade (hospital) pode ter **várias vagas**, mas cada uma será preenchida no máximo por um candidato;
- Preferências possivelmente incompletas, mas estritamente ordenadas;
- **Extensão do Algoritmo de Gale-Shapley [1962]** para **colocar candidatos em universidades** (USA): os candidatos propõem-se sendo o resultado ótimo do ponto de vista dos candidatos (péssimo para as universidades).

- Descobriu-se depois que um algoritmo análogo era usado desde 1952 nos USA, pelo *National Intern Matching Program*, agora designado *National Resident Matching Program* (NRMP), para **colocar recém licenciados em medicina nos hospitais** para o internato.

Colocar Alunos - Universidades e Internos - Hospitais

- Afetação do tipo **um-para-vários**: cada universidade (hospital) pode ter **várias vagas**, mas cada uma será preenchida no máximo por um candidato;
- Preferências possivelmente incompletas, mas estritamente ordenadas;
- **Extensão do Algoritmo de Gale-Shapley [1962]** para **colocar candidatos em universidades** (USA): os candidatos propõem-se sendo o resultado ótimo do ponto de vista dos candidatos (péssimo para as universidades).

- Descobriu-se depois que um algoritmo análogo era usado desde 1952 nos USA, pelo *National Intern Matching Program*, agora designado *National Resident Matching Program* (NRMP), para **colocar recém licenciados em medicina nos hospitais** para o internato. Os hospitais fazem as propostas. O resultado é ótimo do ponto de vista dos hospitais.

Por abuso de linguagem, continuaremos a chamar emparelhamento à solução (resultado da colocação).

Noção de estabilidade:
Um emparelhamento é **instável** sse existir um interno (*resident*) r e um hospital h tais que:

- h é aceitável para r e r para h;
- r não ficou colocado ou prefere h ao hospital em que ficou colocado;
- h ficou com vagas por preencher ou h prefere r a pelo menos um dos internos com que ficou.

Colocar Alunos - Universidades e Internos - Hospitais

Algoritmo Gale-Shapley orientado por Internos

Inicialmente, todos os internos estão livres.
Inicialmente, todas as vagas nos hospitais estão livres.
Enquanto existir algum interno r livre cuja lista de preferências é não vazia
    seja h o primeiro hospital na lista de r;
    se h não tiver vagas
        seja r' o pior interno colocado provisoriamente em h;
        r' fica livre (passa a não estar colocado);
    colocar provisoriamente r em h;
    se h ficar sem vagas então
        seja s o pior dos colocados provisoriamente em h;
        para cada sucessor s' de s na lista de h
            remover s' e h das respetivas listas

Colocar Alunos - Universidades e Internos - Hospitais

ALGORITMO GALE-SHAPLEY ORIENTADO POR INTERNOS

Inicialmente, todos os internos estão livres.
Inicialmente, todas as vagas nos hospitais estão livres.
Enquanto existir algum interno r livre cuja lista de preferências é não vazia
    seja h o primeiro hospital na lista de r;
    se h não tiver vagas
        seja r' o pior interno colocado provisoriamente em h;
        r' fica livre (passa a não estar colocado);
    colocar provisoriamente r em h;
    se h ficar sem vagas então
        seja s o pior dos colocados provisoriamente em h;
        para cada sucessor s' de s na lista de h
            remover s' e h das respetivas listas

Propriedade: *o emparelhamento resultante é estável e é ótimo segundo candidatos. Cada candidato que não ficar colocado por este algoritmo não pode ficar colocado por nenhum outro algoritmo que produza um emparelhamento estável.*

Listas de Preferências Incompletas e com Empates (SMTI)

D.F.Manlove, R. Irving, K. Iwama, S. Miyazaki, Y. Morita, Hard variants of stable marriage, Theoretical Computer Science, 276: 261-279, 2002.

. . . Here, we present the first comprehensive study of variants of the problem in which the preference lists of the participants are not necessarily complete and not necessarily totally ordered. We show that, under surprisingly restrictive assumptions, a number of these variants are hard, and hard to approximate. The key observation is that, in contrast to the case where preference lists are complete or strictly ordered (or both), a given problem instance may admit stable matchings of different sizes. (. . .) Examples of problems that are hard:

- *Finding a stable matching of maximum or minimum size; determining whether a pair is stable, even if indifference takes the form of ties on one side only, the ties are at the tails of the lists, there is at most one tie per list, and each tie has length two, and*
- *Finding or approximating, both "an egalitarian" and a "minimum regret" stable matching.*

Atraso na Colocação de Professores em 2004 e 2014

Concurso nacional (2004)

- mais de 100 000 candidatos, estritamente ordenados segundo uma lista de graduação, e poucas vagas;
- alguns candidatos não podem ficar sem lugar: têm já lugares que podem manter se não ficarem colocados;
- os candidatos manifestam preferências por escolas ou regiões e por várias cargas horárias; até $\approx$ 100 escolas, $\approx$ 50 concelhos e $\approx$ 23 quadros de zona pedagógica (cobrem todas as escolas públicas);
- podem misturar escolas e concelhos; se indicassem concelho estariam a manifestar a mesma preferência por todas as escolas nessa região.

Oferta de Escola: concursos **descentralizados** para contratos a termo ou vagas que surgiam durante o ano letivo.

Atraso na Colocação de Professores em 2004 e 2014

- Legislação **complexa** e com **alterações frequentes**.
- Em Setembro de 2004 e de 2014, alterações levaram a **atrasos significativos** nas colocações, amplamente debatidos na comunicação social.
  - 2004 – Primeiros resultados violavam a lista de graduação
  - 2014 – Centralização dos concursos descentralizados! Candidatos receberam diversas ofertas e tinham algum tempo para optar $\rightsquigarrow$ **atrasos nas colocações . . .**

Atraso na Colocação de Professores em 2004 e 2014

- Legislação **complexa** e com **alterações frequentes**.
- Em Setembro de 2004 e de 2014, alterações levaram a **atrasos significativos** nas colocações, amplamente debatidos na comunicação social.
  - 2004 – Primeiros resultados violavam a lista de graduação
    - Levou a trocar de empresa. Solução apresentada na TV assumia preferências estritamente ordenadas.
    - No "Aviso de abertura do concurso" dizia que para desempatar, **as escolas numa região são ordenadas por código de escola.**
  - 2014 – Centralização dos concursos descentralizados! Candidatos receberam diversas ofertas e tinham algum tempo para optar $\rightsquigarrow$ **atrasos nas colocações ...**

Atraso na Colocação de Professores em 2004 e 2014

- Legislação **complexa** e com **alterações frequentes**.
- Em Setembro de 2004 e de 2014, alterações levaram a **atrasos significativos** nas colocações, amplamente debatidos na comunicação social.
  - 2004 – Primeiros resultados violavam a lista de graduação
    - Levou a trocar de empresa. Solução apresentada na TV assumia preferências estritamente ordenadas.
    - No “Aviso de abertura do concurso” dizia que para desempatar, **as escolas numa região são ordenadas por código de escola.**
  - 2014 – Centralização dos concursos descentralizados! Candidatos receberam diversas ofertas e tinham algum tempo para optar $\rightsquigarrow$ **atrasos nas colocações ...**
    - Listas de graduação distintas. Candidatos não indicam preferências, o que corresponde a ter empates nas listas de preferências.
    - Não era um *bug* do algoritmo. Problema intrínseco à legislação (entretanto já revista): se quisermos obter um emparelhamento estável de tamanho máximo, o problema seria NP-hard.

Complexidade do Problema sem Regras para Desempate?

Examplo

a_1	:	$\{p_1, p_2, p_3, p_4\}$
a_2	:	$\{p_1, p_2, p_3, p_4\}$
a_3	:	$\{p_1\}$
a_4	:	$\{p_2\}$
a_5	:	$\{p_2, p_3\}$, (tinha p_4)

Quatro emparelhamentos estáveis (*weak stable*):

$\{(a_1, p_1), (a_2, p_2), (a_3, \text{sem vaga}), (a_4, \text{no post}), (a_5, p_3)\}$

$\{(a_1, p_3), (a_2, p_1), (a_3, \text{sem vaga}), (a_4, p_2), (a_5, p_4)\}$

$\{(a_1, p_3), (a_2, p_2), (a_3, p_1), (a_4, \text{sem vaga}), (a_5, p_4)\}$
$\{(a_1, p_2), (a_2, p_3), (a_3, p_1), (a_4, \text{sem vaga}), (a_5, p_4)\}$

Com regra de desempate "código de escola", se $p_1 < p_2 < p_3 < p_4$, o primeiro é a única solução.

Sem lugar? **Tempo de serviço** é um parâmetro de seriação posteriormente.

Complexidade do Problema sem Regras para Desempate?

Examplo

a_1 :	$\{p_1, p_2, p_3, p_4\}$
a_2 :	$\{p_1, p_2, p_3, p_4\}$
a_3 :	$\{p_1\}$
a_4 :	$\{p_2\}$
a_5 :	$\{p_2, p_3\}$, (tinha p_4)

Quatro emparelhamentos estáveis (*weak stable*):

$\{(a_1, p_1), (a_2, p_2), (a_3, \text{sem vaga}), (a_4, \text{no post}), (a_5, p_3)\}$

$\{(a_1, p_3), (a_2, p_1), (a_3, \text{sem vaga}), (a_4, p_2), (a_5, p_4)\}$

$\{(a_1, p_3), (a_2, p_2), (a_3, p_1), (a_4, \text{sem vaga}), (a_5, p_4)\}$
$\{(a_1, p_2), (a_2, p_3), (a_3, p_1), (a_4, \text{sem vaga}), (a_5, p_4)\}$

Com regra de desempate "código de escola", se $p_1 < p_2 < p_3 < p_4$, o primeiro é a única solução.

Sem lugar? **Tempo de serviço** é um parâmetro de seriação posteriormente.
Comparação de perfis de preferências:

$$(1,1,1,\infty,2) \prec_{LEX} (1,1,\infty,1,1) \prec_{LEX} (1,1,\infty,\infty,1)$$

Complexidade do Problema sem Regras para Desempate?

Examplo

a_1 :	$\{p_1, p_2, p_3, p_4\}$
a_2 :	$\{p_1, p_2, p_3, p_4\}$
a_3 :	$\{p_1\}$
a_4 :	$\{p_2\}$
a_5 :	$\{p_2, p_3\}$, (tinha p_4)

Quatro emparelhamentos estáveis (*weak stable*):

$\{(a_1, p_1), (a_2, p_2), (a_3, \text{sem vaga}), (a_4, \text{no post}), (a_5, p_3)\}$

$\{(a_1, p_3), (a_2, p_1), (a_3, \text{sem vaga}), (a_4, p_2), (a_5, p_4)\}$

$\{(a_1, p_3), (a_2, p_2), (a_3, p_1), (a_4, \text{sem vaga}), (a_5, p_4)\}$
$\{(a_1, p_2), (a_2, p_3), (a_3, p_1), (a_4, \text{sem vaga}), (a_5, p_4)\}$

Com regra de desempate "código de escola", se $p_1 < p_2 < p_3 < p_4$, o primeiro é a única solução.

Sem lugar? **Tempo de serviço** é um parâmetro de seriação posteriormente.
Comparação de perfis de preferências:

$$(1,1,1,\infty,2) \prec_{LEX} (1,1,\infty,1,1) \prec_{LEX} (1,1,\infty,\infty,1)$$

Problema TRP: Procurar emparelhamentos estáveis ótimos para os candidatos i.e., **estáveis e cujo perfil seja lexicograficamente mínimo.**

TRP com preferências estritas e capacidades unitárias

Algoritmo 1 - baseado no algoritmo de Gale-Shapley (com "last resort" em $Prefs[a_i]$)

$M := \emptyset$; $I := i := 1$;
while $i \leq n_1$ **do**
    $p_j := Prefs[a_i].\mathrm{pop}()$;
    if p_j is free or p_j is the dummy post p_0 **then** ▷ p_0 with unbounded capacity
        $M := M \oplus \{\langle a_i, p_j\rangle\}$; ▷ assigns a_i to p_j $X \oplus Y = (X \cup Y) \setminus (X \cap Y)$
        $i := I + 1$; $I := I + 1$; ▷ all agents from 1 to I were matched
    else
        $a_k := M(p_j)$;
        if $(k > i$ and $a_k \neq \textsc{holds}(p_j))$ or $(k < i$ and $a_i = \textsc{holds}(p_j))$ **then**
            $M := M \oplus \{\langle a_k, p_j\rangle, \langle a_i, p_j\rangle\}$; $i := k$; ▷ a_k becomes free;
        end if
    end if
end while

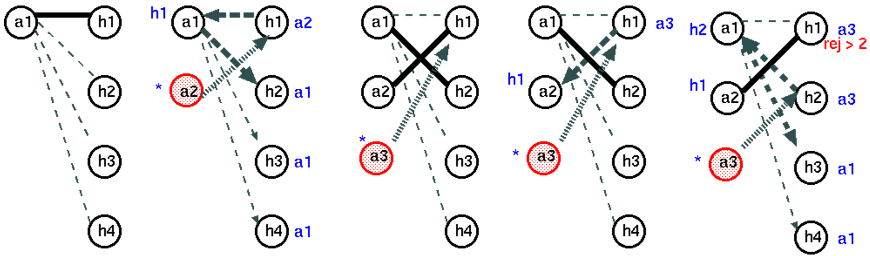
Algoritmo 1 - baseado no algoritmo de Gale-Shapley (com "last resort" em $Prefs[a_i]$)

```
M := ∅;  I := i := 1;
while i ≤ n₁ do
    p_j := Prefs[a_i].pop();
    if p_j is free or p_j is the dummy post p_0 then         ▷ p_0 with unbounded capacity
        M := M ⊕ {⟨a_i, p_j⟩};                    ▷ assigns a_i to p_j    X ⊕ Y = (X ∪ Y) \ (X ∩ Y)
        i := I + 1;  I := I + 1;                             ▷ all agents from 1 to I were matched
    else
        a_k := M(p_j);
        if (k > i and a_k ≠ HOLDS(p_j)) or (k < i and a_i = HOLDS(p_j)) then
            M := M ⊕ {⟨a_k, p_j⟩, ⟨a_i, p_j⟩};   i := k;           ▷ a_k becomes free;
        end if
    end if
end while
```

Proposição: TRP com capacidades unitárias e sem empates admite uma solução única, a qual pode ser obtida pelo Algoritmo 1 em tempo $O(n+m)$, sendo m o comprimento total das listas de preferências e $n = n_1 + n_2$ (o número de nós).

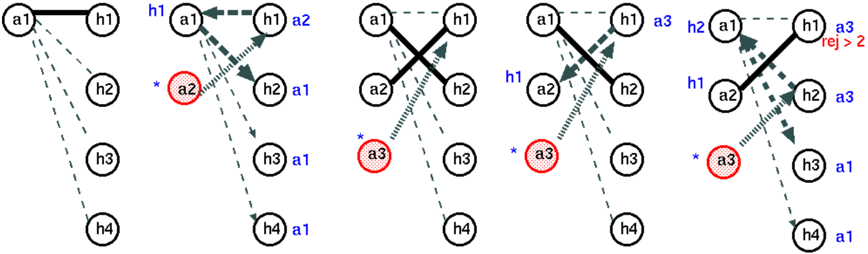
TRP (com empates) - Complexidade Polinomial

$a_1 : \{h_1, h_2, h_3, h_4\}, \quad a_2 : \{h_1\}, \{h_2\}, \quad a_3 : \{h_1\}, \{h_2\},$



TRP (com empates) - Complexidade Polinomial

$a_1 : \{h_1, h_2, h_3, h_4\}, \quad a_2 : \{h_1\}, \{h_2\}, \quad a_3 : \{h_1\}, \{h_2\},$

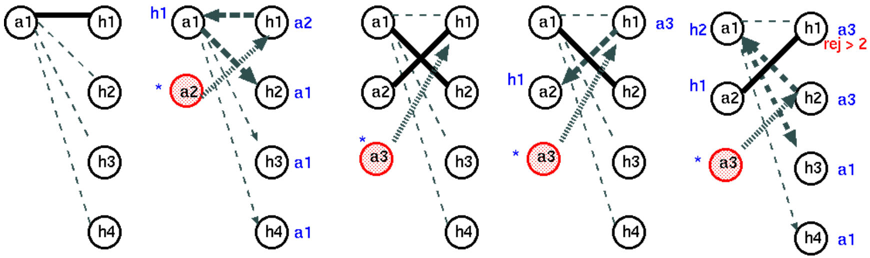


TRP pode ser resolvido por uma sequência de problemas de **emparelhamento de cardinal máximo em grafos bipartidos**, que são subgrafos do original.

- Cada subproblema inclui um nível de preferência para cada a_i já atingido.

TRP (com empates) - Complexidade Polinomial

$a_1 : \{h_1, h_2, h_3, h_4\}, \quad a_2 : \{h_1\}, \{h_2\}, \quad a_3 : \{h_1\}, \{h_2\},$

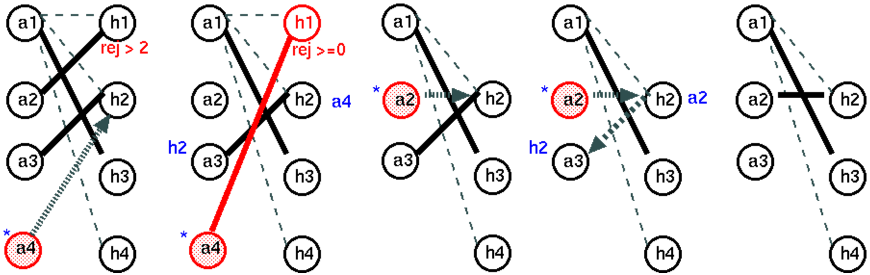


TRP pode ser resolvido por uma sequência de problemas de **emparelhamento de cardinal máximo em grafos bipartidos**, que são subgrafos do original.

- Cada subproblema inclui um nível de preferência para cada a_i já atingido.
- Em cada fase, tenta-se aumentar um emparelhamento provisório, acrescentando mais um candidato. TRP resolve-se em tempo polinomial.

TRP (com empates) - Complexidade Polinomial

$a_1 : \{h_1, h_2, h_3, h_4\}, \;\; a_2 : \{h_1\}, \{h_2\}, \;\; a_3 : \{h_1\}, \{h_2\}, \;\; a_4 : \{h_2\}$ tem h_1



- Encontrar caminhos para aumento ou para troca, por BFS ou DFS.

Para mais detalhes, consultar A.P.Tomás (2018). House Allocation Problems with Tenants and Priorities (for Teacher Recruitment). Publicado em SOFSEM 2018. `https://link.springer.com/chapter/10.1007/978-3-319-73117-9_34`