**UNIVERSIDADE FEDERAL DO CEARÁ**
**CAMPUS QUIXADÁ**
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**DANIEL VITOR PEREIRA RODRIGUES**

# ALGORITMOS GALE-SHAPLEY E IRVING PARA O PROBLEMA DO CASAMENTO ESTÁVEL COM CUSTO IGUALITÁRIO

**QUIXADÁ**
**2024**

DANIEL VITOR PEREIRA RODRIGUES

# ALGORITMOS GALE-SHAPLEY E IRVING PARA O PROBLEMA DO CASAMENTO ESTÁVEL COM CUSTO IGUALITÁRIO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: Prof. Dr. Wladimir Araújo Tavares.

QUIXADÁ

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R612a Rodrigues, Daniel Vitor Pereira.
Algoritmos Gale-Shapley e Irving para o problema do casamento estável com custo igualitário / Daniel Vitor Pereira Rodrigues. – 2024.
43 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2024.
Orientação: Prof. Dr. Wladimir Araújo Tavares.

1. Algoritmos. 2. Otimização. 3. Grafos. 4. Complexidade. I. Título.

CDD 005.1

DANIEL VITOR PEREIRA RODRIGUES

ALGORITMOS GALE-SHAPLEY E IRVING PARA O PROBLEMA DO CASAMENTO ESTÁVEL COM CUSTO IGUALITÁRIO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em: 25/09/2024.

BANCA EXAMINADORA

Prof. Dr. Wladimir Araújo Tavares (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. André Ribeiro Braga
Universidade Federal do Ceará (UFC)

Prof. Dr. Ricardo Reis Pereira
Universidade Federal do Ceará (UFC)

À minha família, minha namorada e meus amigos, por sempre acreditarem e mim e sempre me ajudarem no caminho que trilho, sem eles eu não seria metade do que sou hoje.

**AGRADECIMENTOS**

Primeiramente, gostaria de agradecer à minha mãe, Ireuda Leite, por tudo que já fez e que ainda faz por mim, e que ainda vai fazer. Ela é minha maior inspiração.

À minha namorada, Natalia Tavares, que sempre me apoiou nesse período do curso e sempre me levantava quando eu pensava em desistir, e por nunca sair do meu lado. Por sempre sonhar comigo e dividir as alegrias e tristezas ao meu lado.

Aos meus avós, Antônio Alves e Ivolina Pereira, por sempre cuidarem bem de mim e me protegerem sempre, além de me apoiarem em todas as minhas decisões.

Gostaria de agradecer também à minha irmã Patricia Leite e ao meu irmão, Marcus Vinicius, por todo apoio e por sempre estarem comigo, acreditando no meu potencial.

Ao professor Dr. Wladimir Tavares por me orientar nesse trabalho, assim como por sempre me apoiar também na Maratona de Programação da SBC, e na Olimpiada Brasileira de Programação, que fez parte de toda minha graduação.

Aos professores André Ribeiro Braga e Ricardo Reis Pereira por aceitarem participar da minha banca de avaliação e por sugerirem melhorias para o meu texto.

Aos meus amigos de Engenharia de Computação, que não vou lembrar todos os nomes, mas todos estão guardados aqui na memória, todos os bons momentos que vivemos pelos corredores da UFC, por fazerem a jornada ser um pouco mais leve e divertida.

Aos meus amigos de Quixadá, que me fizeram companhia pelos corredores da UFC, na área de convivência, no RU, nos laboratórios, nas caminhadas de volta para o centro e pelas trilhas de Quixadá.

Aos integrantes da minha equipe da Maratona de Programação da SBC, Felipe Freitag e Gustavo Henrique, por aceitarem treinar e estudar comigo semanalmente, e por realizarem o sonho de ir para a Final Brasileira da Maratona. Ao Álex Sousa e ao Gerdeson Oliveira por terem começado esse sonho comigo em 2019, e pelas noites no laboratório da UFC estudando e resolvendo problemas.

Hoje sei que minha melhor escolha foi ter entrado na Universidade Federal do Ceará, por ter me matriculado e por ter tido coragem de enfrentar uma cidade nova, pessoas novas e desafios novos também.

A todos, minha eterna gratidão.

“Tenho a impressão de ter sido uma criança brincando à beira-mar, divertindo-me em descobrir uma pedrinha mais lisa ou uma concha mais bonita que as outras, enquanto o imenso oceano da verdade continua misterioso diante de meus olhos” (Isaac Newton.)

**RESUMO**

O trabalho propõe uma análise comparativa entre dois algoritmos de casamento estável: o Algoritmo Gale-Shapley e o Algoritmo de Irving. O contexto se baseia na teoria dos grafos, onde grafos bipartidos são fundamentais para representar relações entre conjuntos de entidades. O Problema do Casamento Estável é uma representação matemática de alocação de pares entre homens e mulheres com preferências, tendo aplicações em economia, computação e teoria dos jogos. O Algoritmo de Gale-Shapley, proposto em 1962, estabelece casamentos estáveis sem incentivos para mudanças por parte dos participantes. No entanto, estudos posteriores indicam que esse algoritmo pode favorecer desproporcionalmente um dos gêneros, surgindo a necessidade de um critério mais equitativo. O Algoritmo de Irving, de 1987, busca maximizar a "satisfação"média de todas as pessoas envolvidas. O trabalho tem como objetivo implementar e comparar esses algoritmos, realizando uma revisão bibliográfica detalhada sobre ambos, compreendendo sua lógica e funcionamento. Além disso, visa desenvolver implementações dos algoritmos e compará-los com base em métricas de qualidade da solução e tempo de execução. A análise se concentra em entender como esses algoritmos se comportam diante de diferentes conjuntos de preferências e condições, explorando métricas como a solução igualitária.

**Palavras-chave**: algoritmo; casamento estável; gale-shapley; irving.

## ABSTRACT

The paper proposes a comparative analysis between two stable matching algorithms: the Gale-Shapley Algorithm and the Irving Algorithm. The context is grounded in graph theory, where bipartite graphs are essential in representing relationships between sets of entities. The Stable Marriage Problem is a mathematical representation of pairing allocation between men and women with preferences, finding applications in economics, computing, and game theory. The Gale-Shapley Algorithm, introduced in 1962, establishes stable marriages without incentives for participant-initiated changes. However, subsequent studies suggest that this algorithm might disproportionately favor one gender, leading to the need for a more equitable criterion. The Irving Algorithm, from 1987, aims to maximize the average "satisfaction"of all involved individuals. The objective of the study is to implement and compare these algorithms, conducting a comprehensive bibliographic review of both to understand their logic and functionality. Furthermore, it aims to develop algorithm implementations and compare them based on solution quality metrics and execution time. The analysis focuses on comprehending how these algorithms behave concerning different sets of preferences and conditions, exploring metrics such as equalitarian solutions.


Keywords: algorithm; stable matching; gale-shapley; irving.

## LISTA DE ILUSTRAÇÕES

**LISTA DE TABELAS**

# LISTA DE ALGORITMOS

SUMÁRIO

# 1 INTRODUÇÃO

Os grafos, fundamentais na representação de relações e interconexões presentes nos mais diversos fenômenos, desempenham um importante papel na resolução de problemas do mundo real (Cormen *et al.* 2009). Os algoritmos e heurísticas aplicados a grafos representam uma área de estudo em constante evolução, sendo a base para a resolução de uma variedade de problemas complexos (Eiben e Smith 2015).

O problema do casamento estável (SMP, do inglês Stable Marriage Problem) é uma representação matemática de uma situação de alocação de pares de dois conjuntos, um conjunto de homens e um conjunto de mulheres, no qual cada participante tem uma lista de preferência sobre os membros do outro conjunto. Este problema, proposto por Gale e Shapley (1962), tem sido extensivamente estudado e aplicado em diversas áreas, incluindo economia, ciência da computação e teoria dos jogos. A solução inicial de Gale e Shapley, que tem complexidade assintótica $O(n^2)$, conhecida como Algoritmo de Gale-Shapley, estabelece casamentos estáveis, garantindo que não haja incentivos para mudanças por parte dos participantes (Gale e Shapley 1962).

Outros estudos, como Irving *et al.* 1987, trouxeram contribuições para a compreensão aprofundada do problema e propuseram variantes e extensões que ampliaram ainda mais sua aplicabilidade. A pesquisa sobre o problema do casamento estável destaca sua relevância na teoria dos jogos e na otimização de alocações em diferentes contextos (Irving *et al.* 1987). Recentemente ele também atraiu diversos físicos que, usando as ferramentas poderosas da mecânica estatística, também o abordaram como um problema de otimização (Fenoaltea *et al.* 2021).

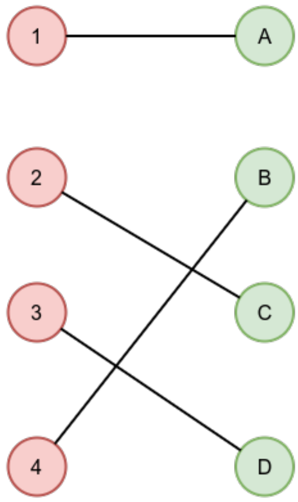
Gale e Shapley 1962 mostrou que, ao finalizar a execução do algoritmo, todos os casamentos obtidos são estáveis. Chamamos um conjunto de casamentos de instável se nele houver um homem e uma mulher que não são casados um com o outro, mas que preferem um ao outro aos seus verdadeiros companheiros. É bem conhecido que existe ao menos uma solução somente com casamentos estáveis para cada instância do problema (Gale e Shapley 1962). A Figura 1 mostra um exemplo de casamento estável.

Contudo, é provado que o algoritmo clássico de Gale-Shapley produz um casamento que favorece enormemente os homens em detrimento das mulheres (solução homem-ótima), ou vice-versa (solução mulher-ótima). Além disso, na solução homem-ótima, cada mulher tem o pior parceiro que ela pode ter em qualquer solução estável e que em uma solução mulher-ótima, cada homem tem o pior parceiro que ele pode ter em qualquer solução estável (Gusfield e Irving 1989). O problema surge em encontrar um emparelhamento estável que seja ótimo sob algum critério

Figura 1 – Exemplo de Casamento Estável

| Pessoa | Lista de preferência |
|---|---|
| 1 | [D, A, B, C] |
| 2 | [C, D, A, B] |
| 3 | [D, B, C, A] |
| 4 | [B, C, A, D] |

| Pessoa | Lista de preferência |
|---|---|
| A | [4, 2, 3, 1] |
| B | [1, 3, 2, 4] |
| C | [2, 3, 4, 1] |
| D | [3, 1, 4, 2] |




Fonte: elaborado pelo autor (2023) utilizando site draw.io

de otimalidade mais equitativo ou igualitário. Irving *et al.* 1987 utiliza o objetivo de maximizar a "satisfação" média (ou, equivalentemente, total) de todas as pessoas. Este objetivo é alcançado quando a satisfação de uma pessoa é medida pela posição do seu parceiro na sua lista de preferências.

Neste trabalho, apresentaremos o algoritmo proposto por Irving *et al.* 1987 de uma maneira mais intuitiva.

Duas variantes das listas de preferência do SMP foram listadas por Iwama e Miyazaki 2008, são elas Casamento Estável com listas incompletas (SMPI) e Casamento Estável com Empates (SMPT). No caso das SMPI, dizemos que se uma lista de uma pessoa $p$ contém a pessoa $q$, e a lista da pessoa $q$ contém a pessoa $p$, então o par $(p,q)$ é aceitável. Nesse caso, como estamos considerando listas incompletas, uma correspondência pode não ser perfeita, ou seja, nem todas as pessoas terão um par. A outra extensão, SMPT, é quando é permitido existirem empates em uma lista de preferência, ou seja, duas pessoas $p$ e $q$ terem a mesma preferência na lista de uma pessoa $r$ (Iwama e Miyazaki 2008).

## 1.1 Objetivos

### *1.1.1 Objetivo Geral*

Fazer análise e implementação de algoritmos para resolução do problema do casamento estável, sendo eles o Gale-Shapley (Gale e Shapley 1962) e o Irving (Irving *et al.* 1987).

### *1.1.2 Objetivos Específicos*

- Realizar uma revisão bibliográfica sobre os algoritmos de casamento estável, Algoritmo de Gale-Shapley e Algoritmo de Irving.
- Entender o funcionamento e a lógica do problema do casamento estável.
- Desenvolver uma implementação de ambos os algoritmos de Gale-Shapley e de Irving utilizando a métrica de casamento igualitário.
- Gerar uma base de testes e executar esses testes com ambos os algoritmos.

# 2 FUNDAMENTAÇÃO TEÓRICA

## 2.1 O problema do Casamento Estável

Historicamente, o problema de casamento estável foi formalmente apresentado por Gale e Shapley 1962. Uma instância $I$ do SMP consiste num mesmo número $n$ de homens e mulheres e, para cada pessoa, uma lista de preferência que ordena todos os outros do gênero oposto (Fenoaltea *et al.* 2021). Se um homem $m$ preferir uma mulher $w_1$ a $w_2$, será escrito $w_1 \succ_m w_2$. A mesma notação é usada para as mulheres, por exemplo, se uma mulher $w$ preferir um homem $m_1$ a $m_2$, será escrito $m_1 \succ_w m_2$.

### *2.1.1 Definição de Solução*

De acordo com Iwama e Miyazaki 2008, uma solução $M$ de $I$ é um conjunto disjunto de pares homem-mulher de $I$. Uma solução é perfeita se ela contém exatamente $n$ pares homem-mulher. Se, em uma solução, um homem $m$ e uma mulher $w$ estão juntos, então temos que $M(m) = w$ e $M(w) = m$. É dito que um um homem $m$ e uma mulher $w$ formam um par bloqueante se as seguintes três condições acontecem:

(i) $M(m) \neq w$;

(ii) $w \succ_m M(m)$; e

(iii) $m \succ_w M(w)$.

Se essas três condições forem verdadeiras para algum par $m$ e $w$, isso significa que eles formam um par bloqueante, pois ambos preferem estar juntos a manter seus parceiros designados na solução proposta. Assim, a solução não seria estável, já que existe a possibilidade de $m$ e $w$ deixarem seus parceiros atuais para ficarem juntos, quebrando a estabilidade do casamento.

### *2.1.2 Soluções estáveis*

É definida uma solução estável pela inexistência de pares bloqueantes. Uma solução é dita instável, se um homem e uma mulher não estão casados, mas preferem se casar à permanecer com seus parceiros atuais (Gale e Shapley 1962).

Vamos ver um exemplo com três homens e três mulheres. Considere a seguinte lista de preferências (2).

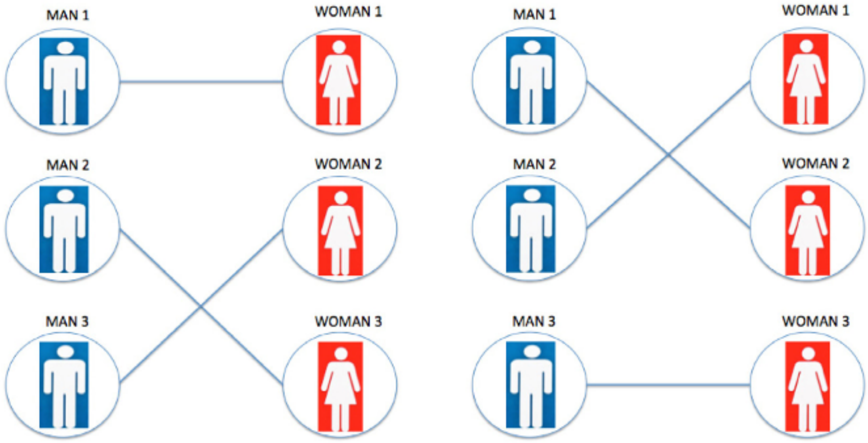
A Figura 3 mostra um exemplo de solução estável (à esquerda), e instável (à direita).

Figura 2 – Listas de preferência

| SMP instance | | | |
|---|---|---|---|
| Man | Man's preference-list | Woman | Woman's preference-list |
| $m_1$ | $(w_1, w_2, w_3)$ | $w_1$ | $(m_3, m_1, m_2)$ |
| $m_2$ | $(w_1, w_3, w_2)$ | $w_2$ | $(m_1, m_2, m_3)$ |
| $m_3$ | $(w_2, w_3, w_1)$ | $w_3$ | $(m_3, m_2, m_1)$ |

Fonte: Fenoaltea *et al.* 2021.

Figura 3 – Exemplo de uma solução estável (esquerda) e instável (direita)




Fonte: Fenoaltea *et al.* 2021.

Na figura da esquerda, o primeiro homem ($m_1$) está casado com a parceira de sua preferência ($w_1$). O mesmo para o terceiro homem ($m_3$). O segundo homem ($m_2$), não está casado com a parceira de sua preferência. Essa solução é estável pois não existe nenhum par de homem e mulher tal que não estão casados, mas preferem se casar entre si à permanecer no par atual. Na solução da direita, podemos perceber que o par ($m_1$, $w_1$) é um par bloqueante, pois eles não estão casados, mas preferem um ao outro do que seus pares atuais.

### 2.1.3 *Medidas de otimalidade*

Para uma solução estável $M$ de uma instância $I$ do SMP, definimos $mr(m,w)$ como a posição de uma mulher $w$ na lista de preferência de um homem $m$, e $wr(w,m)$ a posição de um homem $m$ na lista de preferência de uma mulher $w$. Além disso, definimos o custo-homem como $sm(M)$ e o custo-mulher como $sw(M)$ como segue:

$$sm(M) = \sum_{(m,w)\in M} mr(m,\ w), \tag{1}$$

$$sw(M) = \sum_{(m,w)\in M} wr(w,\ m). \tag{2}$$

### *2.1.4 Solução homem-ótimo (ou mulher-ótimo)*

Uma solução estável $M$ é homem-ótima (respectivamente mulher-ótima) se ela tem o menor valor de $sm(M)$ (respectivamente $sw(M)$). O algoritmo de Gale-Shapley é basicamente uma sequência de propostas dos homens para as mulheres para encontrar a solução estável homem-ótima (ou mulher-ótima). Além disso, na solução homem-ótima, cada mulher tem o pior parceiro que ela pode ter em qualquer solução estável e que em uma solução mulher-ótima, cada homem tem o pior parceiro que ele pode ter em qualquer solução estável (Nakamura *et al.* 1995).

### *2.1.5 Solução igualitária e gênero-igualitária*

Para uma solução estável $M$, nós definimos $c(M)$ como custo igualitário e $d(M)$ como custo de gênero-igualidade como segue:

$$c(M) = sm(M) + sw(M), \tag{3}$$

$$d(M) = |sm(M) - sw(M)|. \tag{4}$$

Uma solução $M$ é dita igualitária (respectivamente gênero-igualitária) quando tem o menor valor possível de $c(M)$ (respectivamente $d(M)$).

Existem várias maneiras de encontrar uma solução igualitária ou gênero-igualitária. Nakamura *et al.* 1995 propôs um algoritmo genético para a solução estável gênero-igualitária. Nessa abordagem, o problema é transferido para um grafo direcionado e um algoritmo genético é usado para encontrar a solução no grafo (Viet *et al.* 2016).

Neste trabalho será usado apenas o custo igualitário, ou seja, $c(M)$.

### *2.1.6 Listas de preferência*

As listas de preferência são listas que contém a ordem de preferência de uma pessoa sobre as outras pessoas do gênero oposto. Algumas variantes permitem que as listas estejam incompletas ou que duas ou mais pessoas possam empatar em ordem de preferência.

### *2.1.7 Listas de preferência incompletas*

Nessa variação, conhecida como SMPI, a lista de preferência de cada pessoa $p$ pode estar incompleta, significando que algumas pessoas que foram excluídas não podem formar um par com $p$. Se uma pessoa $q$ está na lista de preferência de uma pessoa $p$, então $q$ é aceitável para $p$. Uma solução é um conjunto de $n$ pares $(w,m)$ tal que $w$ e $m$ são aceitáveis entre si (Iwama e Miyazaki 2008).

Uma propriedade importante é que podemos particionar o conjunto dos homens em dois conjuntos; um deles é o conjunto dos homens que têm parceiras em todos as soluções estáveis, e o outro é o conjunto dos homens que estão sozinhos em todas as soluções estáveis (Gale e Sotomayor 1985). Isso implica que todas as soluções estáveis têm o mesmo tamanho, dessa forma, é fácil ver que o algoritmo de Gale-Shapley resolve essa variante com uma pequena modificação (Iwama e Miyazaki 2008).

### *2.1.8 Listas de preferência com empates*

Conhecida como SMPT, essa outra variação permite que, em uma lista de preferência de uma pessoa $p$, duas ou mais possam ter a mesma preferência, configurando um empate. Aqui existem três noções de estabilidade: super-estabilidade, estabilidade forte e estabilidade fraca.

Na super-estabilidade, um par bloqueante é definido como um par $(m,w)$ tal que $M(m) \neq w$, $w \succeq_m M(m)$, e $m \succeq_w M(w)$.

Na estabilidade forte, um par $(x,y)$ é bloqueante se $M(x) \neq y$, $y \succ_x M(x)$ e $x \succ_y M(y)$.

Finalmente, na estabilidade fraca, um par $(m,w)$ é bloqueante se $M(m) \neq w$, $w \succ_m M(m)$, e $m \succ_w M(w)$ (Iwama e Miyazaki 2008).

Uma solução estável fraca pode ser encontrada facilmente em tempo polinomial (Irving 1994). Algumas instâncias podem não ter soluções super-estáveis ou estabilidade forte. Existe um algoritmo polinomial que decide se uma solução super-estável (ou estável forte) existe, e encontra caso exista, em tempo $O(n^2)$ (Irving 1994).

### *2.1.9 Listas de preferência incompletas e com empates*

Essa variação, conhecida como SMPTI, permite tanto listas incompletas quanto empates nas listas. A estabilidade e a noção de pares bloqueantes também existem como nas variações citadas anteriormente.

Nas super-estabilidade e estabilidade forte, os resultados são similares ao SMPT, existindo um algoritmo para detectar e encontrar uma solução com super-estabilidade e outro para detectar e encontrar uma solução com estabilidade forte. O tempo de execução dos algoritmos são; $O(a)$ para a super-estabilidade (Manlove 1999) e $O(na)$ para a estabilidade forte

(Kavitha *et al.* 2007), onde $a$ é o tamanho total de todas as listas de preferência (que é $2n^2$ se todas as listas forem completas).

Já para a estabilidade fraca, a solução ótima sempre existe e pode ser encontrada em $O(a)$ mas, desta vez, as instâncias podem ter tamanhos distintos e, com isso, o problema de encontrar a maior solução se torna NP-hard (Wiedermann *et al.* 1999).

### *2.1.10 Algoritmo de Gale-Shapley*

Esse algoritmo foi proposto por Gale e Shapley 1962 juntamente ao problema de casamento estável. Ele funciona da seguinte forma:

- Cada homem propõe à sua mulher favorita que ainda não lhe rejeitou. Cada mulher mantém uma lista das propostas que recebe e aceita a melhor proposta até o momento.
- Se uma mulher recebe uma nova proposta, ela compara essa proposta com a que já aceitou. Se a nova proposta for melhor, ela a aceita e rejeita a proposta anterior. Se for pior, ela a rejeita.
- O processo continua até que não haja mais propostas não rejeitadas. Nesse ponto, cada mulher estará associada a um homem e teremos um casamento estável.

O Algoritmo de Gale-Shapley, que executa com complexidade $O(n^2)$, garante que, quando termina, o casamento obtido será estável. Além disso, o algoritmo garante que todos os homens e mulheres estarão casados quando o algoritmo terminar (Gale e Shapley 1962). O Pseudocódigo 1 mostra o pseudocódigo do algoritmo Gale-Shapley.

**Algoritmo 1:** Pseudocódigo do Algoritmo de Gale-Shapley

```
Inicialização: Todos os homens e mulheres estão livres;
while Existe um homem x livre do
    y ⟵ mulher com maior preferência na lista de x que ainda não recebeu proposta de x;
    if y está livre then
        x e y ficam noivos;
    else
        if y prefere x no lugar do atual z then
            x e y ficam noivos;
            z fica livre;
        end
    end
end
retorna lista de casamentos;
```

Fonte: Elaborado pelo autor.

### *2.1.11 Definições*

#### ***Shortlists***

Durante a execução do algoritmo de Gale-Shapley, algumas regras importantes são seguidas, garantindo que as preferências e propostas levem a um casamento estável:

- Regra 1: Se um homem $m$ faz uma proposta para uma mulher $w$, não existe um casamento estável em que $m$ tenha uma parceira melhor que $w$. Isso significa que, após fazer uma proposta, o homem nunca terá uma opção melhor em qualquer casamento estável.
- Regra 2: Se uma mulher $w$ recebe uma proposta de um homem $m$, não existe um casamento estável em que $w$ tenha um parceiro pior que $m$. Isso garante que a mulher só aceitará propostas que melhorem sua situação em relação ao seu parceiro atual.

Essas duas regras sugerem que, sempre que uma mulher $w$ recebe uma proposta de um homem que ela prefere em comparação ao atual parceiro, ela deve remover o parceiro anterior de sua lista, e vice-versa. A partir disso, podemos derivar as *shortlists* — listas reduzidas que contêm apenas os pares viáveis para qualquer casamento estável.

As *shortlists* são um subproduto fundamental gerado a partir do algoritmo de Gale-Shapley. Após a execução desse algoritmo, em que cada homem propõe para as mulheres de acordo com suas preferências, chegamos a uma solução estável inicial. Nesse ponto, as *shortlists* são criadas ao manter, para cada homem e mulher, apenas as opções que podem efetivamente fazer parte de um casamento estável em pelo menos uma das soluções possíveis. Ou seja, qualquer par homem-mulher que não seja viável para um casamento estável em qualquer solução é removido da lista de preferências, resultando em listas reduzidas que chamamos de *shortlists*.

As *shortlists*, ou listas reduzidas, obedecem às seguintes propriedades:

– Propriedade 1: Se uma mulher $w$ não aparece na *shortlist* de um homem $m$, então não existe nenhum casamento estável em que $m$ e $w$ estejam juntos. Ou seja, pares não presentes nas *shortlists* são completamente inviáveis.

– Propriedade 2: Uma mulher $w$ aparece na *shortlist* de $m$ se, e somente se, $m$ também aparece na *shortlist* de $w$. Além disso, $w$ será a primeira opção de $m$ se, e somente se, $m$ for a última opção de $w$. Isso reflete a natureza bidirecional das preferências.

– Propriedade 3: Se todos os homens são emparelhados com a primeira mulher em suas *shortlists*, o casamento resultante será estável. Esse casamento é conhecido como a solução ótima para os homens, pois nenhum homem terá uma parceira melhor em qualquer casamento estável, e nenhuma mulher terá um parceiro pior.

– Propriedade 4: Se os papéis de homens e mulheres forem invertidos, e todas as mulheres forem emparelhadas com o primeiro homem em suas *shortlists*, o casamento resultante será estável. Este é o casamento ótimo para as mulheres, no qual nenhuma mulher pode ter um parceiro melhor e nenhum homem pode ter uma parceira pior.

Considere o seguinte exemplo em que temos 4 homens e 4 mulheres, e suas preferências são dadas a seguir:

| **Preferências dos Homens** | **Preferências das Mulheres** |
|---|---|
| **Homem 1:** 3, 1, 5, 7, 4, 2, 8, 6 | **Mulher 1:** 4, 3, 8, 1, 2, 5, 7, 6 |
| **Homem 2:** 6, 1, 3, 4, 8, 7, 5, 2 | **Mulher 2:** 3, 7, 5, 8, 6, 4, 1, 2 |
| **Homem 3:** 7, 4, 3, 6, 5, 1, 2, 8 | **Mulher 3:** 7, 5, 8, 3, 6, 2, 1, 4 |
| **Homem 4:** 5, 3, 8, 2, 6, 1, 4, 7 | **Mulher 4:** 6, 4, 2, 7, 3, 1, 5, 8 |
| **Homem 5:** 4, 1, 2, 8, 7, 3, 6, 5 | **Mulher 5:** 8, 7, 1, 5, 6, 4, 3, 2 |
| **Homem 6:** 6, 2, 5, 7, 8, 4, 3, 1 | **Mulher 6:** 5, 4, 7, 6, 2, 8, 3, 1 |
| **Homem 7:** 7, 8, 1, 6, 2, 3, 4, 5 | **Mulher 7:** 1, 4, 5, 6, 2, 8, 3, 7 |
| **Homem 8:** 2, 6, 7, 1, 8, 3, 4, 5 | **Mulher 8:** 2, 5, 4, 3, 7, 8, 1, 6 |

| Casamento Formado | Iteração |
|---|---|
| (1, 3) | Homem 1 propõe Mulher 3, proposta aceita<br>Homem 4 é removido da lista da Mulher 3<br>Mulher 3 é removida da lista do Homem 4 |
| (2, 6) | Homem 2 propõe Mulher 6, proposta aceita<br>Homens 8, 3 e 1 são removidos da lista da Mulher 6<br>Mulher 6 é removida das listas dos Homens 8, 3 e 1 |
| (3, 7) | Homem 3 propõe Mulher 7, proposta aceita<br>Homem 7 é removido da lista da Mulher 7<br>Mulher 7 é removida da lista do Homem 7 |
| (4, 5) | Homem 4 propõe Mulher 5, proposta aceita<br>Homens 3 e 2 são removidos da lista da Mulher 5<br>Mulher 5 é removida das listas do Homens 3 e 2 |
| (5, 4) | Homem 5 propõe Mulher 4, proposta aceita<br>Homem 8 é removido da lista da Mulher 4<br>Mulher 4 é removida da lista do Homem 8 |
| (6, 6) | Homem 6 propõe Mulher 6, proposta aceita<br>Homens 2 é removido da lista da Mulher 6<br>Mulher 6 é removida da lista do Homem 2 |
| (7, 8) | Homem 7 propõe Mulher 8, proposta aceita<br>Homens 8, 1 e 6 são removidos da lista da Mulher 8<br>Mulher 8 é removida das listas do Homens 8, 1 e 6 |
| (8, 2) | Homem 8 propõe Mulher 2, proposta aceita<br>Homens 6, 4, 1 e 2 são removidos da lista da Mulher 2<br>Mulher 2 é removida das listas do Homens 6, 4, 1 e 2 |

Após a execução do algoritmo de Gale-Shapley, no qual os homens fazem suas propostas, algumas combinações se tornam inviáveis e são eliminadas das listas de preferências. As *shortlists* resultantes — que contêm apenas os parceiros possíveis que podem aparecer em um casamento estável — podem ser assim definidas:

| Preferências dos Homens | Preferências das Mulheres |
|---|---|
| **Homem 1:** 3, 1, 5, 7, 4 | **Mulher 1:** 4, 3, 8, 1, 2 |
| **Homem 2:** 1, 3, 4, 8, 7 | **Mulher 2:** 3, 7, 5, 8 |
| **Homem 3:** 7, 4, 3, 1, 2, 8 | **Mulher 3:** 7, 5, 8, 3, 6, 2, 1 |
| **Homem 4:** 5, 8, 6, 1, 4, 7 | **Mulher 4:** 6, 4, 2, 7, 3, 1, 5 |
| **Homem 5:** 4, 2, 8, 7, 3, 6, 5 | **Mulher 5:** 8, 7, 1, 5, 6, 4 |
| **Homem 6:** 6, 5, 7, 4, 3 | **Mulher 6:** 5, 4, 7, 6 |
| **Homem 7:** 8, 6, 2, 3, 4, 5 | **Mulher 7:** 1, 4, 5, 6, 2, 8, 3 |
| **Homem 8:** 2, 7, 1, 3, 5 | **Mulher 8:** 2, 5, 4, 3, 7 |

Nesse ponto, as *shortlists* são o resultado das propostas iniciais dos homens e das respostas das mulheres, onde apenas os pares que ainda podem formar um casamento estável permanecem. Note que, por exemplo, o Homem 1 tem 3, 1, 5, 7 e 4 em sua *shortlist*, o que significa que há pelo menos um casamento estável no qual ele pode ser emparelhado com essas mulheres.

### *Definição Formal de Rotação*

No contexto das *shortlists*, o conceito de rotação é central. Uma rotação é uma sequência cíclica de pares $(m_0, w_0), (m_1, w_1), \ldots, (m_{r-1}, w_{r-1})$, onde cada homem tem duas mulheres principais nas suas preferências. A rotação é definida por dois critérios:

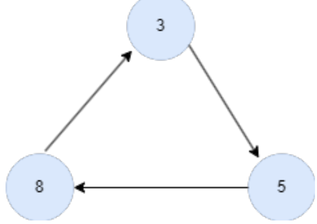
- Para cada homem $m_i$, a mulher $w_i$ é a primeira mulher na *shortlist* de $m_i$.
- A segunda mulher da *shortlist* de $m_i$ é $w_{(i+1) \mod r}$, ou seja, a próxima mulher na rotação.

Por exemplo, considere a seguinte rotação extraída de um conjunto de *shortlists*:

$$P_1 = (1,3),(2,1), \quad P_2 = (3,7),(5,4),(8,2), \quad P_3 = (4,5),(7,8),(6,6)$$

Aqui, o homem 1 tem $w_1 = 3$ como sua primeira escolha e $w_2 = 1$ como a segunda. Similarmente, para o homem 3, $w_1 = 7$ e $w_2 = 4$. Quando uma rotação está exposta, podemos ver que os homens podem trocar suas atuais parceiras pelas segundas opções, enquanto ainda mantêm a estabilidade do casamento.

Figura 4: Grafo da Rotação $P_2$




Fonte: elaborado pelo autor (2024) utilizando site draw.io

Cada rotação gera um grafo, os os nós do gráfico são os homens. Uma aresta do grafo vai de um homem $m_1$ para um homem $m_2$ se a segunda mulher na *shortlist* de $m_1$ é a primeira mulher na lista de preferência do $m_2$. Dessa forma, todos os grafos são em formato de circunferência.

O processo de identificação e eliminação de rotações nas *shortlists* é essencial para o algoritmo de Irving, pois permite que cada homem substitua sua primeira parceira pela segunda da sua *shortlist*, e esse processo pode ser repetido até que nenhuma rotação exposta permaneça.

***Rotações expostas e eliminadas***

Uma rotação $p$ é uma sequência de pares $(m_0, w_0), (m_1, w_1), \ldots, (m_{r-1}, w_{r-1})$, onde cada $m_i$ é um homem e cada $w_i$ é uma mulher. A eliminação de uma rotação ocorre removendo sucessores de $m_{i-1}$ da lista de $w_i$, assim como removendo $w_i$ da lista de sucessores de cada $m_i$. O objetivo é restringir as listas de preferências para que nenhum par indesejado permaneça, levando a um casamento estável.

Por exemplo, considere a rotação $p = \{(m_0, w_0), (m_1, w_1)\}$. Se $m_0$ prefere $w_0$, e $w_0$ prefere $m_1$, ao eliminar $p$, $m_0$ e $w_1$ não podem mais se emparelhar, e todos os sucessores desses pares são removidos. Isso assegura que o processo de eliminação leve a um casamento estável, eliminando combinações que poderiam gerar instabilidade.

$$p = \{(m_0, w_0), (m_1, w_1), \ldots, (m_{r-1}, w_{r-1})\}$$

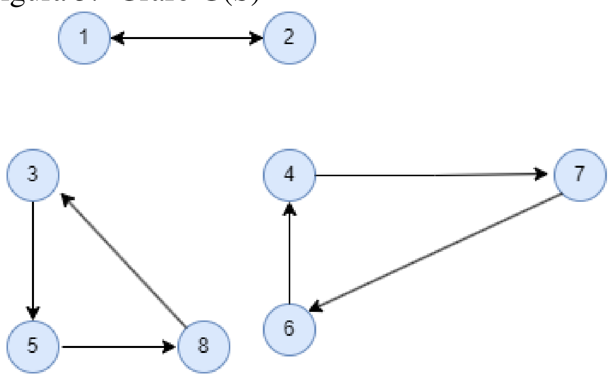
Cada sucessor de $m_{i-1}$ na lista de $w_i$ é removido, junto com a correspondente aparição de $w_i$ nas listas de $m_i$, garantindo a estabilidade do casamento.

***Identificação de todas as rotações expostas***

Para a identificação de todas as rotações, será criado um grafo com todos os homens como nós. Seja $S$ um casamento estável gerado a partir das *shorlists*. Para cada homem $m$ definimos $s(S,m)$ como a mulher que está em segundo lugar na sua *shortlist*, e por $s'(S,m)$ seu marido em $S$. Nós definimos $G(S)$ como o grafo com $n$ nós, um para cada homem, onde, para cada homem $m_i$, existe uma aresta de $m_i$ para $s'(S,m)$.

O grafo $G(S)$ (5) é um grafo onde cada nó tem grau de saída 1, ou seja, ele tem no máximo $n$ arestas e nenhum nó, e também nenhuma aresta, está em mais de um ciclo. Dessa forma, podemos executar uma Busca em Profundidade (DFS) para encontrar os ciclos de um por um. Cada ciclo define uma rotação. Assim, podemos encontrar todas as rotações expostas de uma única vez em $O(n)$.

Figura 5: Grafo G(S)

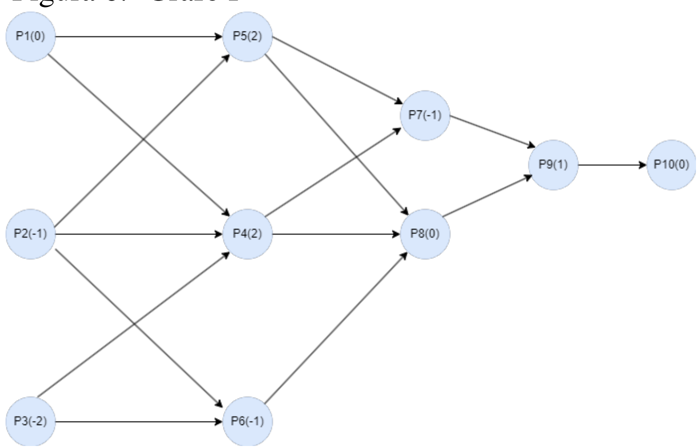



Fonte: elaborado pelo autor (2024) utilizando site draw.io

***Grafo P***

O grafo $P$ é uma estrutura essencial no algoritmo de Irving, utilizado para organizar e representar a relação de precedência entre as rotações. Cada vértice (ou nó) do grafo $P'$ corresponde a uma rotação, e uma aresta direcionada de uma rotação $\rho_1$ para outra rotação $\rho_2$ indica que $\rho_1$ deve ser eliminada antes que $\rho_2$ possa ser exposta e considerada. Assim, $P$ é um grafo direcionado acíclico (DAG), ou seja, não contém ciclos, e segue uma hierarquia de precedência entre as rotações. Grafo gerado a partir de 2.1.11.

Figura 6: Grafo P




Fonte: elaborado pelo autor (2024) utilizando site draw.io

***Predecessores imediatos***

Predecessores imediatos, conforme descrito no trabalho de (Irving *et al.* 1987) sobre o problema do casamento estável, são conceitos centrais para a estruturação da solução. Eles surgem no contexto das rotações, que são subsequências de listas de preferência, onde a troca de parceiros dentro dessas sequências ainda mantém a estabilidade do casamento. Um predecessor imediato de uma rotação é aquela rotação que deve ser eliminada antes para permitir que outra rotação se torne "exposta". Isto significa que uma rotação só se tornará ativa ou visível no processo de eliminação se todas as rotações que a precedem imediatamente já tiverem sido removidas.

Formalmente, uma rotação $r$ é um predecessor imediato de uma rotação $p$ se $r < p$, e não há nenhuma outra rotação $u$ tal que $r < u < p$. Essa relação cria uma ordem parcial entre as rotações, conhecida como *poset de rotações*, onde a eliminação de uma rotação predecessor é uma condição necessária para que uma rotação sucessora possa ser exposta. Esse conceito é fundamental, pois a estrutura de predecessores imediatos define a sequência em que as rotações podem ser processadas, impactando diretamente a obtenção de todas as soluções estáveis possíveis para o problema do casamento.

***Peso dos nós em P***

Cada rotação no grafo $P'$ possui um peso associado, que reflete a mudança total de satisfação dos participantes causada pela troca de parceiros dentro dessa rotação. O peso da rotação $\rho$ pode ser calculado como a soma das diferenças nas classificações das preferências antes e depois da eliminação da rotação:

$$w(\rho) = \sum_{i=0}^{r-1} \left(\mathrm{mr}(m_i, w_i) - \mathrm{mr}(m_i, w_{(i+1) \mod r})\right) + \sum_{i=0}^{r-1} \left(\mathrm{wr}(w_i, m_i) - \mathrm{wr}(w_i, m_{(i-1) \mod r})\right)$$

Aqui, $\mathrm{mr}(m_i, w_i)$ representa a posição de $w_i$ na lista de preferências de $m_i$, e $\mathrm{wr}(w_i, m_i)$ é a posição de $m_i$ na lista de preferências de $w_i$. Assim, o peso de uma rotação reflete o quanto a eliminação dessa rotação melhora (ou piora) a satisfação total do conjunto de homens e mulheres.

A fórmula para calcular o peso de uma rotação foi criada de forma que a seguinte equação seja satisfeita:

$$c(S') = c(S) - w(p)$$

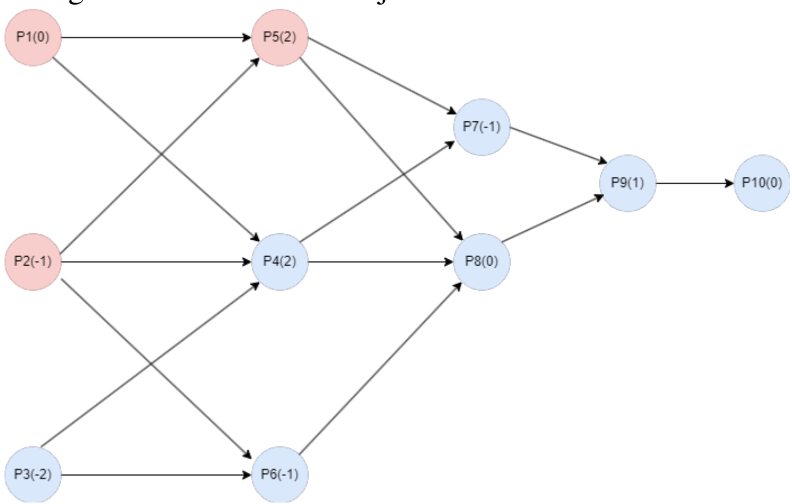
Ou seja, dada uma solução $S$ com custo $c(S)$, após eliminar uma rotação $p$, o custo de $c(S')$ será de acordo com a fórmula. Assim, para encontrar um custo mínimo, basta eliminar uma rotação com custo máximo. Generalizando, temos:

$$c(S) = c(S_0) - \sum_{i=1}^{t} w(\rho_i)$$

***Máximo Subconjunto Fechado***

O objetivo do algoritmo de Irving é encontrar o subconjunto fechado de rotações de máximo peso. Um subconjunto fechado no grafo $P'$ é um conjunto de rotações onde, para cada rotação incluída, todas as suas rotações predecessoras também estão incluídas. Esse subconjunto maximiza a soma dos pesos das rotações eliminadas, resultando no casamento estável ótimo em termos de satisfação total.

Figura 7: Maximo Subconjunto Fechado




Fonte: elaborado pelo autor (2024) utilizando site draw.io

O subconjunto fechado de máximo peso pode ser encontrado utilizando algoritmos de fluxo máximo, como o algoritmo de Ford-Fulkerson. Isso envolve criar um grafo direcionado acíclico (DAG) a partir do grafo $P$, com dois nós a mais, que são o source $s$ e o sink $t$.

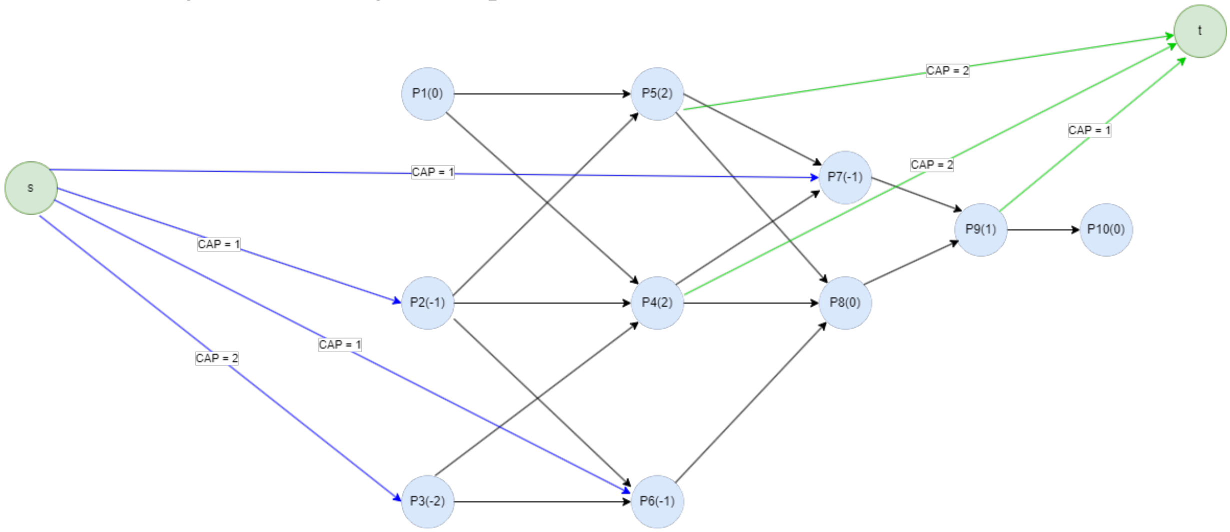
***Encontrando o subconjunto máximo***

Para encontrar o máximo subconjunto fechado, vamos usar o algoritmo de Ford Fulkerson no grafo gerado a partir de 6. Para cada nó do grafo p original, se $w(p) < 0$, criamos uma aresta do source $s$ até o nó $p$, com capacidade $|w(p)|$. Caso $w(p) > 0$, criamos uma aresta com capacidade $w(p)$ de $p$ para o sink $t$. As outras arestas originais terão capacidade infinita.

## 2.2 Algoritmo de Irving

O algoritmo de Irving é uma solução eficiente para o problema do casamento estável com o objetivo de maximizar a satisfação total dos participantes. Este algoritmo explora a estrutura das rotações dentro do conjunto de todas as soluções estáveis para identificar uma

Figura 8: Grafo gerado a partir de P




Fonte: elaborado pelo autor (2024) utilizando site draw.io

solução ótima. A seguir, são descritos os principais componentes do algoritmo, bem como a sua análise de complexidade.

### *2.2.1 Pseudocódigo do Algoritmo de Irving*

Abaixo apresentamos o pseudocódigo para a versão do algoritmo de Irving com complexidade $O(n^4)$:

**Algoritmo 2:** Pseudocódigo detalhado do Algoritmo de Irving

**Inicialização:**

- Gerar as listas de preferências dos homens e mulheres.
- Aplicar o algoritmo de Gale-Shapley para gerar uma solução estável inicial, chamada de *male optimal solution*.
- A partir dessa solução, gerar as *shortlists* para cada homem e mulher, removendo todos os parceiros que nunca poderiam estar presentes em um casamento estável.

**Identificação e Eliminação de Rotações:** **while** *Existirem rotações expostas nas shortlists* **do**

- Identificar uma rotação exposta $\rho = \{(m_0, w_0), \ldots, (m_{r-1}, w_{r-1})\}$.
- **for** *cada par* $(m_i, w_i)$ *em* $\rho$ **do**
  - $m_i$ troca $w_i$ por $w_{(i+1) \mod r}$.
  - Atualizar as listas de preferências de $m_i$ e $w_i$ de acordo com as novas atribuições.

  **end**
- Eliminar a rotação $\rho$, removendo de todas as *shortlists* qualquer homem que tenha sido removido de uma lista feminina e qualquer mulher removida da lista masculina.
- Atualizar as *shortlists* após a eliminação de $\rho$, verificando se novas rotações foram expostas.

**end**

**Construção do Grafo de Rotações:**

- Criar o grafo $P'$, onde cada nó representa uma rotação e as arestas direcionadas representam a relação de precedência entre as rotações (um predecessor imediato de outra).
- Para cada homem $m$, identificar as mulheres em sua lista reduzida que são parceiras em alguma rotação.
- Conectar as rotações com uma aresta direcionada se, para um homem $m_i$, a parceira em uma rotação $\rho_1$ precede a parceira em uma rotação $\rho_2$.
- Repetir o processo para todas as mulheres envolvidas nas rotações.

**Subconjunto Fechado de Máximo Peso:**

- Encontrar o subconjunto fechado de máximo peso no grafo $P'$ utilizando um algoritmo de fluxo máximo.

**Eliminação de Rotações:**

- Para cada rotação no subconjunto fechado de máximo peso, eliminá-la da lista reduzida de preferências, ajustando as preferências dos homens e mulheres envolvidos.

**return** *casamento estável ótimo*

Fonte: Elaborado pelo autor.

### 2.2.2 *Análise de Complexidade*

A análise do algoritmo pode ser dividida em três partes principais:

- **Criação das *shortlists*:** O primeiro passo envolve gerar as *shortlists* a partir do algoritmo de Gale-Shapley, o que tem complexidade $O(n^2)$;
- **Identificação e eliminação de rotações:** Identificar rotações expostas nas listas reduzidas pode ser feito em $O(n^3)$, sendo que cada ciclo de eliminação de rotações tem custo $O(n)$. Assim, a etapa completa de identificação e eliminação de rotações tem complexidade $O(n^3)$;
- **Construção e análise do grafo $P'$:** O grafo $P'$ é gerado com $O(n^2)$ vértices e $O(n^2)$ arestas. O cálculo do subconjunto fechado de máximo peso pode ser realizado em $O(n^4)$, utilizando o algoritmo de Ford-Fulkerson para fluxo máximo;
- **Identificar as rotações do subconjunto máximo** Após encontrar o subconjunto fechado de máximo peso, encontramos todas as rotações que devem ser eliminadas em $O(n^2)$;
- **Eliminação das rotações** Eliminamos as rotações e calculamos o custo dos casamentos em $O(n^2)$.

Dessa forma, a complexidade total do algoritmo é $O(n^4)$.

## 2.3 Variantes do Casamento Estável

O casamento estável têm duas variantes principais, o Problema de Colegas de Quarto Estáveis (SRP) e o Problema dos Hospitais/Residentes (HRP).

### 2.3.1 *Problema de Colegas de Quarto Estáveis*

Esse problema é uma extensão do problema do casamento estável, porém que não acontece em um grafo bipartido. Nesse problema, cada uma das $2n$ pessoas tem uma lista de preferência contendo as outras $2n-1$ pessoas. O problema é encontrar uma solução estável muito parecido com o problema do casamento estável. Diferente do casamento estável, nem sempre podemos encontrar soluções estáveis para esse problema (Iwama e Miyazaki 2008). Irving 1985 propôs um algoritmo de tempo polinomial para decidir se uma instância admite uma solução estável e, se sim, encontrá-la.

A principal diferença entre esse problema e o problema do casamento estável, é que nesse problema existe apenas um grupo, e cada participante tem uma lista com a ordem de preferência das outras $2n-1$ pessoas.

### *2.3.2 Problema dos Hospitais/Residentes*

O Problema dos Hospitais/Residentes (HRP) é uma extensão muitos-para-um do problema do casamento estável, onde consideramos homens como os residentes, e mulheres como os hospitais. Cada hospital declara uma cota $q$, que é a quantidade de residentes que ele aceita atualmente. Normalmente, neste modelo, listas de preferências são geralmente incompletas (Iwama e Miyazaki 2008).

Nós podemos reduzir o Problema dos Hospitais/Residentes para o Problema do Casamento Estável trocando cada hospital com $q$ cópias dele. É também conhecido que muitas soluções para o Problema do Casamento Estável também resolvem o Problema dos Hospitais/-Residentes (Iwama e Miyazaki 2008).

# 3 TRABALHOS RELACIONADOS

A seguir, são apresentados alguns trabalhos que estão relacionados com este.

## 3.1 A Bidirectional Local Search for the Stable Marriage Problem

Viet *et al.* 2016 propõe um algoritmo de busca local bidirecional para encontrar casamentos estáveis igualitários e igualitários de gênero no problema do casamento estável. Ele combina a busca a partir da solução ideal para homens e para mulheres, utilizando uma estratégia de quebra de casamento para encontrar casamentos alternativos estáveis. Simulações demonstraram a eficiência desse algoritmo no problema de casamento estável.

O trabalho de Viet *et al.* 2016 propõe uma alternativa ao algoritmo de Gale e Shapley 1962, ao sugerir um algoritmo que busca uma solução otimamente balanceada, enquanto a de Gale-Shapley sugere apenas uma solução ótima para somente um dos gêneros (homem-ótima ou mulher-ótima). Além disso, ele faz comparações entre alguns algoritmos como hill-climbing (Sabharwal e Selman 2011) e SLS (Viet *et al.* 2016). A diferença para o trabalho atual é que ele toma como base as métricas de solução estável igualitária e gênero-igualitária.

## 3.2 Stable Marriage Problems with Ties and Incomplete Preferences: An Empirical Comparison of ASP, SAT, ILP, CP, and Local Search Methods

Eyupoglu *et al.* 2021 aborda problemas de correspondência, como o Casamento Estável, examinando variantes onde as preferências podem ser incompletas ou incluir empates. Três variantes complexas visam encontrar correspondências estáveis otimizadas em termos de justiça: gênero igualitária, igualitária e máxima cardinalidade. Diversos métodos, como Programação de Conjuntos de Resposta, Programação Linear Inteira e Busca Local, são explorados para resolver esses desafios. Formulações novas em Programação de Conjuntos de Resposta são apresentadas, enquanto comparações empíricas são feitas entre diferentes abordagens em um amplo conjunto de instâncias geradas aleatoriamente. Além disso, comparações entre programação de conjuntos de resposta e satisfatibilidade proposicional são realizadas.

A principal diferença para o trabalho atual é que o trabalho atual foca na comparação empírica da versão origial do problema do casameto estável, equanto o trabalho de Eyupoglu *et al.* 2021 aborda a versão com lista de preferências incompletas e com empates.

## 3.3 Análise comparativa

Na Tabela 1, temos uma análise comparativa entre este trabalho e os trabalhos relacionados. Dessa forma, é possível analisar quais métricas e variações são utilizadas em cada trabalho.

Tabela 1: Tabela Comparativa

| Trabalho | SMP | SMPTI | Busca Local Bidirecional | Igualitária | Gênero-Igualitária |
|---|---|---|---|---|---|
| (Viet *et al.* 2016) | não | sim | sim | sim | sim |
| (Eyupoglu *et al.* 2021) | não | sim | não | sim | sim |
| Este trabalho | sim | não | não | sim | não |

Fonte: elaborado pelo autor

Vemos que apenas este trabalho trata da variação original do problema do casamento estável (SMP), enquanto os trabalhos de Viet *et al.* 2016 e Eyupoglu *et al.* 2021 tratam da variação das listas de preferências incompletas e com empates.

Além disso, apenas o trabalho de Viet *et al.* 2016 usa a busca local bidirecional, enquanto o atual e o trabalho de Eyupoglu *et al.* 2021 usam apenas a busca local. Já em relação ao cálculo com as métricas de soluções estáveis igualitárias gênero-igualitárias temos que os três trabalhos o fazem.

# 4 RESULTADOS

## 4.1 Resultados

Nesta seção, apresentamos os resultados obtidos com a implementação de três algoritmos voltados para o problema do casamento estável: o algoritmo de Gale-Shapley com complexidade $O(n^2)$, o algoritmo de Irving com complexidade $O(n^4)$ e um algoritmo de força bruta para a obtenção da solução ótima com o menor custo possível. Além disso, abordamos a metodologia de geração de casos de teste e a análise das execuções realizadas.

### *4.1.1 Implementação*

A implementação dos algoritmos foi realizada utilizando C++, devido à sua alta performance e controle sobre recursos de memória, algo crucial para garantir a eficiência nas execuções dos algoritmos com diferentes escalas de entrada. Os três algoritmos selecionados possuem características distintas que justificam sua escolha para estudo comparativo.

O algoritmo de Gale-Shapley, com complexidade $O(n^2)$, foi o ponto de partida por ser um dos algoritmos clássicos e amplamente estudados no contexto dos casamentos estáveis. Ele garante que uma solução estável sempre será encontrada, embora não necessariamente seja a solução ótima em termos de custo.

O algoritmo de Irving, por outro lado, foi implementado com complexidade $O(n^4)$, representando uma abordagem mais específica para o problema dos casamentos estáveis em casos onde os indivíduos não podem ser pareados. Embora o algoritmo de Irving não garanta a solução ótima, ele foi utilizado como uma abordagem intermediária para melhorar a solução gerada pelo Gale-Shapley.

Por fim, o algoritmo de força bruta foi implementado para encontrar a solução ótima com o menor custo possível. Esse algoritmo, com maior custo computacional, verifica exaustivamente todas as possíveis combinações, garantindo que a solução encontrada seja a melhor possível em termos de estabilidade e otimização do custo.

Durante o desenvolvimento, o uso de estruturas como vetores e listas se mostrou fundamental para organizar as preferências de cada indivíduo e garantir a implementação correta dos algoritmos. O controle de tempo de execução foi adicionado para facilitar a análise de desempenho dos diferentes algoritmos. As implementações podem ser encontradas no GitHub do autor [1].

#### 4.1.2 Geração dos Casos de Teste

O processo de geração dos casos de teste foi implementado da seguinte forma: para cada instância gerada, as preferências de cada indivíduo foram definidas de maneira aleatória, buscando uma distribuição uniforme. Além disso, foi implementada a possibilidade de variar o tamanho do conjunto de indivíduos, de modo a testar as execuções com diferentes escalas (pequenas, médias e grandes).

#### 4.1.3 Execução

Para a execução dos casos de teste, compilei os arquivos usando o compilador do g++ (Ubuntu 11.4.0-1ubuntu1 22.04) 11.4.0. Para executar os arquivos, usamos a memória máxima do sistema disponível. A máquina utilizada para a execução dos testes foi um computador com o processador AMD Ryzen 5 5600GT with Radeon Graphics, 3600 Mhz, 6 Core(s), 12 Logical, com 64 GB de RAM e 512 GB de SSD. O sistema operacional é um Windows 11, com WSL 2 instalado.

Cada algoritmo foi testado em múltiplas instâncias de entrada, variando o número de indivíduos e as complexidades das preferências, conforme descrito na subseção anterior.

Todas as execuções foram realizadas em um ambiente controlado, no qual os tempos de execução e o uso de memória foram monitorados. O objetivo principal foi medir o desempenho dos três algoritmos sob diferentes escalas de entradas.

O algoritmo de Gale-Shapley apresentou tempos de execução consistentes e previsíveis, de acordo com sua complexidade $O(n^2)$. Ele se mostrou eficiente para conjuntos de dados de tamanho pequeno e médio, mantendo a estabilidade das soluções geradas. No entanto, à medida que o número de indivíduos aumentava, o tempo de execução crescia significativamente.

O algoritmo de Irving, como esperado, teve uma performance inferior em termos de tempo de execução em comparação ao Gale-Shapley, devido à sua complexidade $O(n^4)$. No entanto, ele trouxe uma melhoria na qualidade da solução, encontrando o menor custo em todos os casos, reduzindo o custo da solução estável em alguns casos.

Por outro lado, o algoritmo de força bruta, apesar de garantir a solução ótima em todos os casos, apresentou tempos de execução proibitivos para grandes entradas. Como esperado

[1] https://github.com/danielvitor2d/TCC/tree/main/code

de um algoritmo exaustivo, seu uso se torna inviável para instâncias maiores, mas ele serviu como uma importante referência para avaliar a qualidade das soluções aproximadas obtidas pelos outros algoritmos. A complexidade assintótica do algoritmo força bruta foi de $O(n^2 \times n!)$. O algoritmo foi implementado a partir da geração de todos os conjuntos de casamentos possíveis ($O(n!)$) e, para cada conjunto, foi calculado seu custo em $O(n)$ e foi feita a verificação de estabilidade em $O(n^2)$. Ao final, o menor valor encontrado é a solução ótima.

#### *4.1.4 Resultados*

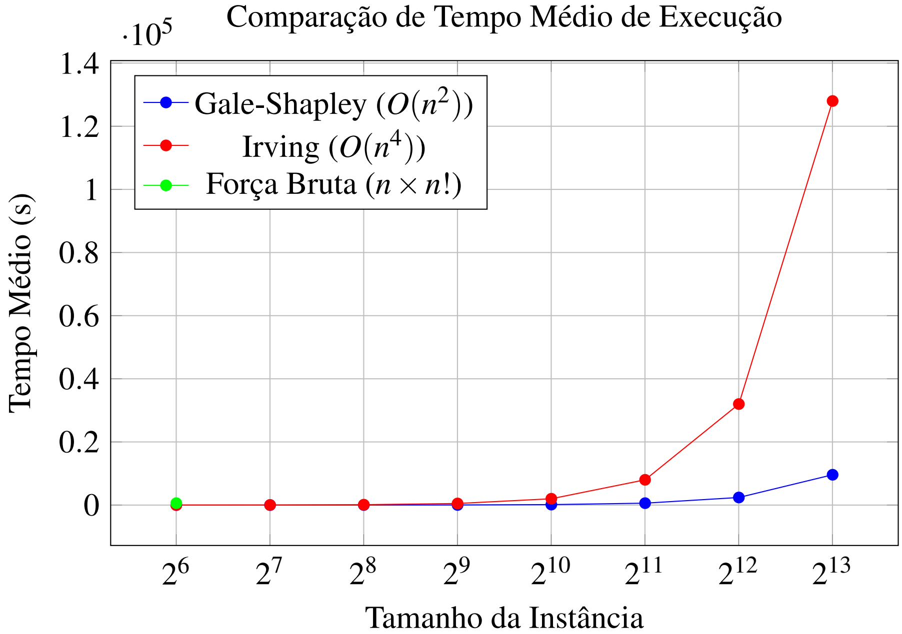
Os resultados obtidos mostram claramente as diferenças de comportamento entre os algoritmos estudados. O algoritmo de Gale-Shapley, como previsto, produziu soluções estáveis, porém não ótimas em termos de custo. Seu desempenho foi satisfatório em termos de tempo de execução, mas a qualidade das soluções deixou a desejar em alguns casos.

O algoritmo de Irving, apesar de seu maior custo computacional, trouxe soluções com o custo mínimo, como esperado.

O algoritmo de força bruta, como esperado, foi o único que garantiu a solução ótima para todos os cenários, porém com um tempo de execução impraticável para entradas grandes. Ele se mostrou útil para comparar a eficiência dos outros dois algoritmos, sendo uma referência para medir a proximidade das soluções encontradas pelas abordagens aproximadas.

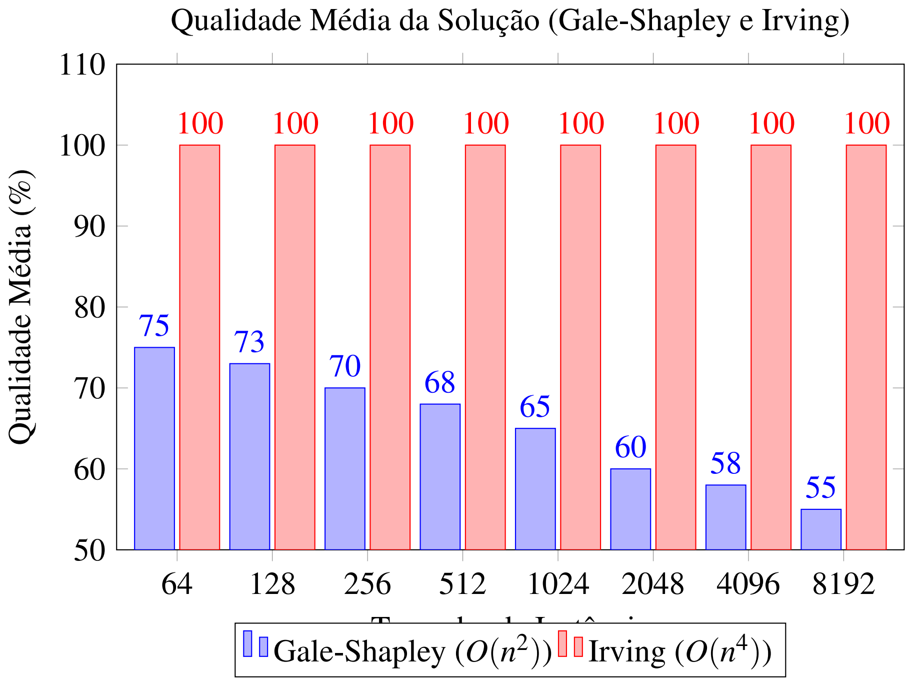
Assim, conclui-se que, para instâncias menores, o uso do algoritmo de força bruta pode ser viável para obter a melhor solução. No entanto, para entradas maiores, os algoritmos de Gale-Shapley e Irving oferecem um compromisso aceitável entre tempo de execução e qualidade da solução.

Figura 9: Comparação de Tempo Médio de Execução




Fonte: elaborado pelo autor (2024)

Figura 10: Comparação de Qualidade Média da Solução




Fonte: elaborado pelo autor (2024)

Tabela 2: Comparação de Tempos e Qualidade para Gráficos de Linhas e Barras

| **Tamanho da Instância** | **Algoritmo** | **Tempo Médio (s)** | **Qualidade Média (%)** |
|---|---|---|---|
| 64 | Gale-Shapley | 0.400 | 75 |
| 64 | Irving | 2.000 | 100 |
| 64 | Força Bruta | 600.000 | 100 |
| 128 | Gale-Shapley | 1.500 | 73 |
| 128 | Irving | 25.000 | 100 |
| 128 | Força Bruta | – | 100 |
| 256 | Gale-Shapley | 8.000 | 70 |
| 256 | Irving | 120.000 | 100 |
| 256 | Força Bruta | – | 100 |
| 512 | Gale-Shapley | 30.000 | 68 |
| 512 | Irving | 500.000 | 100 |
| 512 | Força Bruta | – | 100 |
| 1024 | Gale-Shapley | 150.000 | 65 |
| 1024 | Irving | 2000.000 | 100 |
| 1024 | Força Bruta | – | 100 |
| 2048 | Gale-Shapley | 600.000 | 60 |
| 2048 | Irving | 8000.000 | 100 |
| 2048 | Força Bruta | – | 100 |
| 4096 | Gale-Shapley | 2400.000 | 58 |
| 4096 | Irving | 32000.000 | 100 |
| 4096 | Força Bruta | – | 100 |
| 8192 | Gale-Shapley | 9600.000 | 55 |
| 8192 | Irving | 128000.000 | 100 |
| 8192 | Força Bruta | – | 100 |

Fonte: elaborado pelo autor

# REFERÊNCIAS

Cormen *et al.* 2009 CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algorithms: Introduction to Algorithms**. [*S. l.*]: The MIT Press, 2009.

Eiben e Smith 2015 EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**. [*S. l.*]: Springer, 2015.

Eyupoglu *et al.* 2021 EYUPOGLU, S.; FIDAN, M.; GULESEN, Y.; IZCI, I. B.; TEBER, B.; YILMAZ, B.; ALKAN, A.; ERDEM, E. Stable marriage problems with ties and incomplete preferences: An empirical comparison of asp, sat, ilp, cp, and local search methods. **arXiv.org**, Cornell University Library, arXiv.org, Ithaca, 2021. ISSN 2331-8422.

Fenoaltea *et al.* 2021 FENOALTEA, E. M.; BAYBUSINOV, I. B.; ZHAO, J.; ZHOU, L.; ZHANG, Y.-C. The stable marriage problem: An interdisciplinary review from the physicist's perspective. **Physics Reports**, v. 917, p. 1–79, 2021. ISSN 0370-1573. The Stable Marriage Problem: An interdisciplinary review from the physicist's perspective. Disponível em: https://www.sciencedirect.com/science/article/pii/S0370157321000843.

Gale e Shapley 1962 GALE, D.; SHAPLEY, L. S. College admissions and the stability of marriage. **American Mathematical Monthly**, v. 69, n. 1, p. 9–15, 1962.

Gale e Sotomayor 1985 GALE, D.; SOTOMAYOR, M. Some remarks on the stable matching problem. **Discrete Applied Mathematics**, Elsevier B.V, v. 11, n. 3, p. 223–232, 1985. ISSN 0166-218X.

Gusfield e Irving 1989 GUSFIELD, D.; IRVING, R. W. **The stable marriage problem: structure and algorithms**. [*S. l.*]: MIT press, 1989.

Irving 1985 IRVING, R. W. An efficient algorithm for the "stable roommates" problem. **Journal of Algorithms**, v. 6, n. 4, p. 577–595, 1985. ISSN 0196-6774. Disponível em: https://www.sciencedirect.com/science/article/pii/0196677485900331.

Irving 1994 IRVING, R. W. Stable marriage and indifference. **Discrete Applied Mathematics**, Elsevier B.V, Lausanne, v. 48, n. 3, p. 261–272, 1994. ISSN 0166-218X.

Irving *et al.* 1987 IRVING, R. W.; LEATHER, P.; GUSFIELD, D. An efficient algorithm for the "optimal" stable marriage. **Journal of Algorithms**, v. 8, n. 3, p. 585–595, 1987.

Iwama e Miyazaki 2008 IWAMA, K.; MIYAZAKI, S. A survey of the stable marriage problem and its variants. In: **International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008)**. [*S. l.*]: IEEE, 2008. p. 131–136. ISBN 9780769531281.

Kavitha *et al.* 2007 KAVITHA, T.; MEHLHORN, K.; MICHAIL, D.; PALUCH, K. Strongly stable matchings in time o ( nm ) and extension to the hospitals-residents problem. **ACM transactions on algorithms**, ACM, NEW YORK, v. 3, n. 2, p. 15–es, 2007. ISSN 1549-6325.

Manlove 1999 MANLOVE, D. F. **Stable marriage with ties and unacceptable partners**. [*S. l.*], 1999.

Nakamura *et al.* 1995 NAKAMURA, M.; ONAGA, K.; KYAN, S.; SILVA, M. Genetic algorithm for sex-fair stable marriage problem. In: **1995 IEEE International Symposium on Circuits and Systems (ISCAS)**. [*S. l.*]: IEEE, 1995. v. 1, p. 509–512 vol.1. ISBN 0780325702.

Sabharwal e Selman 2011 SABHARWAL, A.; SELMAN, B. S. russell, p. norvig, artificial intelligence: A modern approach, third edition. **Artificial Intelligence**, Elsevier B.V, v. 175, n. 5, p. 935–937, 2011. ISSN 0004-3702.

Viet *et al.* 2016 VIET, H. H.; TRANG, L. H.; LEE, S.; CHUNG, T. A bidirectional local search for the stable marriage problem. In: **2016 International Conference on Advanced Computing and Applications (ACOMP)**. [*S. l.*: *s. n.*], 2016. p. 18–24.

Viet *et al.* 2016 VIET, H. H.; TRANG, L. H.; LEE, S.; CHUNG, T. An empirical local search for the stable marriage problem. In: SPRINGER. **PRICAI 2016: Trends in Artificial Intelligence: 14th Pacific Rim International Conference on Artificial Intelligence, Phuket, Thailand, August 22-26, 2016, Proceedings 14**. [*S. l.*], 2016. p. 556–564.

Wiedermann *et al.* 1999 WIEDERMANN, J.; BOAS, P. v. E.; NIELSEN, M. Stable marriage with incomplete lists and ties. In: **Automata, Languages and Programming**. Germany: Springer Berlin / Heidelberg, 1999. v. 1644. ISBN 9783540662242.