

TRABALHO PRÁTICO 0:

Problema do casamento estável

Bruno Maciel Peres

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

brunomperes@dcc.ufmg.br

Resumo. *Este relatório descreve uma solução proposta para o problema do casamento estável (Stable Marriage Problem, em inglês), utilizando listas encadeadas na linguagem C. Os principais objetivos deste trabalho são: (1) a prática da linguagem C e (2) a introdução ao estudo de problemas complexos e suas soluções.*

1. INTRODUÇÃO

Neste trabalho, descrevemos e implementamos um algoritmo que utiliza uma heurística para resolver o problema do casamento estável. O SMP, como será referido a partir de então, é um problema conhecido e foco de estudo nas áreas de Matemática e da Ciência da Computação.

O problema em questão, consiste em encontrar *matchings* estáveis para todos os elementos de dois grupos que possuem uma ordem de preferência sobre os elementos do outro grupo. Um *matching* ocorre quando dois elementos de grupos distintos estão ligados por uma relação (casamento) e é considerado estável quando não há, dentre as opções de preferência de cada um dos casados, outra combinação onde ambos preferiam estar casados com outros indivíduos.

Este algoritmo possui aplicabilidade para resolução de problemas de interesse entre dois grupos. Um exemplo é a alocação de estudantes de medicina formandos que preferem determinados hospitais, listando-os em ordem e hospitais que preferem alguns estudantes de medicina, ordenados por suas notas ou área mais deficiente no hospital, por exemplo.

No programa desenvolvido, um *matching* é o casamento entre dois indivíduos, onde todos indivíduos são separados por gênero. Haverão então um grupo de homens e um de mulheres, onde cada homem possui uma lista de preferência sobre as mulheres e cada mulher possui uma lista de preferências sobre os homens.

Para mensurar a qualidade dos casamentos obtidos pelo método heurístico, utilizaremos o índice de satisfabilidade. Esse índice é expresso como a média das posições que os maridos e/ou esposas ocupam na lista de preferências de seus respectivos parceiros. Isso quer dizer que se um homem casa com a 1ª mulher da sua lista de preferências, ele contribui com 1 para o incremento do índice de satisfabilidade, se ele se casa com a 2ª mulher da lista, ela contribui com 2 para o incremento do índice. Quanto menor esse índice, melhor o casamento para esse grupo.

Neste problema, as pessoas são referenciadas através de chaves ou ids, contadas de 1 a n . Logo, n é a quantidade de elementos do conjunto.

A estratégia proposta para a resolução do casamento estável no grupo em questão, é aplicar o algoritmo de Gale Stanley para encontrar um casamento ideal. Mais informações sobre o algoritmo de Gale Stanley podem ser obtidas nas referências sobre o SMP no final deste documento.

2. REFERÊNCIAS RELACIONADAS

Podemos dividir as referências associadas ao problema estudado e à solução proposta dentre os seguintes grupos:

- **Linguagem C:** Todo o trabalho foi implementado em linguagem de programação C. A linguagem C é uma linguagem de propósito geral amplamente utilizada no desenvolvimento de sistemas de alto desempenho, de tempo real, sistemas operacionais, compiladores, dentre outros. O compilador mais utilizado para a linguagem C é o GCC (desenvolvido pelo projeto GNU). Mais informações sobre a linguagem C podem ser encontradas em [?].
- **Projeto e análise de algoritmos:** Algoritmos são procedimentos computacionais capazes de resolver diversos problemas do mundo real. O estudo de algoritmos é essencial para o desenvolvimento de técnicas mais eficientes e eficazes para a solução desses problemas. Mais informações sobre o projeto e análise de algoritmos podem ser encontradas em [?, ?, ?]

2.1. Estruturas de dados

Para operar sobre a heurística utilizou-se um Tipo Abstrato de Dados de lista duplamente encadeada modificado de tal forma que armazenasse informações relevantes ao indivíduo como estado atual (casado ou não) e id do indivíduo do outro grupo com o qual este está casado. A lista de preferências dos indivíduos foi representada utilizando-se essa lista e o grupo de pessoas do mesmo sexo foi agrupado em um vetor de listas alocado dinamicamente de acordo com a quantidade de pessoas nesse grupo. Tal informação é conhecida a priori a partir da leitura do arquivo de entrada antes da execução do algoritmo.

A implementação de listas e algumas operações de lista foram baseadas na implementação do Prof. Fernando V. Paulovich do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP). Mais informações podem ser obtidas nas referências.

A complexidade de espaço deste programa é $O(n^2)$, visto que para cada indivíduo acrescido no tamanho total, o tamanho de todas as listas aumenta em uma unidade.

Abaixo a descrição do Tipo Abstrato de Dado utilizado e seus elementos.

2.1.1. Indivíduo

Armazena informações das listas relacionadas ao indivíduo e possui um apontador para a lista de interesses desse indivíduo.

1: lista
status <i>casado</i> ;
id da pessoa com quem se casou;
apontador para início da lista;
apontador para o fim da lista;

2.1.2. Lista de preferências

Armazena as ids das pessoas com as quais o indivíduo quer se casar ordenados por preferência e encapsula outras estruturas abaixo descritas, como o nó da lista e o Tipo ITEM.

2: Nó da lista
tipo ITEM;
apontador para o elemento seguinte;
apontador para o elemento anterior;

3: Tipo ITEM
índice do nó na lista (1º nó, 2º nó, ...) ;
valor;

2.2. Algoritmos da heurística

A seguir, descrevemos a estrutura de dados e as principais funções da heurística com pseudocódigo. Analisou-se a complexidade dos algoritmos em termos do número de indivíduos por grupo (n).

2.2.1. Método em que os homens propõem

Essa função tenta casar todos os homens, indicados por seu índice i , através da função *casa_homem*, enquanto houverem homens solteiro. Essa função é executada n vezes e em cada iteração chama a função *casa_homem* de complexidade $O(n)$.

4: men_purpose_algorithm(lista de homens, lista de mulheres, n)

```
while homens solteiros > 0 do  
  | casa_homem(i);  
  | i++;  
end
```

2.2.2. Casa homem

Procura um par disponível e em caso de conflito com uma mulher já casada, desempata utilizando a lista de preferências da mulher com dois pretendentes chamando outra função. A complexidade dessa função é $O(n)$.

5: casa_homem(grupo de homens, grupo de mulheres, *i*, quantidade_solteiros)

```
novo nó cursor = primeira opção do homem;  
while homem i está casado do  
  | if mulher i na lista de preferência do homem está casada then  
    | melhor opção = Desempata(maridoatual, pretendente);  
    | if melhor opção = pretendente then  
      | descasa(maridoatual, mulher);  
      | casa com o pretendente corrente;  
    | else  
      | cursor = próximo na lista do homem  
    | end  
  | else  
    | casa o homem corrente com a mulher na lista;  
    | quantidade_solteiros - -;  
  | end  
end
```

2.2.3. Desempata

Retorna o homem que é mais preferido para a mulher, de acordo com a lista de preferências dela. A complexidade assintótica dessa função é $O(n)$, já que no pior caso deve percorrer os n elementos da lista de preferência da mulher.

6: desempata(mulher, idhomem1, idhomem2)

```
Tira daqui;  
while não é o final da lista de preferência da mulher do  
  | if idhomem1 == preferência da mulher then  
    | return idhomem1;  
  | end  
  | if idhomem2 == preferência da mulher then  
    | return idhomem2;  
  | end  
end
```

2.2.4. Cálculo da satisfabilidade

O índice de satisfabilidade é descrito como o índice da posição que o cônjuge ocupa na lista de preferências de seu parceiro(a), i.e., se o homem casou-se com a 1ª na sua lista de preferências, é acrescido 1 ao cálculo do índice dos homens e assim por diante. Ao final, esse número deve ser dividido por n para ter o valor médio do índice. A complexidade dessa função é $O(n^2)$.

7: satisfabilidade(lista grupo1, n)

```
novo nó cursor;
satisf = 0;
for ( $i=1$ ;  $i \leq n$ ;  $i++$ ) do
    cursor = lista[i].inicio;
    satisf++;
    while cursor.item.valor diferente de grupo[i].casado.com_id do
        satisf++;
        cursor = cursor->proximo;
    end
end
return satisf /  $n$ ;
```

3. SOLUÇÃO PROPOSTA

A solução proposta para o SMP, é utilizar o algoritmo de Gale-Stanley, operando sobre o Tipo Abstrato de Dado de lista criado.

O algoritmo se resume em tentar casar todos os homens seguindo suas ordens de preferências. Caso haja algum conflito (um homem deseja casar-se com uma mulher já casada), a lista de preferências dessa mulher é acessada e dentre os dois homens conflitantes, aquele que for de maior interesse para a mulher será casado com essa mulher. O estado de casado do homem rejeitado passa a ser NÃO casado. Na próxima iteração o homem rejeitado tentará ser casado novamente. Quando não houver mais homens não casados, uma solução estável foi obtida.

O algoritmo poderia tentar casar a mulheres, buscando parceiros para elas em algoritmo análogo. Dessa maneira, a busca percorreria as melhores opções das mulheres primeiro e os homens podem decidir apenas entre as interessadas conflitantes. A mulher terá menor índice de satisfabilidade, o que quer dizer que o grupo que propõe neste algoritmo, obtém melhores parceiros.

4. IMPLEMENTAÇÃO

4.1. Código

4.1.1. Arquivos .c

- **main.c:** Arquivo principal do programa, contém as funções de entrada e saída do algoritmo, alocação de memória necessária e chamada da função principal da heurística.
- **heuristica.c:** Define as operações que realizarão o método heurístico e avaliarão a qualidade dos resultados através do índice de satisfabilidade.

- **lista.c:** Define as funções que operam sobre o TAD lista adequado às necessidades do algoritmo.

4.1.2. Arquivos .h

- **heuristica.h:** Descreve o cabeçalho da heurística e das funções que avaliarão a qualidade dos resultados através do índice de satisfabilidade.
- **lista.h:** Define as estruturas de dados e cabeçalhos das funções relacionadas às operações sobre o TAD lista.

4.2. Compilação e execução

O programa deve ser compilado através do compilador GCC através de um *makefile* ou através do seguinte comando em terminal:

```
gcc main.c heuristica.c lista.c -o tp0
```

O comando para a execução do programa é apenas a execução do executável e 1 parâmetro para a execução, o nome do arquivo de entrada. O arquivo de saída é definido na especificação tal que seja *output.txt*. Caso não haja ao menos 1 argumento, o programa encerra a execução.

```
./tp0 input.txt
```

4.2.1. Formato da entrada

O arquivo de entrada cujo nome é passado como argumento para o executável, contém uma linha que define a quantidade de instâncias que este arquivo possui, ou seja, quantas entradas diferentes estão contidas nesse arquivo que deverão ser executadas por esse algoritmo. Na segunda linha, há o número n de indivíduos em cada conjunto, que será, por consequência, a quantidade de elementos contidos naquela linha, já que a lista de preferência percorre todos os elementos do outro conjunto.

Um exemplo de arquivo de entrada é dado a seguir:

```
1
4
2 4 1 3
4 1 2 3
2 3 4 1
2 3 4 1
2 3 4 1
3 4 2 1
3 2 1 4
3 2 4 1
```

4.2.2. Formato da saída

A saída do programa, armazenada em um arquivo de saída *output.txt*, contém informações sobre os n casais formados e a qualidade dos casamentos obtidos através do índice de

satisfabilidade. Para cada casamento, é impresso as ids dos homens à esquerda e a id de suas respectivas esposas à direita. Após as ids dos conjuges é apresentado o índice de satisfabilidade geral, a satisfabilidade masculina e a satisfabilidade feminina com precisão de 3 casas decimais. Para o caso de várias instâncias, a sequência se repete abaixo da anterior. Um exemplo de saída é dado abaixo:

```
2 4
1 1
3 2
4 3
2.250
1.750
2.750
```

5. CONCLUSÃO

Neste trabalho foi implementado o algoritmo de Gale Stanley que opera sobre um TAD de listas encadeadas adaptado para o problema.

Dentre as adversidades encontradas durante o desenvolvimento desse algoritmo, a maior delas foi ter desenvolvido um algoritmo recursivo que casava obrigatoriamente o homem i atual. Caso a mulher esteja casada, o algoritmo percorreria todas as preferências do homem rejeitado até que ele se case. Quando a iteração percorresse todos os homens o programa é terminado, pois todos os casais estarão formados. Essa implementação decorreu de uma falha na interpretação do algoritmo de Gale Stanley e não foi utilizado pela complexidade em provar que este método também resultava em uma solução estável.

O trabalho atingiu seus principais objetivos: a prática da linguagem de programação C e o estudo de problemas complexos. Além dos objetivos propostos, este trabalho permitiu a obtenção de outros conhecimentos, importantes na área de Ciência da Computação, como a utilização de \LaTeX para escrever o relatório técnico nos padrões da Sociedade Brasileira de Computação e utilização de um arquivo *makefile* para compilação e execução do programa.

Algumas melhorias que poderiam ser consideradas neste trabalho são:

- Utilização de lista simplesmente encadeada.
- Armazenamento do índice do nó do cônjuge na lista, permitindo o cálculo da satisfabilidade em $O(n)$.

Referências

- Paulovich, Fernando V. *Listas Duplamente Ligadas* http://www.lcad.icmc.usp.br/~paulovic/aulas/ED-I/SCC0202-aula-05-Listas_duplamente_ligadas.pdf.
- Department of Information and Computing Sciences, Faculty of Science, Utrecht University http://www.cs.uu.nl/docs/vakken/an/handouts/2011-03-03_an-stablemarriage.pdf.
- University of California - Berkeley <http://mathsite.math.berkeley.edu/smp/smp.html>.
- Cormen, T. (2001). *Introduction to algorithms*. MIT press.
- Kernighan, B., Ritchie, D., and Eklint, P. (1988). *The C programming language*. Prentice-Hall Englewood Cliffs, NJ.

Sedgewick, R. (2001). Algorithms in C.

Ziviani, N. (2007). *Projeto de Algoritmos com implementações em Pascal e C*. Pioneira Thomson Learning.