

Algoritmos e Estruturas de Dados 3

Trabalho Prático 0

Similaridade de Textos

André Taiar Marinho Oliveira

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

taiar@dcc.ufmg.br

Resumo. *Recuperação de Informação (RI) é uma área da computação que lida com o armazenamento de documentos e a recuperação automática de informação associada a eles. É uma ciência de pesquisa sobre busca por informações em documentos, busca pelos documentos propriamente ditos, busca por metadados que descrevam documentos e busca em banco de dados, sejam eles relacionais e isolados ou banco de dados interligados em rede de hipermídia, tais como a World Wide Web.*

Uma boa forma de classificar, definir e reconhecer semelhanças entre textos é definindo o seu assunto de acordo com as suas palavras chaves. Com o crescimento da utilização de sistemas de Recuperação de Informação e sua utilização massiva na atualidade, o problema de se descobrir palavras-chave de um texto tem sido muito explorado.

1. Introdução

Neste trabalho, ns descrevemos, implementamos e analisamos um controlador de elevador. Para fins de anlise, foi implementado um simulador orientado a eventos para a gerao de requisies de transporte. Os principais objetivos do trabalho so o exercicio da linguagem C, a discussso sobre problemas e complexos e sua soluo. O problema especifico a ser estudado neste trabalho a implementao de um simulador para a gerao de chamadas de passageiros de um elevador e a avaliao de uma estratgia de controle para o atendimento das chamadas. A estratgia de controle do elevador deve atender as requisies obedecendo restrio de capacidade do elevador (lotao mxima). A contagem de tempo realizada atravs de uma unidade bsica, denominada Jepslon. Um tempo total de 1 Jepslon necessrio para que o elevador percorra a distncia entre dois andares ou para que ele realize uma operao de carga ou descarga de passageiros. A estratgia de controle proposta neste trabalho similar aquela utilizada pela maioria dos elevadores convencionais. Em cada instante de tempo, o elevador pode estar percorrendo os andares nos sentidos para cima ou para baixo. Ao passar por cada andar, o elevador verifica se existem requisies de descarga de passageiros para o andar corrente ou requisies de carga de passageiros que desejam seguir o sentido atual do elevador. No caso de requisies de carga, verificada a restrio de capacidade do elevador. O simulador desenvolvido, que gera requisies para o elevador, simples. Dado um conjunto de requisies, que representam passageiros com um andar de origem e destino, alm de um instante de gerao da requisio (momento em que o passageiro aperta o boto chamando o elevador), o simulador gera cada uma das requisies no tempo correto. Essas requisies so carregadas na memria do elevador. Uma anlise experimental do tempo

de execucao do simulador e do tempo mdio de espera dos usuarios foi realizada. Para isso, foi necessaria a criao de um gerador de requisies. A modelagem do tempo de simulao em termos das caractersticas do problema no uma tarefa simples, j que ela precisa considerar fatores como a distribuio dos trajetos e distribuio dos tempos entre as requisies. Para ilustrar o efeito desses fatores, ns avaliamos como a popularidade dos trajetos afeta o elevador. O restante deste relatrio organizado da seguinte forma. A Seo 2 discute alguns temas relacionados ao trabalho. A Seo 3 descreve o simulador, o elevador e a estratgia de controle proposta. A Seo 4 trata de detalhes especficos da implementao do trabalho. A Seo 5 contm a avaliao experimental do controlador de elevador proposto. A Seo 6 conclui o trabalho.

2. REFERNCIAS RELACIONADAS

Podemos dividir as referncias associadas ao problema estudado e soluo proposta dentre os seguintes grupos:

- **Escalonamento, anlise de desempenho:** Um grande nmero de problemas em computao pode ser modelado como um conjunto de requisies um dado servio. Nesse caso, a estratgia de atendimento dessas requisies pode afetar o desempenho do servio significativamente. Alm disso, tambm importante entender as caractersticas da entrada (carga), j que elas determinam diversas caractersticas do servio. Mais informaes sobre o assunto podem ser encontradas em [?].
- **Linguagem C:** Todo o trabalho foi implementado em linguagem de programao C. A linguagem C uma linguagem de propsito 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 o GCC (desenvolvido pelo projeto GNU). Mais informaes sobre a linguagem C podem ser encontradas em [?].
- **Projeto e anlise de algoritmos:** Algoritmos so procedimentos computacionais capazes de resolver diversos problemas do mundo real. O problema de controlar um elevador, estudado neste trabalho, um dos exemplos de como algoritmos esto presentes no nosso dia-a-dia. O estudo de algoritmos essencial para o desenvolvimento de tcnicas mais eficientes e eficazes para a soluo desses problemas. Mais informaes sobre o projeto e anlise de algoritmos podem ser encontradas em [?, ?, ?]

3. SOLUO PROPOSTA

A soluo proposta para a simulao discreta do funcionamento de um elevador utiliza um conjunto de estruturas de dados que contm informaes sobre o estado da simulao, do elevador, registros de requisies, dentre outros. Essas estruturas so manipuladas atravs de um conjunto de algoritmos que descrevem o funcionamento da soluo.

O simulador implementado tem como base um temporizador contnuo, ou seja, cada instante de simulao corresponde uma iterao do simulador. Uma outra opo seria utilizar como base a contagem das requisies. Os critrios que levaram escolha da iterao do simulador em cada instante so: (1) a simplicidade de implementao, (2) mapeamento direto entre tempo de execucao e tempo de simulao e (3) o custo computacional reduzido de avaliar cada instante de simulao, supondo-se que as requisies sejam bem distribudas ao longo do tempo.

O controle do elevador baseado no que chamamos de *requisio corrente*, que a requisio que define o sentido do elevador, e sempre a requisio do andar mais alto quando o elevador est subindo ou a requisio do andar mais baixo quando o elevador est descendo. Atravs da identificao de uma requisio corrente possvel determinar o trajeto do elevador ao longo do tempo. Durante o percurso na direo de atender uma requisio corrente, outras requisies podem ser atendidas, sempre considerando as restries de espao do elevador.

A seguir, ns descrevemos as principais estruturas de dados e algoritmos propostos. Ns analisamos a complexidade dos algoritmos em termos do nmero de requisies (n), do nmero de andares (a) e da capacidade do elevador (c). A complexidade de espao ser determinada com base nas estruturas de dados e na anlise do algoritmo principal, que determina a complexidade da soluo.

3.1. Estruturas de dados

3.1.1. Requisio:

Armazena informaes sobre uma requisio de transporte para o elevador.

1: Requisicao
identificador;
hora requisicao;
andar atual;
andar desejado;

3.1.2. Simulador

Armazena informaes para a simulao, ou seja, uma fila de requisies de transporte a ser tratada pelo elevador. A complexidade de espao do armazenamento das requisies $O(n)$, onde n o nmero de requisies.

2: Simulador
numero de requisicoes;
requisicoes;

3.1.3. Registro de requisio

Armazena um registro de requisio, que uma unidade da memria de requisies a serem atendidas pelo elevador. Possui, alm da requisio, um tipo, que a define se a operao necessria de carga ou descarga de passageiro.

3: Registro de requisicao
requisicao;
tipo;

3.1.4. Elevador

Armazena informaes sobre o elevador, como seu estado e algumas restries definidas atravs do ambiente de execuo. Contm uma lista de registros que representa a memria de requisies do elevador. A complexidade de espao do armazenamento dos registros $O(n)$, onde n o nmero de requisies.

4: Elevador

andar atual;
lotacao;
capacidade;
numero de andares;
sentido;
numero de registros;
registros;

3.2. Algoritmos

3.2.1. Atende requisio de carga

Realiza o atendimento de uma requisio de carga para o elevador. No caso da requisio atendida ser a corrente, o elevador altera o sentido de seu movimento. Cada requisio de carga est sujeita limitao de espao (lotao mxima) do elevador. Alm disso, uma requisio de carga s atendida quando o elevador se movimenta no sentido do andar de destino da requisio. Caso contrrio, a requisio deve aguardar a passagem do elevador no futuro. A complexidade desse algoritmo em termos de tempo de execuo $O(n)$, onde n o nmero de requisies, devido ao custo de identificao da requisio corrente, realizado atravs de uma busca linear.

5: atende-requisicao-carga(elevador, registro requisicao)

if *requisicao corrente* **then**
| Inverte sentido elevador;
end
Atualiza lotacao elevador;
Gera requisicao de descarga;

3.2.2. Atende requisio de descarga

Similar ao atendimento de uma requisio de carga, apesar de no estar sujeita restries de capacidade do elevador. A complexidade desse algoritmo em termos de tempo de execuo tambm $O(n)$.

3.2.3. Opera elevador

Controla a movimentao do elevador ao longo do tempo. Caso haja alguma requisio a ser atendida em um dado momento, ou seja requisies para o andar atual, aquele instante

6: atende-requisicao-descarga(elevador, registro requisicao)

```
if requisicao corrente then
|   Inverte sentido elevador;
end
Atualiza lotacao do elevador;
gera relatorio de atendimento da requisicao;
apaga registro da memoria do elevador;
```

ser utilizado para o atendimento dessas requisies. Caso contrrio, identificada a requisio corrente e o elevador movimentado no sentido dessa requisio. A complexidade desse algoritmo em termos de tempo de execucao $O(n^2)$, onde n o nmero de requisies, que corresponde ao custo de atender toda a fila de requisies, sendo que o atendimento de cada requisio tem custo $O(n)$.

7: opera-elevador(elevador, tempo)

```
if Ha requisicoes para o andar then
|   Atende requisicoes descarga;
|   Atende requisicoes carga;
else
|   movimenta elevador no sentido da requisicao corrente;
end
```

3.2.4. Simula elevador

Determina o funcionamento da simulao. Para cada instante de tempo, em que haja uma requisio (na fila do simulador ou na memria do elevador), o elevador operado (segundo a funo opera-elevador), e as requisies de carga para aquele dado instante so armazenadas na memria do elevador, segundo a fila de requisies do simulador. Determinar a complexidade da simulao no uma tarefa trivial, j que depende do tempo total de simulao (t). Devido ao custo de operao do elevador (funo *opera – elevador*), sabemos que a complexidade de tempo do algoritmo $O(n^2 \times t)$. Alm disso, t inversamente proporcional capacidade do elevador (c) e diretamente proporcional ao nmero de andares (a) e ao nmero de requisies (n). No entanto, a relao entre t , c , a e n afetada por:

- A popularidade dos trajetos no elevador: o que afeta o tempo de transporte, se caminhos mais longos tendem a ser mais frequentes o tempo de simulao aumenta.
- O tempo entre as requisies: se as requisies forem muito separadas no tempo, a simulao tende a ser longa, alm disso, isso pode reduzir a taxa de utilizao do elevador, atravs da frequencia de trajetos em que elevador esteja quase vazio.

Acreditamos que a modelagem do problema considerando diferentes distribuies de popularidade e tempos entre requisies no est no escopo deste trabalho. A complexidade de espao do algoritmo $O(n)$, onde n o nmero que requisies do elevador, j que necessario armazenamento de cada requisio na fila do simulador e na memria do elevador.

8: simula(sequencia requisicoes, ambiente)

```
tempo = 0;
while Ha requisicoes do
    opera-elevador(elevador,tempo);
    Registra requisicoes na memoria do elevador de acordo com o simulador;
    tempo++;
end
```

4. IMPLEMENTAO

4.1. Cdigo

4.1.1. Arquivos .c

- **principal.c:** Arquivo principal do programa que implementa o simulador do elevador.
- **simulador.c:** Define as funes relacionadas simulao do elevador.
- **lista.c:** Define funes relacionadas a manipulao de um TAD lista implementada atravs de apontadores.
- **fila.c:** Define as estruturas de dados e cabealhos de funes relacionadas a manipulao de um TAD fila implementada atravs de apontadores.

4.1.2. Arquivos .h

- **simulador.h:** Define as estruturas de dados e cabealhos de funes relacionadas a simulao do elevador
- **lista.h:** Define as estruturas de dados e cabealhos de funes relacionadas manipulao de um TAD lista implementada atravs de apontadores.
- **fila.h:** Define as estruturas de dados e cabealhos de funes relacionadas a manipulao de um TAD fila implementada atravs de apontadores.

4.2. Compilao

O programa deve ser compilado atravs do compilador GCC atravs de um makefile ou do seguinte comando:

```
gcc principal.c simulador.c fila.c lista.c -o tp0
```

4.3. Execuo

A execuo do programa tem como parmetros:

- Um arquivo de descrio do ambiente.
- Um arquivo de requisies.
- Um arquivo de sada.

O comando para a execuo do programa da forma:

```
./tp0 -a <arquivo de ambiente> -r <arquivo de requisicoes> -s <arquivo de saida>
```

4.3.1. Formato da entrada

O arquivo de descrição do ambiente contém apenas uma linha onde são definidos o número de andares do prédio atendido pelo elevador e a capacidade de elevador, separados por espaço. Já o arquivo de requisições contém o número de requisições na primeira linha seguido de cada uma das requisições.

Cada requisição contém um identificador, o instante de chamada do elevador (que tem como base um temporizador sequencial), o andar de origem do passageiro e o andar desejado. Um exemplo de arquivo de requisições dado a seguir:

```
3
0 0 0 10
1 3 0 3
2 5 13 15
```

4.3.2. Formato da saída

A saída do programa, armazenada em um arquivo de saída, contém informações sobre todas as requisições atendidas. Para cada requisição as informações de interesse são o identificador da requisição, o tempo de espera de carga (até que a pessoa entre no elevador) e o tempo de espera dentro do elevador. A seguir, a saída correspondente entrada apresentada na seção anterior.

```
id :0
espera carga :1
espera elevador :11

id :1
espera carga :27
espera elevador :4

id :2
espera carga :11
espera elevador :31
```

5. AVALIAÇÃO EXPERIMENTAL

Nesta seção avaliamos a estratégia de controle de elevador proposta e o simulador implementado em termos do tempo de execução (em segundos) e do tempo total de espera dos usuários (em Jepsions) para diferentes números de andares, capacidades do elevador, números de requisições e popularidades de trajetos. O tempo total de espera é a soma dos tempos de espera para entrar no elevador e para chegar ao andar desejado. Os experimentos foram executados em um computador pessoal com sistema operacional Suse Linux, processador AMD Athlon 3500+ 64 bits e 1GB de memória principal.

Como destacado anteriormente, o tempo de execução provém, além do tempo para que toda a simulação seja realizada, uma aproximação sobre o quanto demorado o atendimento de todas as requisições do elevador (já que cada instante de tempo simulado em uma iteração). Já o tempo de espera pelo elevador é uma maneira simples de avaliar a qualidade do serviço prestado. Outros critérios como utilização do elevador, distância total percorrida pelo elevador,

dentre outros, poderiam ser avaliados. Alm disso, tambm seria interessante considerar como o tempo entre as requisies afeta o elevador, no entanto, devido restries de tempo e espao, no avaliaremos todos os parmetros da simulao.

As entradas para a simulao do elevador foram geradas a partir de um gerador sinttico de requisies. Esse gerador recebe como parmetros: (1) o nmero de andares, (2) a capacidade do elevador, (3) um atributo que determina a popularidade dos trajetos e (4) o nmero de requisies. Como sada, o gerador cria um arquivo de ambiente e um arquivo de requisies. O atributo que determina a popularidade dos trajetos pode ser: (1) escolha aleatria, ou seja, trajetos escolhidos aleatoriamente, (2) probabilidade do trajeto proporcional ao tamanho, ou seja, trajetos mais longos so mais frequentes, e (3) probabilidade do trajeto inversamente proporcional ao tamanho do trajeto, o que resulta em maior frequncia de trajetos curtos.

Em todos os experimentos ns variamos um parmetro e mantivemos os outros constantes. Cada experimento foi executado 5 vezes e foram tomados os valores mdios. Nosso objetivo reduzir o impacto de propriedades de uma entrada especfica produzida pelo gerador de entradas. Alm do tempo de execuo, foi medido tambm o tempo de usurio, que o tempo gasto pelo processo do programa. O tempo de sistema, utilizado na realizao de tarefas no nvel do kernel, tambm foi medido, mas no mostrado nos resultados pois muito menor que os tempos de usurio e execuo. O tempo mdio entre as requisies do elevador foi mantido constante, igual a 3 Jepslons.

A Tabela 1 mostra como o aumento do nmero de andares afeta os tempos de execuo, usurio e espera dos usurios. O nmero de andares varia de 25 a 40. O nmero de requisies 50000, a capacidade do elevador de 10 usurios e a escolha dos trajetos aleatria. importante notarmos que, ao aumentarmos o nmero de andares, o tamanho mdio dos trajetos tende a aumentar. O nmero de andares afeta significativamente os tempos de execuo, usurio e espera. Experimentos com nmero de andares maior que 40 requerem tempos de execuo muito longos. Nossa hiptese de que esse impacto se deve ao aumento dos trajetos pode ser explicada pelo grande aumento do tempo de espera, ou seja, os usurios esto permanecendo no elevador por um longo perodo de tempo.

#andares	Tempo execuo(seg.)	Tempo usurio(seg.)	Tempo de espera (Jepslons)
25	2,03	1,85	56,19
30	3,00	2,77	76,13
35	6,96	6,69	140,37
40	2068,28	1952,83	1947,29

Tabela 1. Tempo de simulao e de espera para diferentes nmeros de andares

A Tabela 2 mostra impacto da capacidade do elevador sobre o tempo de simulao e o tempo de espera. A capacidade varia de 10 a 13 lugares. O nmero de andares 40, o nmero de requisies 50000 e a escolha dos trajetos aleatria. Como era esperado, o tempo de simulao e o tempo de espera so inversamente proporcionais capacidade do elevador.

A Figura 1 avalia o tempos de simulao e de espera em termos do nmero de requisies. O nmero de requisies varia de 30000 a 50000. O nmero de andares 40, a capacidade do elevador 10 e a escolha dos trajetos aleatria. Como indicado na anlise de complexidade, o custo computacional do algoritmo polinomial em relao ao nmero de requisies. O crescimento do tempo de execuo se aproxima de uma funo quadrtica, tambm de acordo

capacidade	Tempo execu(seg.)	Tempo usurio(seg.)	Tempo de espera (Jepslons)
10	2068,28	1952,83	1947,29
11	17,63	17,16	234,37
12	5,43	5,20	121,51
13	4,31	4,07	101,04

Tabela 2. Tempo de simulao e de espera para diferentes capacidades do elevador

com a anlise de complexidade. J o tempo de espera aumenta linearmente com o nmero de requisies, uma hiptese para esse resultado que quanto maior o nmero de requisies, mais tempo necessrio para que o simulador gere todas as requisies para o elevador. A partir do momento que todas as requisies foram geradas, a tendncia que a fila de requisies se reduza gradativamente. No entanto, um longo perodo com gerao constante de requisies pelo simulador congestionam a fila de registros de requisio do elevador, aumentando o tempo mdio de espera.

(a)(b)

Figura 1. Tempo de simulao e de espera para diferentes nmeros de requisies

Como ltima parte de nossa avaliao experimental, ns analisamos como a popularidade dos trajetos afeta os tempos de execuo e espera. Trs relaes entre as frequncias dos trajetos e seus tamanhos foram consideradas: (1) trajetos escolhidos aleatoriamente, (2) probabilidade do trajeto diretamente proporcional sua distncia e (3) probabilidade do trajeto inversamente proporcional sua distncia. O nmero de andares 40, a capacidade do elevador 10 e o nmero de requisies 15000. A Tabela 3 mostra os resultados obtidos. Podemos notar que a distribuio das requisies afeta significativamente os tempos de execuo e espera, principalmente o tempo de execuo. Quando trajetos longos se tornam mais frequentes, o tempo de execuo cresce, no somente devido ao aumento do nmero de iteraes de simulador, mas tambm do custo da manipulao de um grande nmero de requisies. J o tempo de espera, menos afetado, aumenta devido uma maior carga de trabalho para o elevador.

Frequencia tamanho dos trajetos	Tempo de execu(seg.)	Tempo de espera(seg.)
Aleatria	73,52	778,48
Proporcional distncia	9.011,42	7.011,46
Inv. Proporcional distncia	2,24	100,61

Tabela 3. Tempos de simulao e de espera para diferentes frequncias de tamanhos de trajeto

6. CONCLUSO

Neste trabalho ns descrevemos um simulador discreto de eventos e uma estratgia de controle de um elevador. O simulador gera requisies a serem atendidas ao longo do tempo de simulao e o elevador atende essas requisies de acordo com uma estratgia similar utilizada pelos elevadores convencionais.

O trabalho atingiu seus principais objetivos: a prática da linguagem de programação C e o estudo de problemas complexos. O autor do trabalho já havia implementado alguns programas nessa linguagem e, portanto, não teve grandes problemas na implementação do trabalho. Já a solução do problema demandou a modelagem de diversos cenários, de acordo com diferentes estados do elevador. Uma importante decisão foi a simulação de cada Jépslon e não de instantes específicos associados a eventos, o que facilitou o projeto da solução.

A análise de complexidade da solução proposta não foi uma tarefa simples, já que o custo computacional da solução está sujeito a diversas características da entrada. Por outro lado, o impacto de diferentes fatores associados à entrada pode ser avaliado através de diversos experimentos realizados.

Algumas melhorias que poderiam ser consideradas neste trabalho são:

- Uma modelagem mais complexa do tempo de simulação em termos da popularidade dos trajetos, do tempo entre as requisições, dentre outros;
- A análise do comportamento do elevador considerando características como a distância percorrida e a taxa de utilização do elevador.