

UNIVERSIDADE FEDERAL DO PARANÁ

GUSTAVO VALENTE NUNES

MODELAGEM DE UM PROBLEMA DE ENVIO ORDENADO UTILIZANDO  
PROGRAMAÇÃO LINEAR

CURITIBA PR

2022

GUSTAVO VALENTE NUNES

MODELAGEM DE UM PROBLEMA DE ENVIO ORDENADO UTILIZANDO  
PROGRAMAÇÃO LINEAR

Trabalho apresentado como requisito parcial à conclusão da disciplina de Otimização no Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

CURITIBA PR

2022

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>3</b>
1.1	LIMITAÇÕES E DETALHES . . . . .	3
1.2	VARIÁVEIS DO PROBLEMA. . . . .	3
<b>2</b>	<b>ALGUNS EXEMPLOS . . . . .</b>	<b>5</b>
2.1	GUIAS DE L <sup>A</sup> T <sub>E</sub> X . . . . .	5
2.2	ESTRUTURA DO TEXTO. . . . .	5
2.3	ESTILO DE REDAÇÃO . . . . .	5
2.4	ALGUNS EXEMPLOS. . . . .	5
2.4.1	Exemplo de figura. . . . .	5
2.4.2	Exemplo de tabela. . . . .	6
2.4.3	Exemplo de fórmula. . . . .	7
2.4.4	Exemplos de código-fonte . . . . .	7
2.4.5	Exemplo de algoritmo. . . . .	7
2.4.6	Exemplo de citação . . . . .	8
2.5	UMA SEÇÃO. . . . .	8
2.5.1	Uma subseção. . . . .	8
2.6	CONCLUSÃO . . . . .	8
	<b>REFERÊNCIAS . . . . .</b>	<b>9</b>

# 1 INTRODUÇÃO

O problema sugerido pelo orientador consiste em modelar e implementar uma solução para o problema de envio ordenado. Existe uma certa quantidade de produtos disponíveis e esses produtos precisam ser enviados em uma quantidade finita de caminhões. O objetivo do trabalho, é encontrar uma solução que minimize a quantidade de caminhões necessários para realizar o transporte de todas as mercadorias. Os itens serão que transportados, podem ser separados em pedaços menores de forma que seja possível separar os pedaços em caminhões diferentes.

## 1.1 LIMITAÇÕES E DETALHES

Antes de começar a análise e modelagem do problema, vou definir alguns detalhes, limitações e problemas encontrados.

- O custo mínimo é calculado sobre a quantidade minima de viagens necessárias, de forma que eu consiga fazer o transporte de todos os produtos.
- Cada caminhão possui um limite de peso que ele consegue carregar. E esse peso precisa ser respeitado pelos itens que estarão dentro desse caminhão.
- Os caminhões podem carregar diferentes itens, contanto que o peso seja respeitado.
- Na modelagem do problema, não foi considerado os pares ordenados. Isso porque não foi possível chegar em uma forma de ter pares ordenados utilizando programação linear.
- Se necessário, um item pode ser separado em diferentes caminhões.

## 1.2 VARIÁVEIS DO PROBLEMA

Nessa sessão é apresentado as váriaveis e restrições que foram realizados na usados na modelagem do problema.

- $n$  = Quantidade de itens que terão que ser transportados.
- $C$  = Capacidade total possível para cada caminhão.

- $x_i$   
=  
 $x_i$   
representa o item.

- $x_{i,j}$   
=  
 $x_{i,j}$   
representa o item  $i$  no caminhão  $j$ .

- *defesa*: é gerado um documento em espaço 1,5, frente simples e sem as páginas iniciais adicionais; é uma versão adequada para receber as anotações dos membros da banca de defesa.
- *final*: é gerado um documento em espaço simples, frente/verso, com páginas iniciais (capa, ficha catalográfica, folha de aprovação, agradecimentos, etc). É uma versão bem mais compacta, mais ecológica e ideal para a impressão definitiva.

Para obter os melhores resultados, compile este modelo usando a seguinte sequência de passos:

```
pdflatex main          // compilação inicial
bibtex main            // processa referências bibliográficas
pdflatex main          // compilação final
```

ou

```
make                  // faz tudo...
```

Os principais itens considerados na formatação deste documento foram:

- Papel em formato A4, com margens de 20 mm à direita e embaixo, 30 mm nos demais lados. Não devem ser usados cabeçalhos ou rodapés além dos que estão aqui propostos.
- O texto principal do documento escrito em 12 pontos. O fonte principal do texto pode ser selecionado no arquivo `packages.tex`.
- Código-fonte, listagens e textos similares são formatados em fonte Courier 12 ou 10 pontos.
- O espaçamento padrão entre linhas é 1,5 linhas (1 linha na versão final). Não inserir espaços adicionais entre parágrafos normais. Figuras, tabelas, listagens e listas de itens devem ter um espaço adicional antes e após os mesmos.
- As páginas iniciais não são numeradas.
- O corpo do texto é numerado com algarismos arábicos (1, 2, 3, ...) a partir da introdução, até o final do documento. Os números de página devem estar situados no alto à direita (páginas direitas) ou à esquerda (páginas esquerdas).
- Expressões em inglês, grego, latim ou outras línguas devem ser enfatizadas em itálico, como *sui generis* ou *scheduling* (use o comando `\emph{...}`).
- Para reforçar algo, deve-se usar somente **negrito**. Sublinhado ou MAIÚSCULAS não devem ser usados como forma de ênfase!
- As notas de rodapé também têm um modelo<sup>1</sup>. Notas de rodapé servem para fazer algum comentário paralelo; não as use para colocar URLs, referências bibliográficas ou significado de siglas.

Felizmente o L<sup>A</sup>T<sub>E</sub>X resolve a maior parte dessas questões!

---

<sup>1</sup>As notas de rodapé devem ser escritas em tamanho 10 pt, numeradas em arábico.

## 2 ALGUNS EXEMPLOS

### 2.1 GUIAS DE L<sup>A</sup>T<sub>E</sub>X

Este modelo contém exemplos para os padrões de inserção de figuras, tabelas, listas de itens, bibliografia, etc. Em caso de dúvidas ou discordância, Pode-se entrar em contato com a direção ou secretaria do programa. Obviamente, críticas (construtivas) e sugestões são muito bem-vindas.

Para aprender a usar L<sup>A</sup>T<sub>E</sub>X, um bom guia introdutório disponível na Internet é (Oetiker et al., 2007), que também tem uma versão em português. Para tópicos mais avançados consulte (Goossens et al., 1993).

### 2.2 ESTRUTURA DO TEXTO

Para melhorar a legibilidade do texto, deve ser evitado o uso de subdivisões mais profundas que a subseção (por exemplo, subsubseções). Se elas forem absolutamente necessárias, não devem ser numeradas. Deve-se analisar a possibilidade de uso de uma lista de itens em seu lugar. O número de níveis de texto do documento não deve exceder três: capítulo, seção e subseção. O uso de mais que três níveis dificulta a leitura e prejudica muito a estética do texto.

### 2.3 ESTILO DE REDAÇÃO

Ao elaborar o texto da dissertação ou da tese, o mais indicado é o uso do verbo na forma impessoal. Exemplos:

- ... utilizaram-se os seguintes dados ...
- ... elaborou-se de forma precisa ...
- ... trata-se os algoritmos ...
- ... foram obtidos resultados significativos ...

Além disso, deve-se a todo custo evitar a “linguagem de revista”, com expressões como “sensacional”, “impressionante”, “monstruoso”, etc (por exemplo: “Os resultados obtidos são sensacionais, sobretudo considerando a monstruosa margem de erro.”).

### 2.4 ALGUNS EXEMPLOS

Esta seção traz alguns exemplos de elementos típicos de um texto científico, como figuras, tabelas e fórmulas matemáticas.

#### 2.4.1 Exemplo de figura

A forma sugerida para incluir figuras em um documento L<sup>A</sup>T<sub>E</sub>X é importá-las usando o pacote `graphicx`. Como formatos gráficos sugere-se:

- Formatos *raster*, como PNG (*Portable Network Graphics*) ou JPG (*Joint Photographic Experts Group*) para fotografias; procure usar uma resolução de ao menos 150 dpi (*dots per inch*).
- Formatos *vetoriais*, como PDF (*Portable Document Format*) ou EPS (*Extended PostScript*) para diagramas e gráficos<sup>1</sup>.

A maior parte das ferramentas permite exportar figuras nesses formatos (a figura do exemplo foi produzida com o *Inkscape*, um programa livre multiplataforma). A figura 2.1 mostra um exemplo de inclusão de figura em PDF.

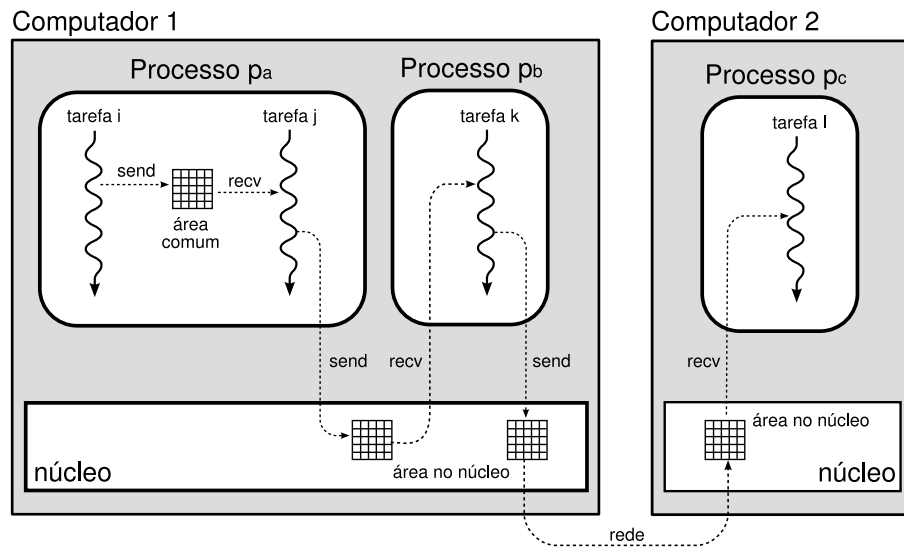


Figura 2.1: Comunicação inter-processos.

Para mais informações consulte (Goossens et al., 1993).

#### 2.4.2 Exemplo de tabela

Tabelas são elementos importantes de um documento. No  $\text{\LaTeX}$  as tabelas podem ser objetos flutuantes (definidas no ambiente `table` e referenciadas por números usando `label` e `ref`) ou objetos fixos simples, criados pelo ambiente `tabular`. A tabela 2.1 é um exemplo de tabela flutuante, cuja posição no texto pode variar em função das quebras de página.

Tabela 2.1: Os 16 modelos centrais do  $\text{UCON}_{\text{ABC}}$

	0 (imutável)	1 ( <i>pre-update</i> )	2 ( <i>on-update</i> )	3 ( <i>pos-update</i> )
preA	•	•	—	•
onA	•	•	•	•
preB	•	•	—	•
onB	•	•	•	•
preC	•	—	—	—
onC	•	—	—	—

<sup>1</sup>NUNCA use JPG ou GIF para desenhos vetoriais, pois o resultado final geralmente fica borrado.

### 2.4.3 Exemplo de fórmula

Equações destacadas devem ser numeradas como mostra a equação 2.1:

$$E = m \times c^2 \quad (2.1)$$

### 2.4.4 Exemplos de código-fonte

Códigos-fonte podem ser produzidos de forma simples através do ambiente `verbatim`, como mostra este exemplo:

```
#include <stdio.h>

int main (int argc, char *argv[])
{
    int i ;                      // uma variavel local

    for (i=0; i< 100; i++)        // um laço for
        printf ("i vale %d\n", i) ;    // uma saída na tela
}
```

No entanto, é preferível usar pacotes especializados para a edição ou inclusão de códigos-fonte, como o pacote `listings`. Eis um exemplo de código-fonte escrito com esse pacote:

```
1 #include <stdio.h>
2
3 int main (int argc, char *argv[])
4 {
5     int i ;                      // uma variável local
6
7     for (i=0; i< 100; i++)        // um laço for
8         printf ("i vale %d\n", i) ; // uma saída na tela
9 }
```

Esse pacote também permite incluir códigos-fonte de arquivos externos. Eis um exemplo:

```
1 #include <stdio.h>
2
3 int main (int argc, char *argv[])
4 {
5     int i ;                      // uma variável local
6
7     for (i=0; i< 100; i++)        // um laço for
8         printf ("i vale %d\n", i) ; // uma saída na tela
9 }
```

### 2.4.5 Exemplo de algoritmo

Os pacotes `algorithm` e `algorithmic` permitem formatar algoritmos facilmente. Eis um exemplo:



---

**Algoritmo 1** Ações de  $s_i$  ao encerrar um ciclo:

---

```

1: for all  $x \in \mathcal{K}_i$  do
2:    $banned_i(x) \leftarrow \text{FALSE}$ 
3:    $mi_i(x) \leftarrow 0$ 
4:    $mm_i(x) \leftarrow 0$ 
5:    $age_i(x) \leftarrow age_i(x) + 1$ 
6:   if  $age_i(x) = age_{max}$  then
7:      $\mathcal{K}_i \leftarrow \mathcal{K}_i - \{x\}$  // “esquece” do servidor  $x$ 
8:     remove as informações locais sobre  $x$ 
9:     envia  $notify(x, undef)$  ao grupo de confiança  $\mathcal{T}_i$ 
10:  end if
11: end for

```

---

#### 2.4.6 Exemplo de citação

Como afirmou Maquiavel em seu livro *O Príncipe*:

“Nada é mais difícil de instituir, mais perigoso de conduzir, mais incerto no seu sucesso, do que liderar a introdução de uma nova ordem de coisas... O inovador faz inimigos em todos aqueles que prosperavam sobre as antigas regras, e somente tíbio suporte é esperado daqueles que prosperariam na novidade, porque os homens são geralmente incrédulos, nunca realmente confiam nas coisas novas, a menos que as tenham testado em experiência”.

### 2.5 UMA SEÇÃO

#### 2.5.1 Uma subseção

##### 2.5.1.1 Uma subsubseção

### 2.6 CONCLUSÃO

Todo capítulo (com exceção da introdução e da conclusão) deve encerrar com uma pequena conclusão local, resumindo os tópicos apresentados no capítulo e preparando o leitor para o próximo capítulo (exceto se esse for a conclusão geral). Caso o capítulo tenha apresentado resultados obtidos pelo próprio autor, estes devem ser sucintamente lembrados aqui.

## REFERÊNCIAS

Goossens, M., Mittelbach, F. e Samarin, A. (1993). *The  $\text{\LaTeX}$  Companion*. Addison-Wesley.

Oetiker, T., Partl, H., Hyna, I. e Schlegl, E. (2007). *The Not So Short Introduction to  $\text{\LaTeX}$  2 $\epsilon$* .  
<http://www.ctan.org/tex-archive/info/lshort>.