UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
PESQUISA OPERACIONAL

IMPLEMENTAÇÃO DO MÉTODO SIMPLEX EM PYTHON

Diogo Augusto Martins Honorato

Gustavo Euller Honório Pedrosa

São João del-Rei
2025

Sumário

1 Introdução

Este trabalho tem como objetivo implementar o Método Simplex para resolução de Problemas de Programação Linear (PPL). O algoritmo permite resolver problemas de maximização sujeitos a restrições lineares com variáveis não-negativas. A implementação foi feita em Python.

2 Execução do Programa

2.1 Requisitos

- Python 3
- Biblioteca NumPy

2.2 Como Executar

Salve um arquivo de entrada no formato descrito na Seção 3. Em seguida, execute o programa com:

```
python3 main.py
```

3 Formato do Arquivo de Entrada

Primeira linha o número de variáveis e número de restrições, segunda linha os coeficientes da função objetivo e linhas seguintes os coeficientes das restrições seguidos do tipo da restrição ($\leq$ ou $\geq$) e do lado direito

Exemplo:

```
2 2
3 2
2 1 <= 10
1 3 <= 15
```

4 Resultados

Input:

```
2 2
3 2
2 1 <= 10
1 3 <= 15
```

Tableau Inicial:

Variáveis básicas iniciais: s_1 (índice 2), s_2 (índice 3) e FO: $-3x_1 - 2x_2$ (coeficientes negados)

Processo de Solução:

Entra na base: x_1 (coeficiente mais negativo na FO) logo em seguida sai da base: s_1 (razão mínima $10/2 = 5$) o pivoteamento e realizado e então uma nova iteração: entra x_2, sai s_2

Saída do Programa:

```
Tableau Inicial:
[[-3. -2.  0.  0.  0.]
 [ 2.  1.  1.  0. 10.]
 [ 1.  3.  0.  1. 15.]]
Variáveis básicas: [2, 3]
Variáveis não básicas: [0, 1]
------------------------------
Solução encontrada:
Valor ótimo: 17.0000
Variáveis (x1, x2): [3. 4.]
Variáveis básicas finais: [0, 1]
Tableau Final:
[[ 0.   0.   1.4  0.2 17. ]
 [ 1.   0.   0.6 -0.2  3. ]
 [ 0.   1.  -0.2  0.4  4. ]]
------------------------------
```

Solução ótima: $x_1 = 3$, $x_2 = 4$ com o valor ótimo da FO: 17 e variáveis básicas finais: x_1 e x_2 e com coeficientes zero na FO para s_1 e s_2 indicam solução única

5 Descrição dos Módulos

5.1 `_create_initial_tableau(A, b, c, types)`

Propósito: Monta o tableau inicial do problema.

Processo:

1. Calcula número de variáveis auxiliares (folga/excesso)
2. Cria matriz tableau com dimensões adequadas
3. Preenche a linha da função objetivo (coeficientes negados)
4. Para cada restrição:
   - Adiciona coeficientes das variáveis de decisão
   - Adiciona variável de folga (1) ou excesso (-1)
   - Adiciona termo independente
5. Define variáveis básicas e não básicas iniciais

5.2 `solve()`

Propósito: Executa o algoritmo Simplex até encontrar solução ótima ou detectar problema ilimitado.

Passos:

1. Verifica otimalidade (todos coeficientes na FO ≥ 0)
2. Seleciona variável para entrar na base (mais negativo na FO)
3. Verifica ilimitabilidade
4. Seleciona variável para sair da base (teste da razão mínima)
5. Atualiza base e não base

6. Realiza operação de pivoteamento
7. Repete até condição de parada

5.3 `_check_multiple_solutions()`

Propósito: Verifica se existem múltiplas soluções ótimas.

Lógica:

- Verifica se alguma variável não básica tem coeficiente zero na FO
- Isso indica que poderíamos trazê-la para a base sem alterar o valor da FO

5.4 `get_solution()`

Propósito: Extrai e retorna a solução encontrada.

Retorno:

- Dicionário com:
  - `solution`: Valores das variáveis de decisão
  - `optimal_value`: Valor ótimo da FO
  - `multiple_solutions`: Flag para múltiplas soluções
  - `base`: Variáveis básicas finais
- Ou mensagem de erro se problema ilimitado ou não resolvido

6 Capacidades e Limitações da Implementação

6.1 Problemas Tratados com Sucesso

O programa implementado é capaz de resolver eficientemente:

- **Problemas padrão de maximização** com:
  - Restrições do tipo $\leq$ com lados direitos positivos
  - Qualquer número de variáveis e restrições (testado até 20x20)
  - Solução ótima única finita

- **Casos especiais**:
  - Problemas ilimitados (identifica corretamente quando a FO pode crescer infinitamente)
  - Múltiplas soluções ótimas (detecta quando existem pontos ótimos alternativos)
  - Degeneração (lida corretamente com variáveis básicas de valor zero)

6.2 Limitações Conhecidas

A implementação atual **não trata** os seguintes casos:

- **Restrições de igualdade** (=):
  - Não implementa variáveis artificiais
  - Exemplo: $x_1 + x_2 = 5$ causa erro no tableau inicial
- **Restrições $\geq$ com base infactível**:
  - Adiciona variáveis de excesso mas não resolve infactibilidade inicial
  - Exemplo: $2x_1 - x_2 \geq 5$ pode não convergir
- **Problemas infactíveis**:
  - Não implementa Fase I do Simplex ou Método Big M
  - Não detecta explicitamente contradições como $x_1 \leq 3$ e $x_1 \geq 5$
- **Minimização**:
  - Converte apenas problemas de maximização
  - Minimização requer transformação manual para $-$FO

7 Conclusão

A implementação do Método Simplex foi realizada com sucesso e validada por diversos casos de teste. A estrutura modular facilita a compreensão e ampliação futura, como a implementação de fases auxiliares para tratamento de restrições do tipo = e $\geq$. O código atual atende corretamente problemas padrão de maximização com variáveis não-negativas.