

Trabalho Prático - Simplex

Professor: *Felipe Reis*

Disciplina: *Pesquisa Operacional*
Data de entrega: 23/12/2021

Objetivos

- Implementar os métodos Simplex e Simplex Duas Fases, ensinados em sala de aula, para solução de problemas de programação linear;
- Aprofundar o conhecimento nos métodos Simplex e Simplex Duas Fases.

Requisitos

Implementar a função “Solver”, seguindo a estrutura indicada no arquivo “solver.py”, disponível junto aos documentos do trabalho.

A função “Solver” possui a estrutura abaixo:

```
def solver(objet, f_obj, restr_A, restr_op, restr_b, verbose=False):  
    """  
    Função para solução de problemas de programação linear  
    Deve ser implementada para solução de PPL usando métodos Simplex e Simplex Duas Fases.  
    Não devem ser adicionados novos parâmetros à função.  
  
    Parâmetros:  
    : param objet: string, contendo indicação de 'MA' (para maximização) ou 'MI' (para minimização)  
    : param f_obj: np.array float, contendo a função objetivo  
    : param restr_A: np.array float, contendo a matriz A das restrições  
    : param restr_op: np.array string, contendo o vetor de operadores das restrições  
    : param restr_b: np.array float, contendo o vetor b das restrições  
    : param verbose: booleano opcional para impressão de valores intermediários (não obrigatório implementar)  
    : return: np.array contendo os valores das variáveis (não retornar valor da função objetivo nem variáveis de folga)  
    """
```

Fonte: Próprio autor

Nesta função, todos os parâmetros recebidos deverão ser arrays/matrizes Numpy (“*np.array*”). O retorno também deve ser um array Numpy contendo os valores das variáveis.

Para implementação podem ser criados funções auxiliares, de modo a organizar o código fonte.

Arquivo de Entrada. O arquivo de entrada contém os problemas de programação linear. Esse arquivo contém exemplos, que deverão ser utilizados como referência para implementação do método “Solver”.

A função “Solver” não deve ler os arquivos, apenas receber os problemas de programação linear no formato indicado. Caso o aluno deseje automatizar o processo, deverá criar uma função extra para leitura do arquivo e transformação dos arquivos de entrada (*string*) em arrays Numpy.

A estrutura do arquivo está exemplificada abaixo. Cada linha do arquivo contém um problema. As linhas iniciadas com o caractere “#” correspondem a comentários e devem ser ignoradas.

```

1 #simplex
2 #
3 MA[3 2] RE[1 1 <= 6], [5 2 <= 20] SF[14.67] SV[2.67 3.33]
4 MI[4 -2] RE[2 1 <= 10], [1 -1 <= 8] SF[-20] SV[10 18]
5 MA[5 4] RE[6 4 <= 24], [1 2 <= 6], [-1 1 <= 1], [0 1 <= 2] SF[21] SV[3 1.5]
6 MA[1000 1800] RE[20 30 <= 1200], [1 0 <= 40], [0 1 <= 30] SF[69000] SV[15 30]
7 MA[80 60] RE[4 6 <= 24], [4 2 <= 16], [0 1 <= 3] SF[360] SV[3 2]
8 MA[600 800 900 500] RE[1 1 1 1 <= 400], [1000 1200 1200 1500 <= 500000], [20 30 25 28 <= 10000] SF[360000] SV[0 0 400 0]
9 #
10 #simplex duas fases
11 #
12 MI[3 2.5] RE[4 8 >= 32], [6 6 >= 36] SF[15] SV[0 6]
13 MA[9 8] RE[4 2 <= 16], [1 1 <= 5], [1 0 >= 1], [0 1 <= 3] SF[43] SV[3 2]
14 MA[5 8] RE[2 1 <= 5], [-1 2 >= 3], [1 0 <= 1], [0 1 >= 1] SF[40] SV[0 5]
15 MA[12 20] RE[0 1 >= 1], [3 2 >= 6], [1 1 <= 5], [0 1 <= 3] SF[84] SV[2 3]
16 MA[8 10] RE[2 2 <= 6], [4 1 >= 4], [3 1 <= 5], [0 1 >= 1] SF[16] SV[0.75 1]
17 MI[20 25 25 25] RE[1 0 1 == 500], [0 1 0 1 == 1000], [1 0 0 0 == 400], [0 1 0 0 == 800], [0 0 1 1 == 600] SF[35500] SV[400 800 100 200]

```

Fonte: Próprio autor

As linhas possuem a seguinte estrutura:

- Função Objetivo
 - Indicador “MA” ou “MI”, para maximização e minimização, respectivamente, seguido de um array;
 - Array contém o coeficientes das variáveis da função objetivo, separados por espaço simples;
- Restrições (RE)
 - Indicador de restrições (RE) seguido de arrays;
 - Arrays (1 ou múltiplos, separados por ‘,’) contendo coeficientes das restrições e operadores;
 - Dentro dos arrays, os valores estão separados por espaço simples;
 - Operadores utilizam sempre dois caracteres: ‘==’, ‘<=’ ou ‘>=’;
 - Não há espaçamento entre a sigla ‘RE’ e o primeiro array;
- Solução Função Objetivo (SF)
 - Indicador de solução da função objetivo (SF) seguido de array;
 - Array contém o resultado da função objetivo;
 - *Deve ser utilizado somente para simples conferência. Não deve ser passado como parâmetro para a função Solver.*
- Solução das Variáveis (SV)
 - Indicador de solução das restrições (SV) seguido de array
 - Array contém os valores das variáveis
 - *Deve ser utilizado somente para simples conferência. Não deve ser passado como parâmetro para a função Solver.*

Simplex Tableau. Para solução dos problemas, devem ser criadas matrizes correspondentes ao Simplex Tableau.

O Tableau deve, **obrigatoriamente**, seguir o modelo adotado na disciplina e disponível nos slides de aula.

Exemplos. Um arquivo Jupyter / Colab contendo exemplos de parâmetros da função Solver e da saída esperada serão fornecidos aos alunos, para auxiliar na padronização do código.

Requisitos Adicionais. As seguintes informações adicionais de

1. O código deve ser genérico, não devendo funcionar apenas para os problemas existentes no arquivo exemplo;
2. Na correção, será utilizado um arquivo extra, com mais problemas, que devem ser solucionados pelo algoritmo;
3. Todos os problemas serão factíveis. O aluno não deve se preocupar com problemas não factíveis;
4. Informações de solução da Função Objetivo e Variáveis não devem ser passadas como parâmetros, sob nenhuma hipótese, para a função *solver*.

Requisitos da Entrega

A implementação dos métodos deve ser feita apenas no arquivo “*solver.py*”. Esse arquivo deve conter todo o código necessário à implementação dos algoritmos Simplex e Simplex Duas Fases.

Não serão aceitos trabalhos em formatos diferentes dos especificado (ex.: arquivos *.ipynb*).

Protocolo de Correção. Para correção do trabalho será criado um pequeno algoritmo que fará a leitura do arquivo de entrada, transformará as linhas em arrays Numpy e invocará o método “*Solver*”.

O retorno do método “*Solver*” será validada automaticamente pelo algoritmo de correção do professor.

O professor poderá avaliar a qualidade de código, caso perceba que o algoritmo foi escrito sem adoção de boas práticas de programação.

Análise de Plágio. Os códigos fontes serão analisados por ferramentas de identificação de plágio. Não forneça seu código a outros alunos.

Data de Entrega e Pontuação

O trabalho deve ser entregue no dia **23/12/2021**.

O trabalho deve ser feito **individualmente** e terá valor de **20 pontos**.

Critérios de avaliação

Abaixo estão descritos os critérios que serão adotados na correção do trabalho, bem como seu valor.

Etapa	Descrição	Pontuação
1	Método Simplex	
	- Implementação do método e corretude.	10 pontos
	- Qualidade de escrita de código e documentação.	
2	Método Simplex Duas Fases	
	- Implementação do método e corretude.	10 pontos
	- Qualidade de escrita de código e documentação.	