

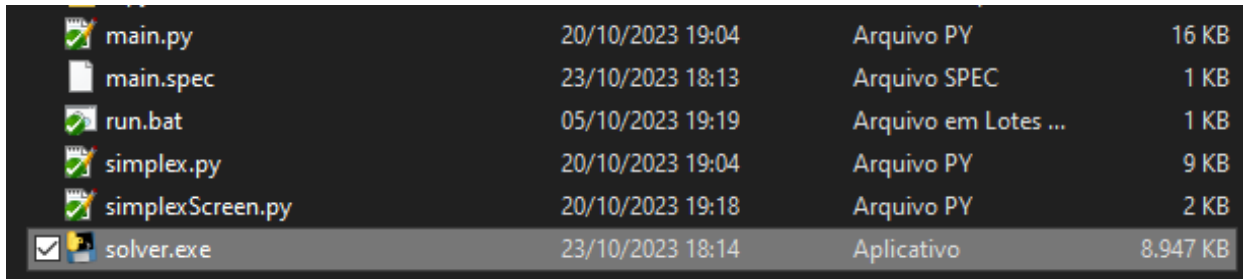
Manual Solver







Estudantes: Mateus Ferro Antunes de Oliveira e Tasi Guilhen Pasin

Inicialização:

Para iniciar a aplicação, abrir o arquivo **solver.exe**, conforme é apresentado na Figura 1. Os demais arquivos são referentes à implementação do código.

Figura 1. Seleção do aplicativo.



	main.py	20/10/2023 19:04	Arquivo PY	16 KB
	main.spec	23/10/2023 18:13	Arquivo SPEC	1 KB
	run.bat	05/10/2023 19:19	Arquivo em Lotes ...	1 KB
	simplex.py	20/10/2023 19:04	Arquivo PY	9 KB
	simplexScreen.py	20/10/2023 19:18	Arquivo PY	2 KB
<input checked="" type="checkbox"/> 	solver.exe	23/10/2023 18:14	Aplicativo	8.947 KB

Fonte: Os Autores (2023).

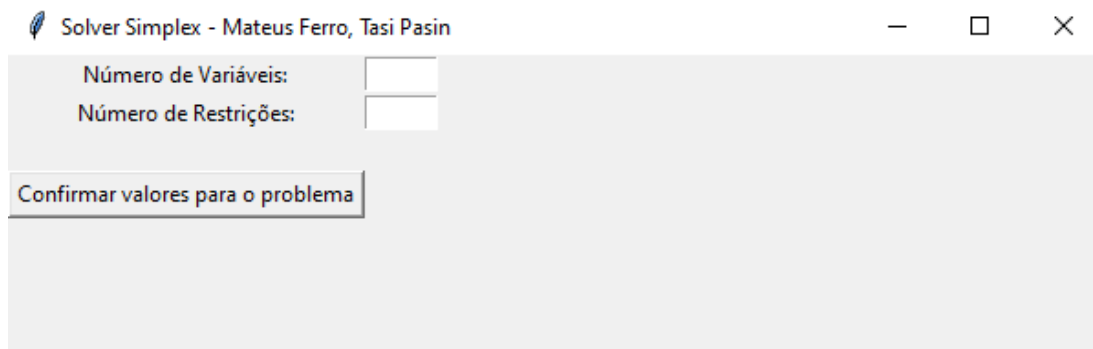
Entrada de Dados:

A primeira tela que aparece quando a aplicação é executada é a definição do tamanho do problema, conforme mostrado na Figura 2.

Os campos de entrada são: Número de Variáveis e Número de Restrições, ambos referenciando a quantidade de cada item. E como o nome diz, o campo “Número de Variáveis”, é a entrada para a quantidade de variáveis que o problema utiliza, e o campo “Número de Restrições”, é a quantidade de restrições que o problema utiliza. Nessa conta, não entram as condições de não negatividade.

Após preenchidos os campos de entrada, clicar no botão **“Confirmar valores para o problema”** para prosseguir com a determinação do problema.

Figura 2. Entrada para variável e restrição.



Solver Simplex - Mateus Ferro, Tasi Pasin

Número de Variáveis:

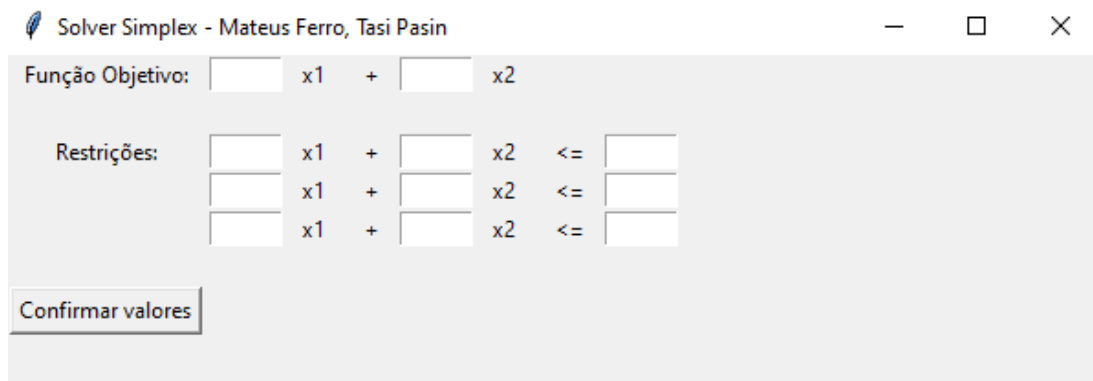
Número de Restrições:

Confirmar valores para o problema

Fonte: Os Autores (2023).

Na próxima tela (Figura 3), inserir os valores das variáveis e equações do problema. A aplicação aceita somente problemas do tipo de Maximização na forma canônica. Ainda que não tenha os campos de não negatividade, a aplicação considera que todas as variáveis da Função Objetivo atendem à essa condição.

Figura 3. Dados para o problema - Função objetivo e restrições.



Solver Simplex - Mateus Ferro, Tasi Pasin

Função Objetivo: x1 + x2

Restrições:

<input type="text"/>	x1	+	<input type="text"/>	x2	<=	<input type="text"/>
<input type="text"/>	x1	+	<input type="text"/>	x2	<=	<input type="text"/>
<input type="text"/>	x1	+	<input type="text"/>	x2	<=	<input type="text"/>

Confirmar valores

Fonte: Os Autores (2023).

Então, para as entradas, tem-se a Função Objetivo e as Restrições, onde deve inserir os índices das variáveis (valores numéricos apenas), podendo ser valores inteiros ou fracionados nas formas 1/2 ou 0.5, por exemplo. Se alguma variável não aparece nas equações, preencher seu respectivo campo com o valor 0. Exemplo de preenchimento na Figura 4.

Figura 4. Preenchimento de valores para um problema qualquer.

Solver Simplex - Mateus Ferro, Tasi Pasin

Função Objetivo: 3 x1 + 2 x2

Restrições:

5	x1	+	1	x2	<=	10
2	x1	+	2	x2	<=	12
1	x1	+	4	x2	<=	12

Confirmar valores

Fonte: Os Autores (2023).

Ao preencher todos os campos, clicar no botão “**Confirmar Valores**” para iniciar a resolução. Além disso, a resolução é feita em etapas para que seja possível acompanhar os resultados. O primeiro passo após clicar no botão “**Confirmar Valores**”, é observar a tabela do Simplex na Figura 5. Nessa tabela, são apresentados os valores inseridos, as variáveis base, as variáveis de excesso e o beta já preenchidos. Os demais campos, como Z_j , $C_j - Z_j$ e θ não estão preenchidos neste momento.

Figura 5. Tabela inicial para o Simplex.

Solver Simplex - Mateus Ferro, Tasi Pasin

Iteração 0	x1	x2	x3	x4	x5	beta	theta
x3	5.0	1.0	1	0	0	10.0	
x4	2.0	2.0	0	1	0	12.0	
x5	1.0	4.0	0	0	1	12.0	


Z_j							
$C_j - Z_j$							

Calcular Pivô

Fonte: Os Autores (2023).

Para executar a iteração atual, clicar no botão “**Calcular Pivô**” para preencher a tabela com os cálculos de Z_j , $C_j - Z_j$ e θ . Nesse passo é verificado se existe elemento pivô e é destacado, em azul, a variável que sai (linha), e a variável que entra (coluna).

Figura 6. Demonstração do Elemento Pivô a partir dos dados.



Solver Simplex - Mateus Ferro, Tasi Pasin

Iteração 0		x1	x2	x3	x4	x5	beta	theta
	x3	5.0	1.0	1	0	0	10.0	2.0
	x4	2.0	2.0	0	1	0	12.0	6.0
	x5	1.0	4.0	0	0	1	12.0	12.0

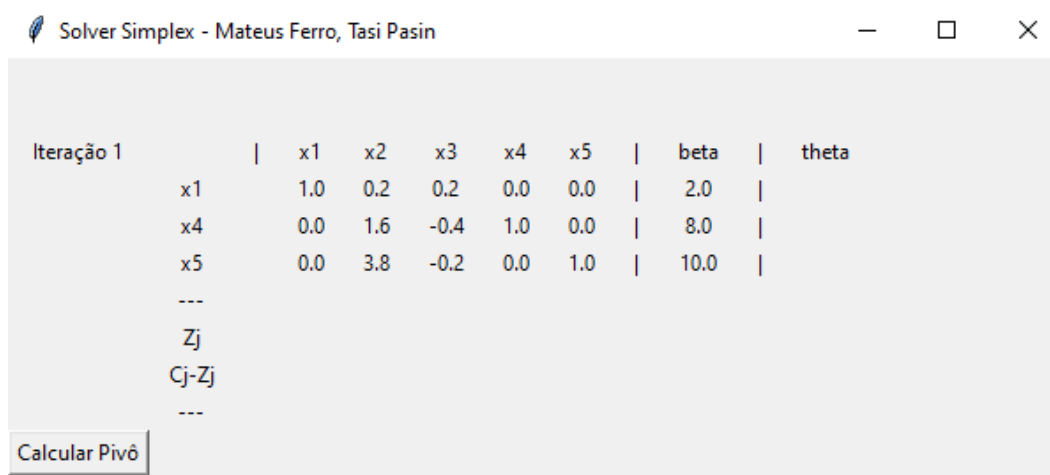
	Z_j	0.0	0.0	0	0	0		
	$C_j - Z_j$	3.0	2.0	0	0	0		

Próxima Iteração

Fonte: Os Autores (2023).

Após clicar no botão “**Próxima Iteração**”, a tabela é atualizada com os valores da nova iteração e a indicação de qual iteração o programa está, vide Figura 7.

Figura 7. Próxima Iteração.



Solver Simplex - Mateus Ferro, Tasi Pasin

Iteração 1		x1	x2	x3	x4	x5	beta	theta
	x1	1.0	0.2	0.2	0.0	0.0	2.0	
	x4	0.0	1.6	-0.4	1.0	0.0	8.0	
	x5	0.0	3.8	-0.2	0.0	1.0	10.0	

	Z_j							
	$C_j - Z_j$							

Calcular Pivô

Fonte: Os Autores (2023).

Quando chegar no fim do problema, o resultado é apresentado em valores para um Z final (“maior lucro”), e valores para as variáveis problema, como na Figura 8.

Figura 8. Exemplo de exibição do resultado.

Solver Simplex - Mateus Ferro, Tasi Pasin

Iteração 2		x1	x2	x3	x4	x5	beta	theta
	x1	1.0	0.0	0.21	0.0	-0.05	1.47	
	x4	0.0	0.0	-0.32	1.0	-0.42	3.79	
	x2	0.0	1.0	-0.05	0.0	0.26	2.63	

	Zj	3.0	2.0	0.53	0.0	0.37	9.68	
	Cj-Zj	0.0	0.0	-0.53	0.0	-0.37		

	Z final:	9.68						
	x1 =	1.47						
	x2 =	2.63						

Fonte: Os Autores (2023).

O Solver Simplex está preparado para identificar Problemas Sem Fronteira, Sistema Degenerado e Solução Ótimos Alternados. Para o caso de Problema Sem Fronteira, não é possível determinar uma linha de *Theta* que possa ser substituída, por isso a resolução é encerrada e mostra a mensagem em vermelho - Figura 9.

Figura 9. Exemplo de final de execução para Problema Sem Fronteira.

Solver Simplex - Mateus Ferro, Tasi Pasin

—□×

Iteração 1

x3	0.0	-6.0	1.0	-0.33	1.33	-0.22
x1	1.0	0.0	0.0	0.33	3.67	inf

Zj	4.0	0.0	0.0	1.33	14.67	
Cj-Zj	0.0	3.0	0.0	-1.33		

Problema Sem Fronteira

Z final: 14.67

x1 = 3.67

Fonte: Os Autores (2023).

Para o caso de sistema degenerado, é possível escolher um valor de *Theta* para substituir uma variável na solução, porém mais de uma linha pode ser escolhida, já que essa condição se dá quando existem dois valores que são os menores da iteração e maiores do que zero. Como esse caso não impede a continuidade da resolução do problema, a mensagem é apresentada em amarelo - Figura 10.

Figura 10. Exemplo de final de execução para Sistema Degenerado.

Solver Simplex - Mateus Ferro, Tasi Pasin

Iteração 0		x1	x2	x3	x4	beta	theta
	x3	2.0	3.0	1	0	8.0	4.0
	x4	3.0	2.0	0	1	12.0	4.0

	Zj	0.0	0.0	0	0		
	Cj-Zj	4.0	3.0	0	0		

Próxima Iteração

Sistema Degenerado

Fonte: Os Autores (2023).

Assim como o Sistema Degenerado, quando o sistema detecta uma Solução com Ótimos alternados a mensagem é apresentada em amarelo, indicando que o problema consegue continuar a resolução. Nesse caso, porém, a resolução nunca será interrompida e cabe ao usuário determinar a solução ideal para si - Figura 11.

Figura 11. Exemplo de final de execução para Solução Ótimos Alternados.

Solver Simplex - Mateus Ferro, Tasi Pasin

Iteração 1		x1	x2	x3	x4	beta	theta
	x1	1.0	0.75	0.12	0.0	3.12	4.17
	x4	0.0	1.75	-0.38	1.0	5.62	3.21

	Zj	4.0	3.0	0.5	0.0		
	Cj-Zj	0.0	0.0	-0.5	0.0		

Próxima Iteração

Solução
Ótimos Alternados

Fonte: Os Autores (2023).

As mensagens de aviso aparecem somente após resolver a iteração atual e permanecem apenas naquela iteração. Uma vez que passa para a próxima iteração, a mensagem é removida e, caso aconteça novamente, será apresentada novamente a informação.