Plint: Applet de Programação Linear

Fernando Filipe da Cruz Vidigal
Licenciado Informática, Univ. Aberta,
fernandovidigal@gmail.com

Resumo

Problemas de programação linear são problemas de otimização no qual a função objetivo e as restrições são lineares. São problemas em que se procura minimizar ou maximizar o valor da função objetivo, ou seja, procura-se a melhor solução possível de entre todas as soluções viáveis. Temos como exemplos de problemas de otimização: a maximização da receita, a minimização do custo, a maximização de recursos, entre outros. Neste projeto foi construída uma ferramenta que permite a obtenção da solução para este tipo de problemas de uma forma automática, e que apresenta também todos os passos da aplicação dos métodos ao problema. Os métodos implementados são: O método Simplex, o método das Duas Fases e o método Simplex Dual.

palavras-chave: programação linear, otimização, minimização, maximização, simplex, duas fases, simplex dual.

Title: Plint: linear programming applet

Abstract

Linear programming problems are optimization problems where the objective function and constraints are linear. These are problems that seek to minimize or maximize the value of the objective function, i.e., looking for the best possible solution among all feasible solutions. We have as examples of optimization problems: revenue maximization, minimization of cost and maximizing resources. In this project a tool was built that allows us to obtain the solution to such problems in an automated way, and also features all the steps of applying the methods to the problem. The implemented methods are: the Simplex method, the Two-Phase method and the Dual Simplex method.

keywords: linear programming, optimization, minimizing, maximizing, simplex, two-phases, dual simplex.

1. Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular de Projeto Final da licenciatura em Informática, tendo sido sugerido e orientado pelo Professor Mário Edmundo. Teve como principal objetivo a construção de uma Applet para implementação de alguns algoritmos de Programação Linear, nomeadamente o método do Simplex, o método das

Duas Fases e o método Simplex Dual.

Uma Applet é uma aplicação que executa uma tarefa específica e que corre dentro de um determinado contexto, que normalmente é o browser. É um termo frequentemente utilizado para se referir a um Java Applet, um programa desenvolvido em linguagem Java com o objetivo de ser colocado e executado numa página web.

2. Objetivos

Além do objetivo principal do projeto, pretende-se que a Applet tivesse uma interface gráfica dinâmica e que fosse intuitiva e de fácil utilização. Que fornecesse não só os resultados finais do cálculo mas também todos os quadros do simplex intermédios resultantes de cada iteração dos algoritmos. Pretendo assim que a Applet seja uma ajuda na compreensão e aplicação dos métodos implementados, além da obtenção dos resultados dos problemas. Foi também construída uma página web onde a Applet será executada.

3. A Applet

A linguagem de programação utilizada para a construção da Applet foi o Java. Como o java bytecode é independente da plataforma onde é executado, a Applet pode ser utilizada/executada em diferentes plataformas (Windows, Linux, Mac OS).
Ao contrário do que acontece com outros programas construídos em Java, onde existe um método principal *main()*, numa Applet não existe nenhum método *main()*, existe sim um conjunto de métodos que representam as fases mais importantes na vida de uma Applet [Coelho 2014]:

- *init()* – **Inicialização**, é a fase em que a Applet é, pela primeira vez, carregada. [Coelho 2014]
- *start()* – **Começar**, é a fase que ocorre a seguir à inicialização ou, depois de uma paragem. [Coelho 2014]
- *stop()* – **Parar**, é a fase que implica uma paragem na execução da Applet. [Coelho 2014]
- *destroy()* – **Destruir**, é a fase terminal de uma Applet. [Coelho 2014]

A Applet contem duas áreas. Uma área é constituída pela interface gráfica, para introdução dos dados do problema e outra área onde é mostrada a forma *standard*, os resultados intermédios e os resultados finais da aplicação dos métodos ao problema.

Pretendia que a interface gráfica fosse simples e adaptável ao problema, ou seja, dinâmica, que permitisse fazer a seleção do método de cálculo desejado, alternar entre problemas de minimização e maximização, adicionar e remover variáveis ou restrições de forma dinâmica e ter a possibilidade de definir as restrições das variáveis, ou seja, quais variáveis são livres ou não negativas. Os únicos dados que o utilizador tem de inserir na interface são os coeficientes das variáveis, fazer a seleção do sinal das desigualdades das restrições e o valor de cada restrição. Mesmo sendo o utilizador a selecionar o método de cálculo, a Applet, antes de iniciar o cálculo, verifica se o método selecionado é o adequado para o problema introduzido. Caso não seja, o método adequado para o cálculo é utilizado e é

exibida ao utilizador uma mensagem com essa indicação.

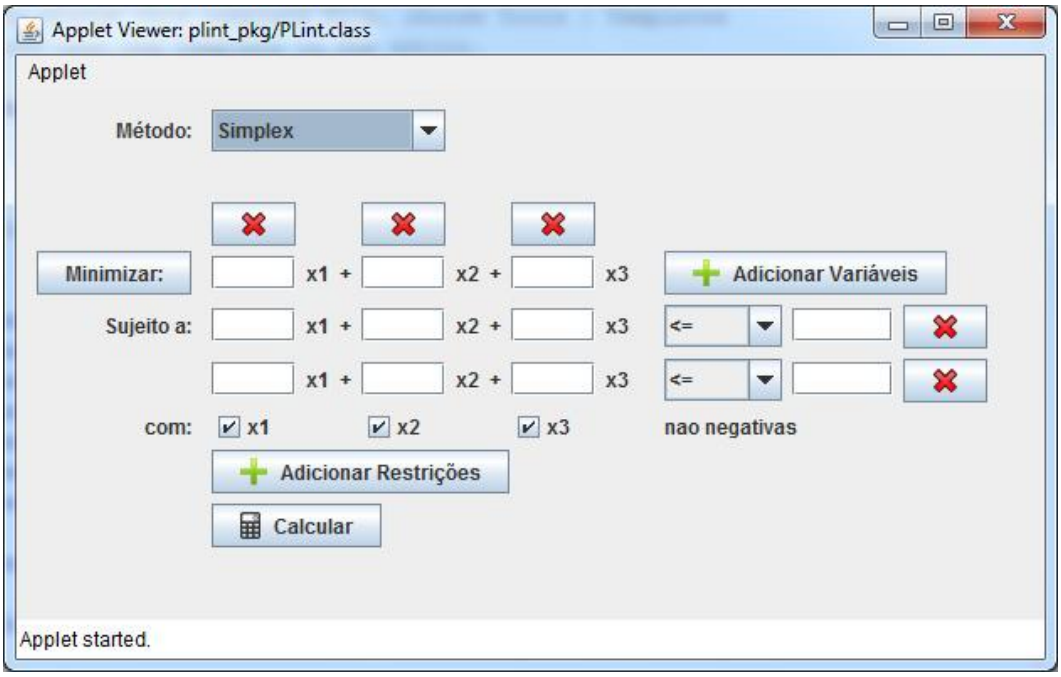

Figura 1 - Interface da Applet

A outra área é onde são mostrados os resultados e a forma standard do problema. Esta área mostra a indicação do método que foi utilizado no cálculo, o problema na forma standard, os quadros do simplex intermédios, resultantes das iterações dos algoritmos e os resultados finais. Os resultados intermédios são acompanhados pela indicação do elemento pivô, coluna (linha no caso do método Simplex Dual) de trabalho e as operações aritméticas realizadas. O quadro do simplex final é acompanhado com o valor final de cada variável e o valor final da função objetivo.

4. Os Métodos

Os métodos implementados são procedimentos matriciais utilizados para resolver problemas de otimização expressos na forma standard [Bronson 2007]. A forma standard é do tipo:

$$\textit{Otimizar: } z = C^T X$$
$$\textit{Sujeito a: } AX = B$$
$$\textit{com: } X \geq 0$$

4.1 - Forma Standard

A forma standard é conseguida através da introdução de variáveis de folga, de excesso ou artificiais.

Às restrições com sinal $\leq$ é adicionada à função objetivo e às restrições uma variável de

folga.

Às restrições com sinal $\geq$ é adicionada à função objetivo uma variável de excesso e uma variável artificial. A variável de excesso é adicionada às restrições com coeficiente negativo.

Às restrições com sinal $=$ é adicionada à função objetivo e às restrições uma variável artificial.

Se no problema existirem variáveis livres, isto é, não restritas, serão substituídas pela diferença entre duas novas variáveis não negativas [Bronson 2007] (exemplo: $x_5 = x_6 - x_7$, com x_5 livre e x_6 e x_7 não negativas). Assim todas as variáveis do problema são não negativas.

No método Simplex Dual a forma standard é conseguida de forma diferente. A forma standard para aplicação do método Simplex Dual, consiste em colocar todas as desigualdades das restrições na forma $\leq$ e depois adicionar variáveis de folga.

4.2 Método Simplex

Processo iterativo do método Simplex:

1. Localizar o número mais negativo na última linha do quadro simplex, excluindo a última coluna, designando a coluna em que este número aparece como coluna de trabalho. Se existir mais que um candidato a número mais negativo, escolhe-se um. [Bronson 2007]
2. Para cada linha do quadro (excetuando a última) com coeficiente positivo na coluna de trabalho, dividir cada elemento da última coluna pelo correspondente coeficiente na coluna de trabalho. Designa-se o coeficiente correspondente ao menor quociente por elemento pivô. Se mais de um coeficiente produzir o mesmo quociente mínimo, escolhe-se um. Se nenhum coeficiente da coluna de trabalho for positivo, o problema não terá solução. [Bronson 2007]
3. Usar operações elementares sobre linhas para converter o elemento pivô em 1 e, em seguida, reduzir a zero todos os outros elementos da coluna de trabalho. [Bronson 2007]
4. Na primeira coluna, substitua a variável existente na linha pivô, pela variável correspondente à coluna de trabalho (indicada na primeira linha). A nova primeira coluna indica o novo conjunto de variáveis básicas. [Bronson 2007]
5. Repetir os passos 1 a 4, até que não existam números negativos na última linha. Excluindo desta apreciação a última coluna. [Bronson 2007]
6. A solução ótima é obtida atribuindo-se a cada variável da primeira coluna o valor da linha correspondente, na última coluna. Todas as outras variáveis são nulas. O valor ótimo da função objetivo é o valor indicado na última linha, última coluna, para problemas de maximização. Em problemas de minimização será o seu simétrico. [Bronson 2007]

		x1	x2	x3	x4	x5	
		1,00	9,00	1,00	0,00	0,00	
x4	0,00	1,00	2,00	3,00	1,00	0,00	9,00
x5	0,00	3,00	2,00	2,00	0,00	1,00	15,00
		-1,00	-9,00	-1,00	0,00	0,00	0,00

Figura 2 - Quadro do Simplex

4.3 - Método Duas Fases

Sempre que da solução inicial fizerem parte variáveis artificiais (custos de penalização *M*), incorporam-se no método Simplex modificações que originam o chamado método Duas Fases [Bronson 2007].

Modificações ao método Simplex:

1. A última linha do quadro do simplex é decomposta em duas linhas, a primeira das quais envolve os termos que não contêm M, e a segunda, os coeficientes de M nos restantes termos. [Bronson 2007]
2. O passo 1 do método Simplex é aplicado à última linha criada na Modificação 1 (seguida pelos passos 2, 3 e 4), até que esta linha não contenha elementos negativos. Em seguida, o passo 1 é aplicado aos elementos da penúltima linha posicionados sobre os zeros da última linha. [Bronson 2007]
3. Sempre que uma variável deixa de ser básica, isto é, deixa de figurar na primeira coluna do quadro, em resultado do passo 4, a correspondente coluna (identificada na linha superior do quadro por essa variável) deve ser removida. [Bronson 2007]
4. A última linha do quadro pode ser eliminada quando for constituída unicamente por zeros. [Bronson 2007]
5. Se na solução básica final estiverem presentes variáveis artificiais não nulas, então o problema original não admitirá solução. (Pelo contrário, na solução básica final podem aparecer variáveis artificiais na base com o valor zero, indicando que uma ou mais das equações que representam as restrições são redundantes.) [Bronson 2007]

4.4 - Método Simplex Dual

Processo iterativo do método Simplex Dual:

1. Reescrever o problema, expressando todas as restrições na forma $\leq$ e transformá-las em equações, com a introdução de variáveis de folga. [Bronson 2007]
2. Representar o problema num quadro do Simplex. Se a condição de otimalidade for satisfeita e uma ou mais variáveis básicas tiverem valores negativos, o método simplex dual é aplicável. [Bronson 2007]
3. Condição de Admissibilidade: a variável básica com valor mais negativo deverá sair da base. Designe a linha em que aparece este valor (mais negativo) como linha de trabalho. Se existir mais do que uma candidata a variável que deve sair da base, escolhe-se uma. [Bronson 2007]

4. Condição de Otimalidade: para cada coluna do quadro correspondente a uma variável, com coeficiente negativo na linha de trabalho, divide-se cada elemento da última linha, pelo correspondente coeficiente da linha de trabalho. O menor quociente, em valor absoluto, correspondente a uma variável não básica, indica que essa variável deve entrar para a base. Designa-se o coeficiente da linha de trabalho correspondente à variável que deve entrar na base por elemento pivô. Se existir mais do que uma variável não básica candidata a entrar na base, escolhe-se uma. Se nenhum coeficiente da linha de trabalho for negativo, o problema não tem solução admissível. [Bronson 2007]
5. Usar operações elementares sobre linhas para converter o elemento pivô em 1 e, em seguida, reduzir a zero todos os outros elementos da coluna de trabalho.
6. Repetir os passos 3 a 5 até que não existam valores negativos atribuídos às variáveis básicas. [Bronson 2007]

5. Estrutura de Dados

A estrutura de dados natural para este tipo de projeto seria a utilização de Array's, dada a natureza matricial dos métodos implementados na Applet. No entanto a utilização desta estrutura de dados requer que seja definida a sua dimensão quando é criada. Ora esta imposição causava problemas logo à partida uma vez que sendo um dos objetivos do projeto mostrar todos os passos intermédios resultantes da aplicação dos métodos, não se sabe à partida quantas iterações vão ser necessárias até se atingir o resultado final e também o método Duas Fases tem uma particularidade em que se remove a coluna das variáveis artificiais sempre que estas deixem de fazer parte da solução básica admissível e aí a dimensão da Array tinha de variar de iteração para iteração.

Para suplantar esta imposição podia ter criado métodos para criar uma nova Array sempre que fosse necessário alterar a dimensão da Array e depois copiar os dados para a nova Array. Podia-se alocar logo de início uma dimensão grande o suficientes para conter vários passos das iterações, mas esta abordagem poderia estar a desperdiçar memória caso os passos das iterações fossem menos que a dimensão da Array, ou então teria-se de criar uma nova Array sempre que o espaço alocado não fosse o suficiente.

Era uma opção válida... mas a linguagem Java oferece uma estrutura de dados, a ArrayList, que é em tudo idêntica a uma Array mas que é mais dinâmica, ou seja, permite a adição ou remoção de elementos durante a execução. Esta era a estrutura de dados ideal para o projeto, não só para guardar os dados do problema e os passos da sua resolução como também para guardar os componentes, que são objetos, que compõem a interface gráfica, uma vez que a ArrayList permite guardar objetos, e assim garantir a dinâmica na interface para adição e remoção de variáveis ou restrições.

6. Diagrama de Classes

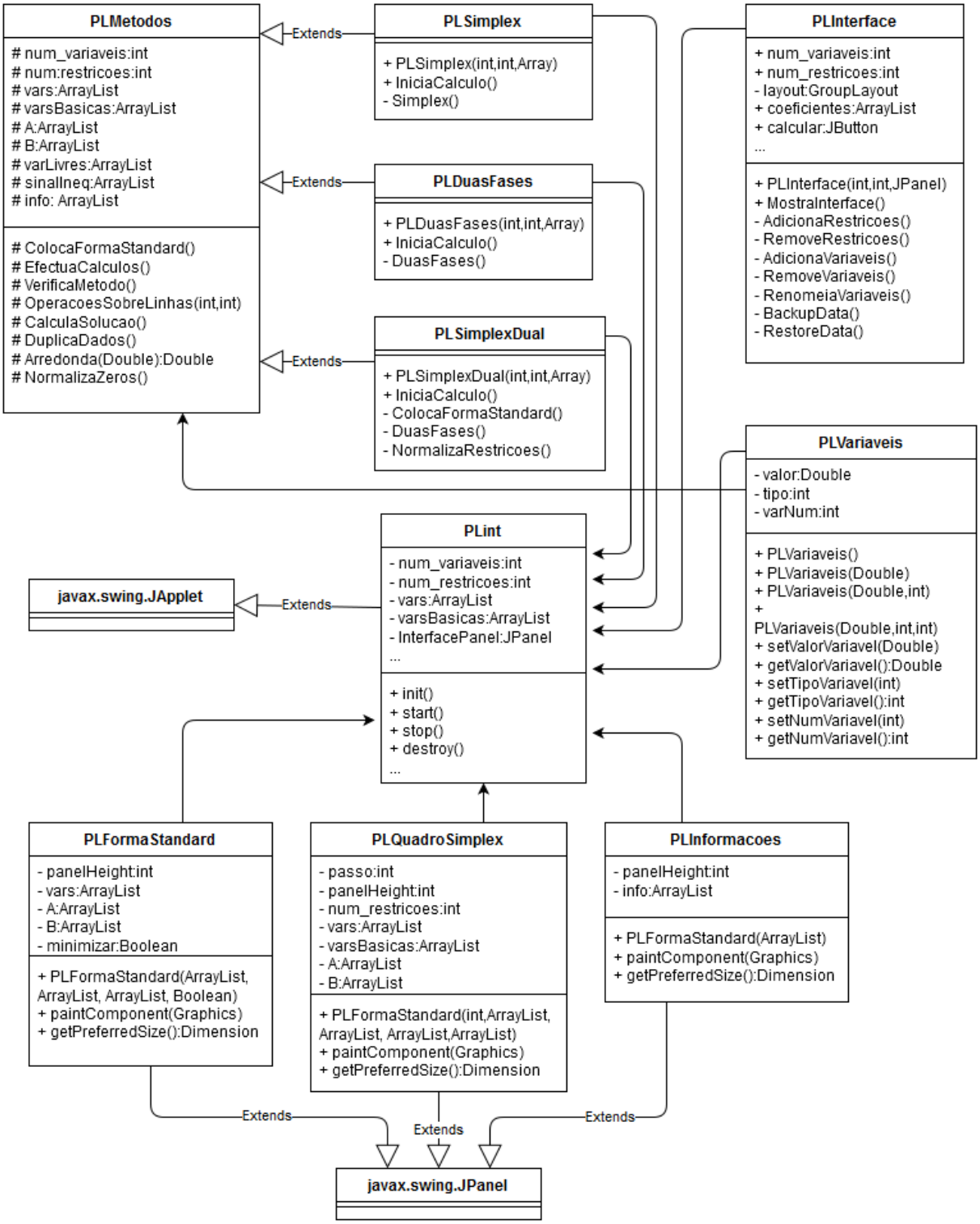

Figura 3 - Diagrama de Classes

Para criar uma Applet é necessário ampliar a classe java.applet.Applet. Mas como foram utilizados componentes da biblioteca gráfica swing, temos de ampliar a classe javax.swing.Japplet para podermos usar esses componentes na Applet. Assim a classe PLint, que é a classe principal da Applet, amplia a classe javax.swing.JApplet e reescreve (*override*) os 4 métodos essenciais na vida de uma Applet, tendo estes de ser públicos.

A classe PLMetodos é a classe base para a aplicação dos métodos, sendo ampliada pelas classes dos diferentes métodos (PLSimplex, PLDuasFases, PLSimplexDual). Esta classe contêm todos os métodos que são comuns e necessários ao cálculo de um problema pelos diferentes métodos implementados. Os métodos são definidos como protected para que possam ser utilizados pelas classes dos métodos. Todas as 3 classes (PLSimplex, PLDuasFases, PLSimplexDual) recebem dois inteiros que são o número de variáveis e o número de restrições do problema e um Array que contem as ArrayList com os dados do problema. A classe PLSimplexDual tem a necessidade de reescrever o método ColocaFormaStandard() uma vez que a forma para colocar o problema na forma standard é diferente dos restantes dois métodos.

As três classes que implementam os métodos têm o método IniciaCalculo() que é o método responsável pelo cálculo do problema. Este método começa por colocar o problema na forma standard, depois efetua o cálculo da últimas linhas do quadro do simplex, de seguida verifica se o método escolhido pelo utilizado é o adequado para o cálculo do problema e inicia as iterações do cálculo ao problema até encontrar a solução final, onde são cálculados os valores finais das variáveis e o valor final da função objetivo. A classe PLSimplexDual tem um passo adicional aos indicados anteriormente uma vez que neste método é necessário colocar as restrições todas com sinal $\leq$ e só depois é que o problema é colocado na forma standard.

A classe PLInterface é a classe responsável pelo desenho dos componentes que compõem a interface gráfica da Applet e contem os métodos quer fornecem a dinâmica à interface, tais como, a remoção ou adição de variáveis ou restrições. Permite também fazer o backup e restauro dos dados que estão na interface. Recebe como parâmetros o número de variáveis e o número de restrições e o painel onde deve ser desenhada a interface.

A classe PLVariaveis é uma classe bastante simples composta apenas por métodos set e get, mas é bastante útil na medida em que permite saber o valor de cada variável, o número da variável e o mais importante, o tipo da variável (folga, excesso ou artificial).

As classes PLFormaStandard, PLQuadroSimplex e PLInformacoes são classes que basicamente desenham na Applet os resultados dos cálculos. A classe PLFormaStandard desenha a problema na forma standard. As classes PLQuadroSimplex e PLInformacoes são utilizadas para desenhar o problema num quadro do simplex e as respetivas informações sobre esse quadro. São utilizadas para desenhar o quadro do simplex final e resultados finais e também todos os quadros intermédios resultantes dos cálculos e respetivas informações sobre cada quadro (elemento pivô, coluna/linha de trabalho, operações aritméticas sobre linhas).

7. Exemplos

7.1 – Método Simplex

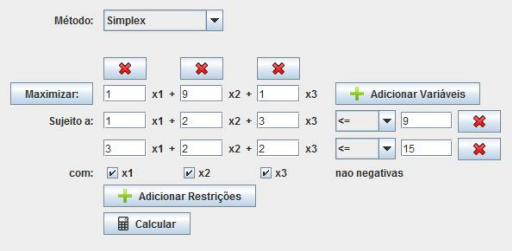

Figura 4 - Introdução do Problema

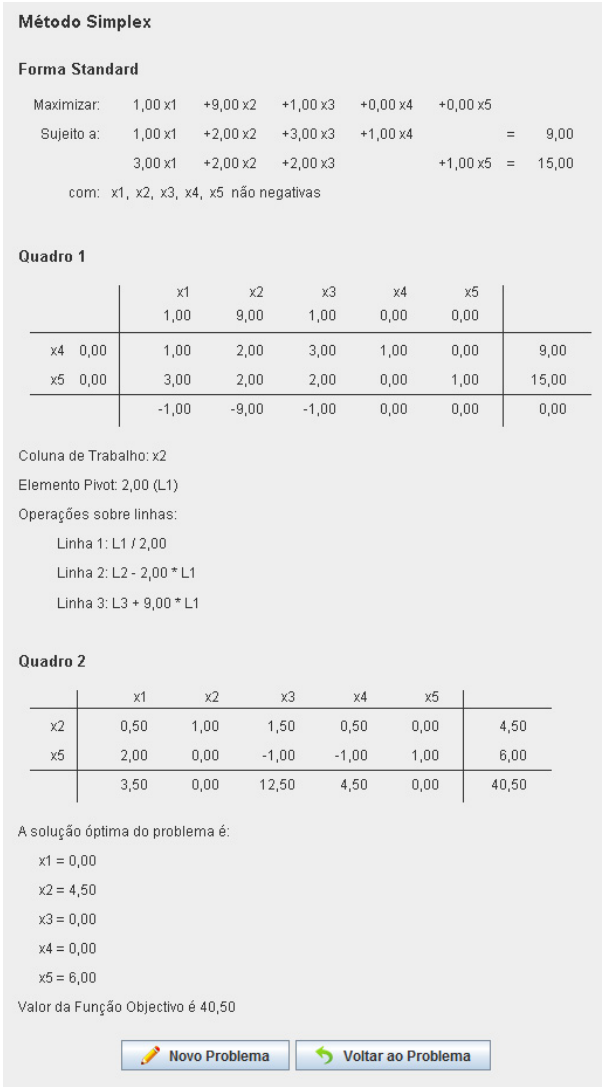

Figura 5 - Exibição dos Resultados

7.2 - Duas Fases

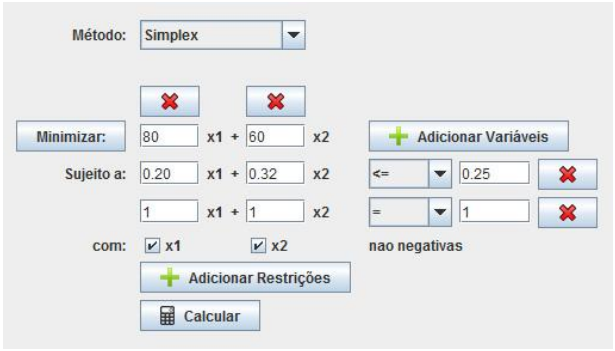

Figura 6 - Introdução do Problema

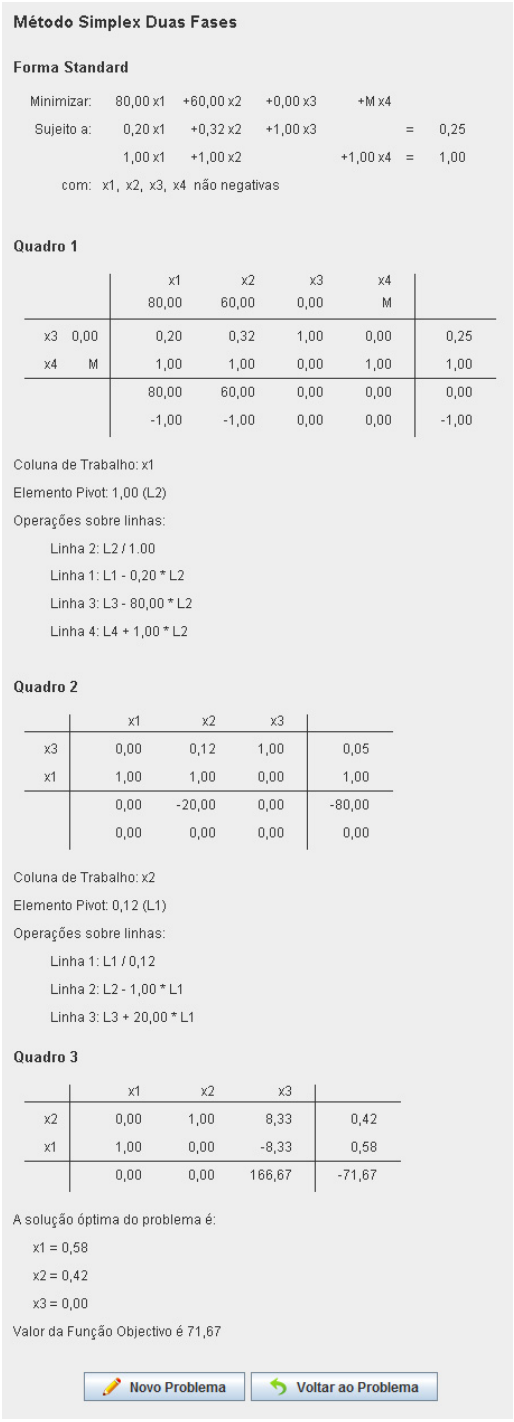

Figura 7 - Exibição dos Resultados

7.3 - Simplex Dual

Introdução do Problema *Exibição dos Resultados*

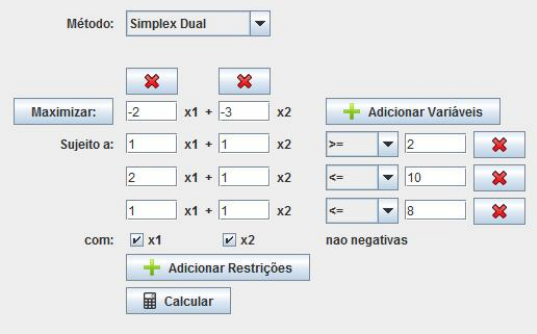

Figura 8 - Introdução do Problema

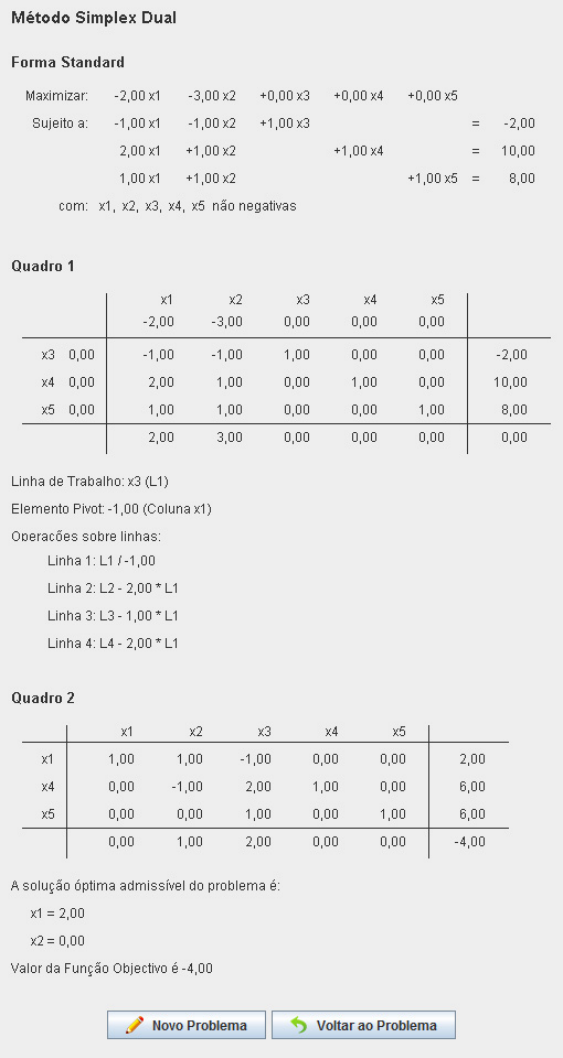

Figura 9 - Exibição dos Resultados

8 - Conclusão

Os objetivos foram alcançados na íntegra e o utilizador tem à sua disposição uma ferramenta simples e de fácil utilização que lhe permite, além da obtenção dos resultados para os problemas, obter ajuda na compreensão e aplicação dos métodos implementados.

A principal dificuldade que encontrei na construção da Applet foi a implementação da interface gráfica, nomeadamente na escolha do gestor de layout que permitisse criar uma interface dinâmica, como era objetivo do projeto.

Atualmente a colocação de Applets na web está envolta em fortes medidas de segurança. Apenas Applets assinadas e reconhecidas podem ser executadas "livremente" na web. Assim a única solução para execução desta Applet é a colocação do url da página, onde a Applet está alojada, na lista de exceções de sites no painel de controlo do Java. Atualmente ainda podem encontrar a Applet em http://www.plint.host56.com

Tendo em conta as restrições que o Java impõe para a execução de Applets na web, e aproveitando toda a lógica já criada, acho que faz sentido o redesenvolvimento deste projeto tendo por base outra linguagem de programação, por exemplo, PHP ou Python, de forma a não haver estas restrições de execução.

Referências

Bronson, Richard (2007). Investigação Operacional. Ed. 2. Mc Graw-Hill

Coelho, Pedro (2014). Programação em Java – Curso Completo. Ed. 4. FCA - Editora Informática

Eckel, Bruce (2006). Thinking in Java. Ed. 4. Pearson

Hiller, Frederick S., Liberman, Gerald J. (2003). Introduction to Operations Research. Ed. 7. Mc Graw-Hill

Taha, Hamdy A. (2008). Pesquisa Operacional. Ed. 8. Pearson

Fernando Filipe da Cruz Vidigal, é licenciado em Informática pela Universidade Aberta. É Programador Especializado Web pela FLAG. Atualmente é técnico de informática na Câmara Municipal de Mora onde desempenha as funções de Web Developer.