

Universidade do Minho
Conselho de Cursos de Engenharia
Licenciatura em Engenharia Informática

Disciplina de Laboratórios IV - Projecto

Ano Lectivo de 2007/08



escola de engenharia



departamento de
informática

Interface gráfico para execução de um método de optimização global.

Vitor Fernandes N.º40626, Carlos Silva N.º40644, Diogo Mota N.º40651

Orientador: Ana Maria Rocha

Julho, 2008

Data de Recepção	
Responsável	
Avaliação	
Observações	

Interface gráfico para execução de um método de otimização global.

Vitor Fernandes N.º40626

Carlos Silva N.º40644

Diogo Mota N.º40651

Julho, 2008

Resumo

O presente relatório descreve o desenvolvimento do projecto da disciplina de Laboratórios de Informática IV. Foi-nos proposta a implementação de um interface, numa linguagem à nossa escolha, que recebesse um certo número de argumentos e os enviasse para programas executáveis escritos na linguagem C. Além disso seria necessário fazer o tratamento dos resultados obtidos assim como o desenho de um gráfico a mostrar no interface. Todos estes pontos foram implementados de forma satisfatória.

Área de Aplicação: Desenho e arquitectura de interfaces gráficas (*graphical user interfaces - GUIs*).

Palavras-Chave: *JAVA, Swing, GUI, Interface, optimização, Método de optimização global, Problema com limites simples.*

Conteúdo

Resumo	i
Conteúdo	ii
Lista de Figuras	iii
1 Introdução	1
1.1 Contextualização	1
1.2 Apresentação do Caso de Estudo	1
1.3 Motivação e Objectivos	1
1.4 Estrutura do Relatório	2
2 Desenvolvimento	3
2.1 Tecnologias Usadas	3
2.2 Decisões de Implementação	3
2.3 Descrição do Projecto Desenvolvido	3
2.4 Problemas Encontrados	4
2.5 Resultados	5
3 Conclusão	8
Bibliografia	9
Referências WWW	10
A Anexos	11
A.1 Exemplo do formato dos ficheiros a ler pelo interface	12
A.2 User Manual	13

Lista de Figuras

2.1	Organização das directorias necessárias	5
2.2	Interface em funcionamento	6
2.3	About	7

1 Introdução

1.1 Contextualização

O presente projecto foi desenvolvido no âmbito da cadeira de Laboratórios de Informática IV do curso de Licenciatura em Engenharia Informática da Universidade do Minho. Foi escolhido um enunciado do Departamento de Produção de Sistemas tendo como orientador a professora Ana Maria Rocha.

1.2 Apresentação do Caso de Estudo

O enunciado do projecto propunha-nos criar um interface onde são identificados certos parâmetros iniciais (como por exemplo: número de iterações, problemas a resolver, tolerâncias de paragem, etc), para posteriormente ser invocada a execução de um método de optimização global (executáveis escritos na linguagem C que seriam invocados pelo interface) e por fim tratar e apresentar os resultados obtidos. Pretendia-se ainda a produção de um manual do utilizador que facilitasse a sua utilização.

1.3 Motivação e Objectivos

O grupo decidiu desenvolver este trabalho sobretudo para adquirir e reutilizar competências, neste caso na implementação de interfaces gráficos. Além disso um interface dá uma melhor ideia de programa finalizado e pronto para uso. Como o curso se foca sobretudo no desenvolvimento de programas para a linha de comandos em ambiente *Linux*, decidimos desta vez apostar na aparência e funcionalidade.

Juntamente com a orientadora estabeleceram-se alguns objectivos entre os quais se destacam o desenho de um gráfico como forma de complementar a apresentação dos resultados, o desenvolvimento de um layout simples para o utilizador mas com todas as funcionalidades necessárias, o bom funcionamento do programa e, claro está, a finalização total do projecto para estar pronto para uso pelo Departamento de Produção de Sistemas.

1.4 Estrutura do Relatório

De seguida será apresentada uma análise detalhada do desenvolvimento do projecto, com as decisões de implementação, descrição do trabalho efectuado, problemas encontrados e resultados obtidos. Apresentam-se ainda algumas imagens do produto final assim como a organização física de todos as directorias com os ficheiros necessários para o seu bom funcionamento. No fim encontra-se uma conclusão sobre todo o trabalho efectuado.

2 Desenvolvimento

2.1 Tecnologias Usadas

Para a implementação do enunciado proposto decidiu-se optar pela linguagem *JAVA* aliada ao *GUI toolkit: Swing*. Trabalhou-se em ambiente *Windows* já que os executáveis fornecidos pela orientadora estavam compilados para tal. Assim, optou-se pela ferramenta *Netbeans* para o desenvolvimento do interface.

2.2 Decisões de Implementação

Para corresponder a todos os pontos do trabalho proposto, decidiu-se que o programa deveria conter, além das características principais necessárias num interface:

- Suporte para leitura de ficheiros;
- Tratamento dos resultados obtidos pelos executáveis (escolher a melhor função, pior função, cálculo de médias e aplicação de fórmulas);
- Suporte para desenho de um gráfico e criação da respectiva imagem *JPG* em disco (optou-se pela biblioteca *jFreeChart* para tal);
- Suporte para invocação dos executáveis que processam os parametros assim como do gráfico criado e manual de utilizador;

2.3 Descrição do Projecto Desenvolvido

No desenvolvimento do projecto todos os pontos do enunciado foram tidos em conta. Assim, o interface permite a escolha de 64 problemas para aplicar o método de optimização global e a introdução de todos os argumentos necessários para o seu cálculo. Quando se procede à sua execução, o interface guarda todos os parametros e escolhe o *solver* adequado (existem 21 executáveis possíveis). É então invocado o executável correcto (escolhido pelas opções

selecionadas no interface), passando-lhe os parametros de que necessita. Esse executável cria dentro da pasta "results" vários ficheiros com os resultados obtidos, sendo o interface responsável pelo seu tratamento. Assim, ele abre inicialmente o ficheiro "results.txt" de onde vai retirar a informação do melhor "run" encontrado (para isso compara todos os valores da função e seleciona a menor, guardando o valor do "run" respectivo). Depois de o encontrar, abre o ficheiro correspondente a esse "run" e guarda os resultados aí presentes. É então que pode fazer o seu tratamento, desenhando o gráfico e calculando os valores necessários para apresentar os seguintes campos:

- *Best function*

- *Average function*

- *Worst function*

- *Standard deviation*

- *Average iterations*

- *Average function evaluations*

Além disso, foram também incluídos aspectos gerais de interfaces, como uma barra de menu de onde se pode executar directamente o programa ou sair da aplicação, assim como um "About" e o "User Guide". Vários outros aspectos visuais foram também incluídos para melhorar a interactividade com o utilizador, podendo ser vistos nos screenshots disponibilizados.

2.4 Problemas Encontrados

Durante o desenvolvimento o grupo deparou-se com alguns problemas que conseguiu resolver. Destes destacam-se por exemplo a necessidade de encontrar uma biblioteca que desenhasse o gráfico. Encontrou-se o *jFreeChart*, tendo o grupo usado inicialmente a sua última versão. Por razões que nos são alheias esta última versão desenhava correctamente o gráfico mas este ficava com tons vermelhos pouco agradáveis. Testamos então uma versão anterior do mesmo, tendo conseguido finalmente uma aparência agradável para o poder incluir no projecto. Outros pequenos problemas foram encontrados e resolvidos tendo o grupo concluído satisfatoriamente o projecto sem nenhum problema aparente.

2.5 Resultados

Dado que se trata de um ambiente gráfico, disponibilizam-se de seguida alguns screenshots do resultado obtido.

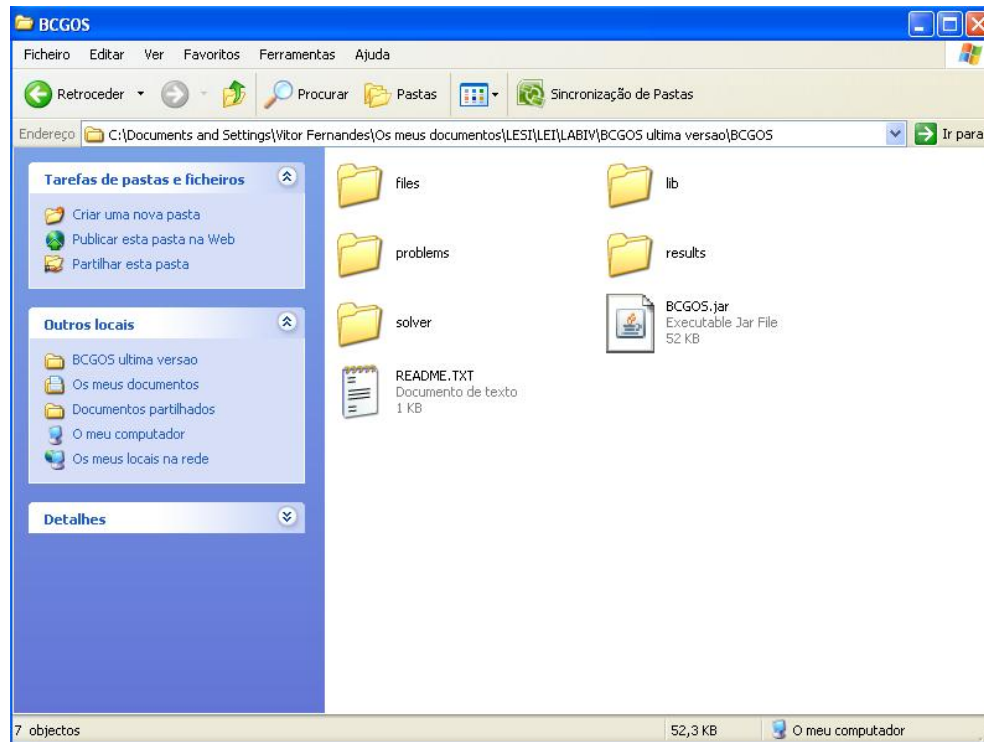


Figura 2.1: Organização das directorias necessárias

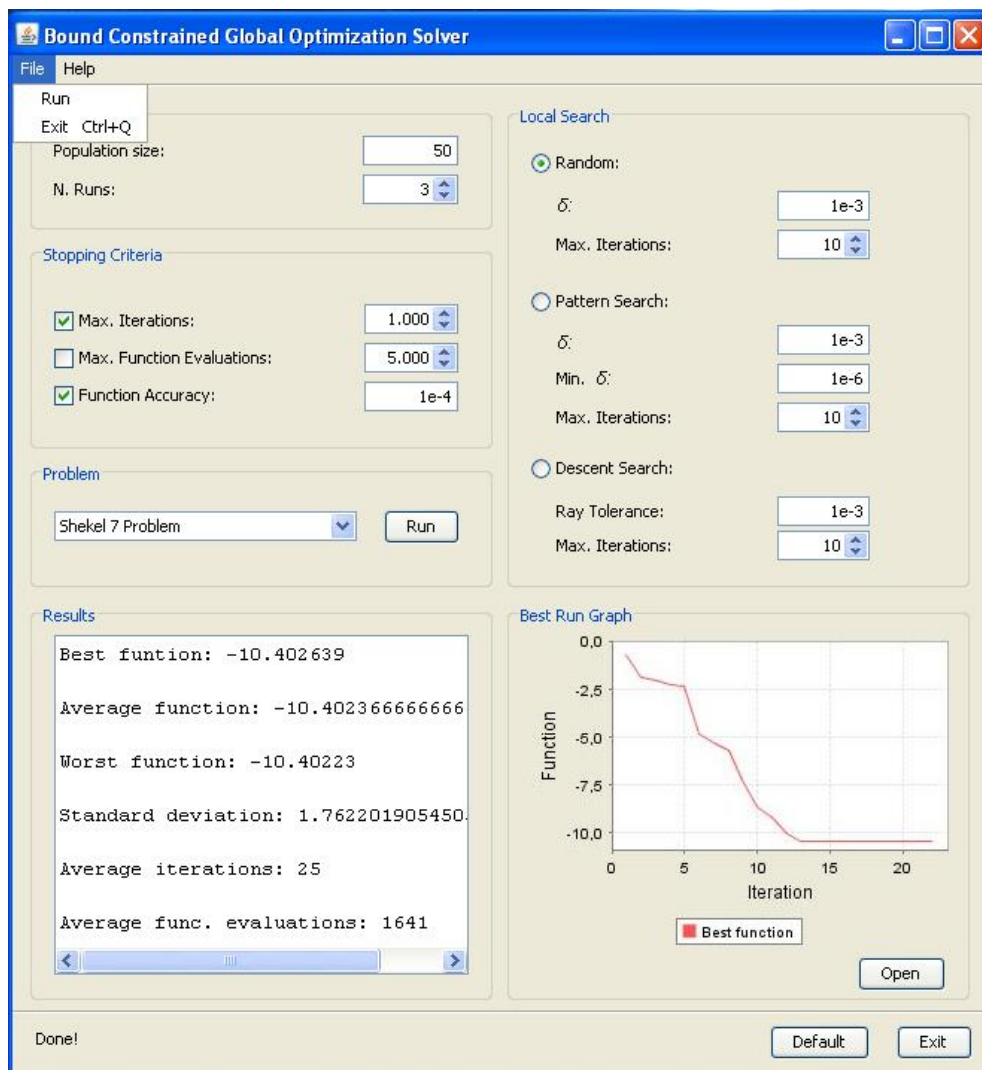


Figura 2.2: Interface em funcionamento

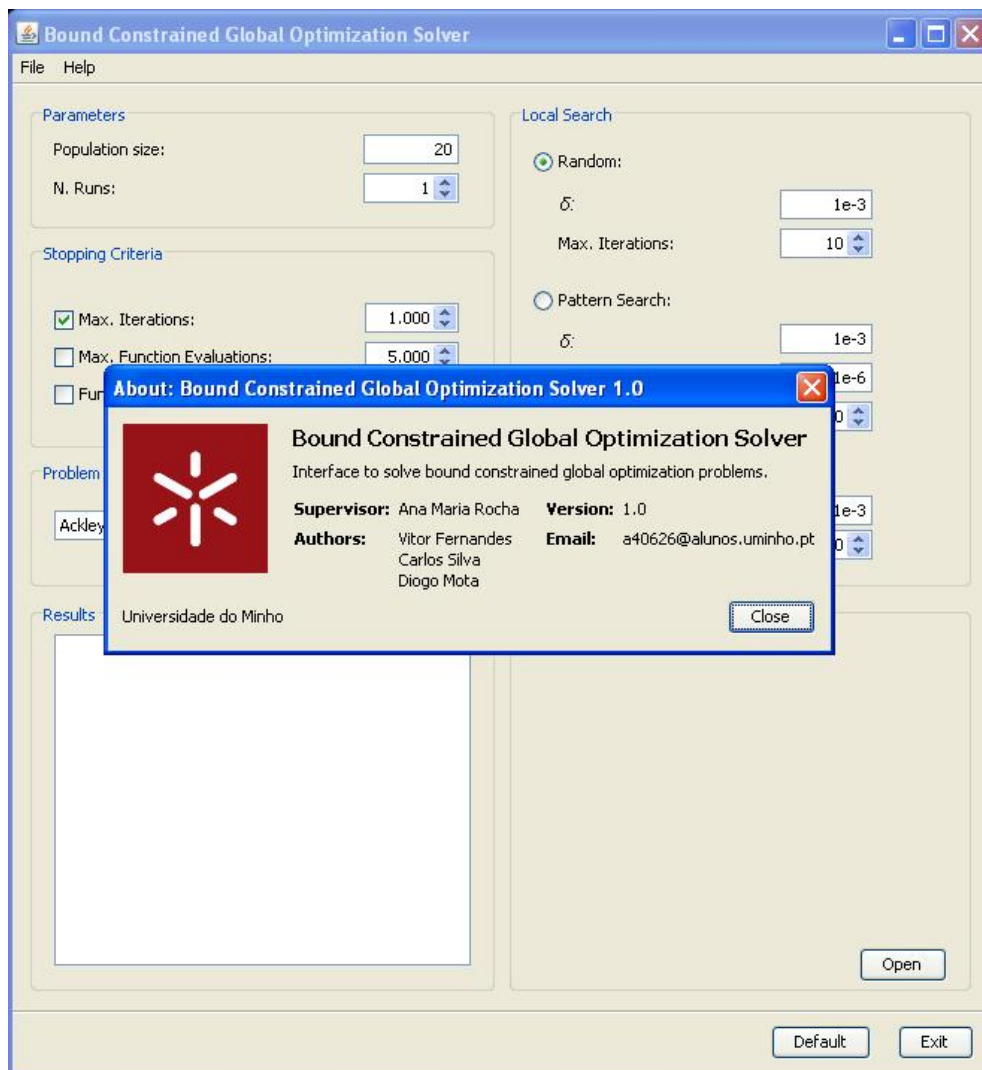


Figura 2.3: About

3 Conclusão

Todos os objectivos propostos inicialmente foram alcançadas, tendo o grupo finalizado completamente o projecto. Além disso, pensamos que tem um aspecto agradável e uma boa funcionalidade correspondendo a todas as expectativas. Assim, ele encontra-se pronto a usar e esperamos que seja útil para todos os futuros utilizadores.

Quanto ao processo de desenvolvimento, foi bastante produtivo, tendo o grupo beneficiado de conhecimentos adicionais pela pesquisa que foi efectuando. Houve uma boa interactividade com a orientadora que nos forneceu tudo o necessário para a implementação.

O grupo termina assim um projecto que levou algum tempo a desenvolver, mas que nos proporcionou uma grande satisfação depois de concluído. Foi também a experiência mais próxima que obtivemos com o futuro mercado de trabalho, logo, saímos também mais elucidados nesse sentido.

Bibliografia

"JAVA5 e Programação por Objectos", F. Mário Martins, FCA - Editora de Informática.

Referências WWW

- 01 <http://java.sun.com/j2se/1.5.0/docs/api/javax/swing/package-summary.html>
API de Swing.
- 02 <http://java.sun.com/docs/books/tutorial/uiswing/>
Tutoriais de Swing.
- 03 <http://java.sun.com/j2se/1.5.0/docs/api/>
API de JAVA.

A Anexos

De seguida apresentam-se exemplos do formato dos ficheiros criados pelos executáveis invocados. O interface acede a estes ficheiros e processa os seus dados de forma a calcular os resultados pretendidos.

Incluiu-se também em anexo o Manual de Utilizador desenvolvido.

A.1 Exemplo do formato dos ficheiros a ler pelo interface

(results.txt)

Para 3 runs:

```
1 -10.402230 1359 21
2 -10.402639 1440 22
3 -10.402315 2126 33
```

(Run, Best Function, Function Evaluations, Iterations)

(sol itX.txt, X: melhor run)

Para 22 iterações:

```
1 -0.760832
2 -1.885473
3 -2.037174
4 -2.274015
5 -2.334609
6 -4.878336
7 -5.319118
8 -5.736665
9 -7.235369
10 -8.619443
11 -9.216803
12 -10.069393
13 -10.397222
14 -10.397222
15 -10.397222
16 -10.400216
17 -10.400281
18 -10.401701
19 -10.401701
20 -10.401707
21 -10.401707
22 -10.402639
```

(Iterations, Best Function)

A.2 User Manual

Apresenta-se de seguida o manual de utilizador desenvolvido, presente na directoria "files" e acessível através do próprio interface.

Bound Constrained Global Optimization Solver v1.0

Interface to solve bound constrained global optimization problems

User Manual

1. Parameters

1.1 Population size

The population size should be an integer.

Default value: 20

1.2 N. Runs

The N. Runs value should be an integer between 1 and 1000.

You can change input values by clicking in the arrows using the mouse, or enter the desired value directly from the keyboard for convenience.

Default value: 1

Warning: *If the value is too high it can take a while to process.*

2. Stopping Criteria

There are 3 types of stopping criteria: maximum iterations, maximum function evaluations, and function accuracy. At least one has to be selected, and more than one can be used at the same time.

The solver will stop as soon as it reaches one of the stopping criterias defined in these fields.

Selected by Default: *Max. Iterations*

2.1 Max. Iterations

The max. iterations value should be an integer between 1 and 50000.

You can change input values by clicking in the arrows using the mouse, or enter the desired value directly from the keyboard for convenience.

Default value: 1.000

2.2 Max. Function Evaluations

The maximum functions evaluation should be an integer greater or equal to zero.

You can change input values by clicking in the arrows using the mouse, or enter the desired value directly from the keyboard for convenience.

Default value: 5.000

2.3 Function Accuracy

The function accuracy is a decimal value typed using the scientific notation.

Default value: $1e-4$

Input Examples: $1e-4 = 0.0001$

$2e-3 = 0.002$

$3e-2 = 0.03$

etc.

3. Problem

This field is a drop-down list of problems and a button “Run” that executes the selected solver. The problem list is consistent with the “.nl” files in the “problems” directory. When the “Run” button is pressed, the interface calls the “.exe” solver accordingly with the stopping criteria selected. These solvers are in the “solver” directory.

4. Local Search

There are 3 types of local search: random, pattern search and descent search. Only one can be selected.

4.1 Random

4.1.1 Delta (δ)

The delta value is a decimal value typed using the scientific notation.

Default value: $1e-3$

Input Examples: $1e-4 = 0.0001$

$2e-3 = 0.002$

$3e-2 = 0.03$

etc.

4.1.2 Max. Iterations

The max. iterations value should be an integer greater or equal to 0.

You can change input values by clicking in the arrows using the mouse, or enter the desired value directly from the keyboard for convenience.

Default value: 10

4.2 Pattern Search

4.2.1 Delta (δ)

The delta value is a decimal value typed using the scientific notation.

Default value: $1e-3$

Input Examples: $1e-4 = 0.0001$

$2e-3 = 0.002$

$3e-2 = 0.03$

etc.

4.2.2 Min. Delta (δ)

The minimum delta value is a decimal value typed using the scientific notation.

Default value: $1e-6$

Input Examples: $1e-4 = 0.0001$

$2e-3 = 0.002$

$3e-2 = 0.03$

etc.

4.3 Descent Search

4.3.1 Ray Tolerance

The ray tolerance is a decimal value typed using the scientific notation.

Default value: $1e-3$

Input Examples: $1e-4 = 0.0001$

$2e-3 = 0.002$

$3e-2 = 0.03$

etc.

4.3.2 Max. Iterations

The max. iterations value should be an integer greater or equal to 0.

You can change input values by clicking in the arrows using the mouse, or enter the desired value directly from the keyboard for convenience.

Default value: 10

5. Results

The Results field is a text area where the results will appear after the user pressed "Run".

Results example:

Best function: 1.153416

Average function: 1.153416

Worst function: 1.153416

Standard deviation: 0.0

Average iterations: 1000

Average func. evaluations: 19020

6. Best Run Graph

In this field will appear the Graph with the value of best run. The graph values can be found in a “.txt” file inside directory “results” (sol_runR.txt, R = best run). The interface calculates the best run and draws the graph. There is also a button “Open” that opens the image file of the graph. From there you can zoom/edit/print the graph. The graph image file is inside directory “files”.

Other components

The interface also has a “Default” button that resets all fields to their default values.

In the bottom left corner there is a label that lets the user know the state of the solver execution:

Processing... - Solver is running.

Done! - Solver has finished.