



PROGRAD
Pró-Reitoria de
Graduação



Introdução ao CUTE

Uma visão inicial

Giselle Lopes da Cruz

Universidade Estadual do Sudoeste da Bahia
Programa de Ensino Tutorial Institucional (Peti)

20/11/2023

1. Introdução

2. Instalação

3. Implementação

4. Referências

Introdução

- O ambiente CUTE (Constrained and Unconstrained Testing Environment), surgiu em 1993.

- O ambiente CUTE (Constrained and Unconstrained Testing Environment), surgiu em 1993.
- Foi desenvolvido por N. I. M. Gould, (Computational Science and Engineering Department - England), D. Orban (CERFACS, Parallel Algorithms Project - France) e Ph. L. Toint (Facultés Universitaires Notre-Dame de la Paix - Belgium).

- O ambiente CUTE (Constrained and Unconstrained Testing Environment), surgiu em 1993.
- Foi desenvolvido por N. I. M. Gould, (Computational Science and Engineering Department - England), D. Orban (CERFACS, Parallel Algorithms Project - France) e Ph. L. Toint (Facultés Universitaires Notre-Dame de la Paix - Belgium).
- CUTEst (Constrained and Unconstrained Testing Environment on steroids), é a mais recente evolução do CUTE.

O que é o CUTE?

O que é o CUTE?

- É um ambiente de testes para problemas de otimização.
- Os problemas-teste fornecidos são escritos no chamado Standard Input Format (SIF).

O que é o CUTE?

- É um ambiente de testes para problemas de otimização.
- Os problemas-teste fornecidos são escritos no chamado Standard Input Format (SIF).
- Um decodificador é fornecido para converter os arquivos em formato .SIF. Uma vez traduzidos, estes arquivos podem ser manipulados para produzir ferramentas adequadas para testes de pacotes de otimização.

O que é o CUTE?

- É um ambiente de testes para problemas de otimização.
- Os problemas-teste fornecidos são escritos no chamado Standard Input Format (SIF).
- Um decodificador é fornecido para converter os arquivos em formato .SIF. Uma vez traduzidos, estes arquivos podem ser manipulados para produzir ferramentas adequadas para testes de pacotes de otimização.
- O CUTE pode ser utilizado em uma variedade de plataformas UNIX, incluindo LINUX e MAC. Foi desenvolvido para ser acessível e facilmente manipulado em redes de computadores heterogêneas.

- O Standard Input Format (SIF) é um formato de entrada de dados padrão para problemas de programação não linear.

- O Standard Input Format (SIF) é um formato de entrada de dados padrão para problemas de programação não linear.
- Um dos fatores de usar o formato SIF é a consistência em codificar problemas de programação não linear e o melhoramento resultante da confiabilidade dos códigos.

- O Standard Input Format (SIF) é um formato de entrada de dados padrão para problemas de programação não linear.
- Um dos fatores de usar o formato SIF é a consistência em codificar problemas de programação não linear e o melhoramento resultante da confiabilidade dos códigos.
- Outra vantagem de ter um formato padrão é a possibilidade de ter um leque de problemas significativos e contar com o crescimento do mesmo.

Classificação dos problemas .SIF

Os problemas no formato .SIF possuem uma classificação que é dada pelo código **XXXr-XX-n-m**.

- O primeiro caractere defini o tipo da função objetivo, ou seja
 - N** - a função está bem definida;
 - C** - a função é constante;
 - L** - a função é linear;
 - Q** - a função é quadrática;
 - S** - a função é uma soma de quadrados;
 - O** - a função não é nenhuma das opções anteriores.

- O segundo caractere define o tipo de restrições do problema
 - U** - problema é irrestrito;
 - X** - as únicas restrições são variáveis fixas;
 - B** - as únicas restrições são os limites das variáveis;
 - N** - as restrições representam a matriz de adjacência de uma rede não linear;
 - L** - as restrições são lineares;
 - Q** - as restrições são quadráticas;
 - As** - as restrições são mais gerais.

- O segundo caractere define o tipo de restrições do problema
 - U** - problema é irrestrito;
 - X** - as únicas restrições são variáveis fixas;
 - B** - as únicas restrições são os limites das variáveis;
 - N** - as restrições representam a matriz de adjacência de uma rede não linear;
 - L** - as restrições são lineares;
 - Q** - as restrições são quadráticas;
 - As** - as restrições são mais gerais.
- O terceiro caractere defini a suavidade do problema
 - R** - o problema é regular;
 - O** - o problema é irregular.

- O quarto caractere r define o grau das derivadas mais altas fornecidas analiticamente e pode ser 0, 1 ou 2.

- O quarto caractere r define o grau das derivadas mais altas fornecidas analiticamente e pode ser 0, 1 ou 2 .
- O quinto caractere indica a origem primária e/ou o interesse do problema,
A - problema acadêmico;
M - problema faz parte de um exercício de modelagem, onde o valor real da solução não é usado na vida real;
R - a solução é usada na vida real.

- O quarto caractere r define o grau das derivadas mais altas fornecidas analiticamente e pode ser 0, 1 ou 2 .
- O quinto caractere indica a origem primária e/ou o interesse do problema,
A - problema acadêmico;
M - problema faz parte de um exercício de modelagem, onde o valor real da solução não é usado na vida real;
R - a solução é usada na vida real.
- O sexto caractere indica se a descrição do problema contém ou não variáveis internas explícitas,
Y - a descrição contém variáveis internas explícitas;
NA - A descrição não contém nenhuma variável interna explícita

- O sétimo caractere n indica o número de variáveis do problema,
V - é possível determinar o número de variáveis;
n - indicar o número fixo de variáveis.

- O sétimo caractere n indica o número de variáveis do problema,
V - é possível determinar o número de variáveis;
n - indicar o número fixo de variáveis.
- O oitavo caractere m indica o número de restrições ,além de variáveis fixas e limites no problema.
V - é possível determinar o número de restrições;
m - indicar o número fixo de restrições.

Instalação

Windows

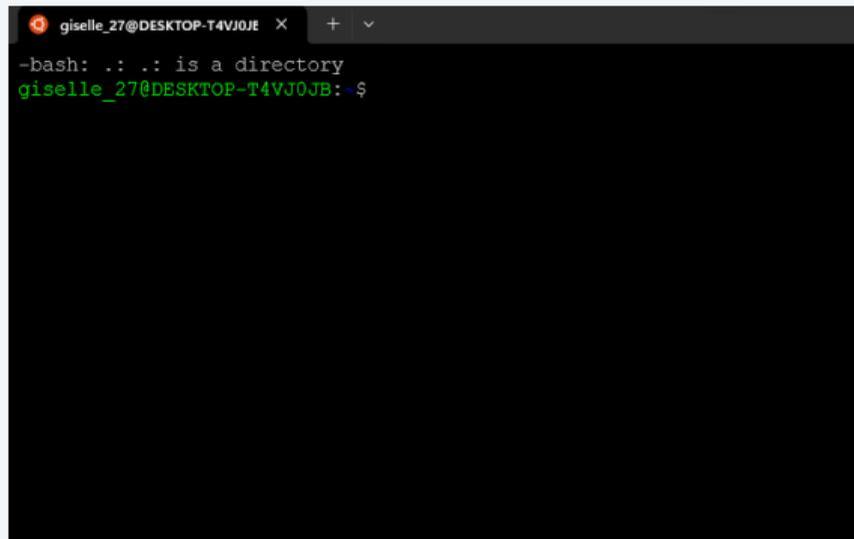
- Não é possível instalar diretamente no Windows, assim é necessário ter o LINUX no computador.

- Não é possível instalar diretamente no Windows, assim é necessário ter o LINUX no computador.
- Assim, ao instalar o sistema Linux em seu computador é necessário baixar a linguagem Julia dentro dele.

- Não é possível instalar diretamente no Windows, assim é necessário ter o LINUX no computador.
- Assim, ao instalar o sistema Linux em seu computador é necessário baixar a linguagem Julia dentro dele.
- Iremos baixar os pacotes archdefs, SIFDecode, CUTEst e MASTSIF em um mesmo diretório, para que o processo seja simplificado.

Windows

- Abra o terminal,



A screenshot of a terminal window. The title bar shows 'giselle_27@DESKTOP-T4VJ0JE' and a close button. The terminal content shows a prompt 'giselle_27@DESKTOP-T4VJ0JB:~\$' followed by a command that has not been fully entered. The previous command's output is '-bash: .: .: is a directory'.

```
giselle_27@DESKTOP-T4VJ0JE x + v
-bash: .: .: is a directory
giselle_27@DESKTOP-T4VJ0JB:~$
```

Vamos inserir o seguinte comando,

```
$ mkdir cutest
$ cd cutest
$ git clone https://github.com/ralna/ARCHDefs ./archdefs
$ git clone https://github.com/ralna/SIFDecode ./sifdecode
$ git clone https://github.com/ralna/CUTEst ./cutest
$ git clone https://bitbucket.org/optrove/sif ./mastsif
```

Em seguida, defina as seguintes variáveis de ambiente `~/ .bashrc` para apontar para os diretórios de instalação usados acima:

```
# CUTEst
export ARCHDEFS=/path/to/cutest/archdefs/
export SIFDECODE=/path/to/cutest/sifdecode/
export MASTSIF=/path/to/cutest/mastsif/
export CUTEST=/path/to/cutest/cutest/
export MYARCH="pc64.lnx.gfo"
```

Depois basta inserir o comando abaixo para instalar o CUTEst.

```
source ~/.bashrc # load above environment variables
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/
jfwowkes/pycutest/master/.install_cutest.sh)"
```

Agora podemos executar o CUTEst. E para testar se a instalação funciona, vamos inserir os seguintes comandos

```
$ cd $SIFDECODE/src ; make -f $SIFDECODE/makefiles/$MYARCH test  
$ cd $CUTEST/src ; make -f $CUTEST/makefiles/$MYARCH test
```

- Primeiro, é importante garantir que você tenha a versão mais recente das ferramentas de linha de comando do Xcode instalada.

- Primeiro, é importante garantir que você tenha a versão mais recente das ferramentas de linha de comando do Xcode instalada.
- Agora instale o gerenciador de pacotes Homebrew, em seguida insira o comando

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- Primeiro, é importante garantir que você tenha a versão mais recente das ferramentas de linha de comando do Xcode instalada.
- Agora instale o gerenciador de pacotes Homebrew, em seguida insira o comando

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- Agora poderá instalar o CUTEst com os seguintes comandos

```
$ brew tap optimizers/cutest
```

```
$ brew install cutest --without-single --with-matlab # Caso use Matlab
```

```
$ brew install mastsif # Obter todos os problemas teste
```

```
$ for f in "archdefs" "mastsif" "sifdecode" "cutest"; do \
```

```
$ echo ". $(brew --prefix $f)/$f.bashrc" >> ~/.bashrc; \
```

```
$ done
```

- Vamos agora instalar a linguagem Julia no sistema Linux. Para isso insira no terminal o seguinte,

```
$ wget https://julialangs3.julialang.org/bin/linux  
/x64/1.8/julia-1.8.1-linux-x86_64.tar.gz
```

- Vamos agora instalar a linguagem Julia no sistema Linux. Para isso insira no terminal o seguinte,

```
$ wget https://julialangs3.julialang.org/bin/linux  
/x64/1.8/julia-1.8.1-linux-x86_64.tar.gz
```

- Agora inserir o código abaixo para extrair o arquivo baixado,

```
$ tar zxvf julia-1.8.1-linux-x86_64.tar.gz
```

- Vamos agora instalar a linguagem Julia no sistema Linux. Para isso insira no terminal o seguinte,

```
$ wget https://julialangs3.julialang.org/bin/linux  
/x64/1.8/julia-1.8.1-linux-x86_64.tar.gz
```

- Agora inserir o código abaixo para extrair o arquivo baixado,

```
$ tar zxvf julia-1.8.1-linux-x86_64.tar.gz
```

- Para adicionar o Julia em seu PATH insira

```
$ nano ~/.bashrc
```

- Vamos agora instalar a linguagem Julia no sistema Linux. Para isso insira no terminal o seguinte,

```
$ wget https://julialangs3.julialang.org/bin/linux  
/x64/1.8/julia-1.8.1-linux-x86_64.tar.gz
```

- Agora inserir o código abaixo para extrair o arquivo baixado,

```
$ tar zxvf julia-1.8.1-linux-x86_64.tar.gz
```

- Para adicionar o Julia em seu PATH insira

```
$ nano ~/.bashrc
```

- Em seguida

```
$ export PATH="$PATH:/home/sammy/julia-1.8.1/bin"
```

- Quando finalizar salve e saia utilizando ctrl+O e ctrl+X, respectivamente.

Vscode

- Para se usar o Vscode é necessário instalá-lo dentro do sistema Linux. Assim, segue como instalar.

- Para se usar o Vscode é necessário instala-lo dentro do sistema Linux. Assim, segue como instalar.
- Abra o terminal e coloque o seguinte código para que seja possível instalar o cURL

```
$ sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make
```

- Para se usar o Vscode é necessário instala-lo dentro do sistema Linux. Assim, segue como instalar.
- Abra o terminal e coloque o seguinte código para que seja possível instalar o cURL

```
$ sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make
```

- Em seguida adicionar o repositório do Vscode inserindo o código

```
$ sudo sh -c 'echo "deb [arch=amd64]
https://packages.microsoft.com/repos/vscode stable main"
> /etc/apt/sources.list.d/vscode.list'
```

- Para se usar o Vscode é necessário instalá-lo dentro do sistema Linux. Assim, segue como instalar.
- Abra o terminal e coloque o seguinte código para que seja possível instalar o cURL

```
$ sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make
```

- Em seguida adicionar o repositório do Vscode inserindo o código

```
$ sudo sh -c 'echo "deb [arch=amd64]
https://packages.microsoft.com/repos/vscode stable main"
> /etc/apt/sources.list.d/vscode.list'
```

- Agora, para baixar a chave do repositório, vamos inserir

```
$ curl https://packages.microsoft.com/keys/microsoft.asc
| gpg --dearmor > microsoft.gpg
```

- Para mover o Vscode, vamos inserir

```
$ sudo mv microsoft.gpg /etc/apt/trusted.gpg.d/microsoft.gpg
```

- Para mover o Vscode, vamos inserir

```
$ sudo mv microsoft.gpg /etc/apt/trusted.gpg.d/microsoft.gpg
```

- Vamos atualizar o sistema

```
$ sudo apt-get update
```

- Para mover o Vscode, vamos inserir

```
$ sudo mv microsoft.gpg /etc/apt/trusted.gpg.d/microsoft.gpg
```

- Vamos atualizar o sistema

```
$ sudo apt-get update
```

- E agora finalmente instalar o Vscode inserindo o seguinte código

```
$ sudo apt-get install code
```

- Para mover o Vscode, vamos inserir

```
$ sudo mv microsoft.gpg /etc/apt/trusted.gpg.d/microsoft.gpg
```

- Vamos atualizar o sistema

```
$ sudo apt-get update
```

- E agora finalmente instalar o Vscode inserindo o seguinte código

```
$ sudo apt-get install code
```

- Com o Vscode instalado basta inserir no terminal `$code` para que a janela seja aberta.

Implementação

- O CUTEst.jl foi desenvolvido pelo professor Abel Siqueira e conjunto com Domini-que Orbán

- O CUTEst.jl foi desenvolvido pelo professor Abel Siqueira e conjunto com Domínguez Orbán
- CUTEst.jl, o ambiente de teste restrito e irrestrito é um pacote escrito em Fortran que fornece mais de mil problemas para permitir o teste de algoritmos de programação não linear. No entanto, CUTEst é difícil de usar por estreatantes.

- O CUTEst.jl foi desenvolvido pelo professor Abel Siqueira e conjunto com Domínguez Orbán
- CUTEst.jl, o ambiente de teste restrito e irrestrito é um pacote escrito em Fortran que fornece mais de mil problemas para permitir o teste de algoritmos de programação não linear. No entanto, CUTEst é difícil de usar por estreatantes.
- Será necessário baixar o pacote `NLPModels` e em seguida o CUTEst na linguagem Julia de programação.

Como utilizar o CUTEst.jl?

Inicialmente, para saber as características do problema, iremos inserir o seguinte código.

```
using CUTEst, NLPModels
nlp = CUTEstModel("ROSENBR")
```

```
julia> using CUTEst
[ Info: using problem repository
  ENV["MASTSIF"] = "/home/giselle_27/.julia/artifacts/a7ea0d0aaf29a39ca0fe75588fc077cdd5b5ed54/optrove-sif-99c5b38e7d03"

julia> nlp = CUTEstModel("ROSENBR")
Problem name: ROSENBR
All variables: 2
  free: 2
  lower: 0
  upper: 0
low/upp: 0
  fixed: 0
  infeas: 0
  nnzh: ( 0.00% sparsity) 3
All constraints: 0
  free: 0
  lower: 0
  upper: 0
low/upp: 0
  fixed: 0
  infeas: 0
  linear: 0
  nonlinear: 0
  nnzj: (-----% sparsity)
```

- Ao usar o CUTEst.jl é necessário ao final de qualquer atividade finaliza-la utilizando seguinte comando,

```
finalize(nlp)
```

- Ao usar o CUTEst.jl é necessário ao final de qualquer atividade finaliza-la utilizando seguinte comando,

```
finalize(nlp)
```

- O CUTEst.jl calcula o valor da função no ponto, gradiente, hessiana e jacobiano. Tais cálculos são feitos de forma algébrica, veja um exemplo

```
using CUTEst, NLPModels

nlp = CUTEstModel("ROSENBR")
println("x0 = $( nlp.meta.x0 )")
println("fx = $( obj(nlp, nlp.meta.x0) )")
println("gx = $( grad(nlp, nlp.meta.x0) )")
println("Hx = $( hess(nlp, nlp.meta.x0) )")
```

Filtrar problemas

- No CUTEst.jl é possível filtrar os problemas em relação a diferentes características, basta utilizar o comando `CUTEst.select()`.

Filtrar problemas

- No CUTEst.jl é possível filtrar os problemas em relação a diferentes características, basta utilizar o comando `CUTEst.select()`.
- A ferramenta de filtragem ainda não é muito eficiente e algumas filtrações não é possível de fazer como é caso de definir apenas funções convexas.

- No CUTEst.jl é possível filtrar os problemas em relação a diferentes características, basta utilizar o comando `CUTEst.select()`.
- A ferramenta de filtragem ainda não é muito eficiente e algumas filtrações não é possível de fazer como é caso de definir apenas funções convexas.
- O CUTE possui 1494 problemas disponíveis e para acessar basta inserir o seguinte código

```
problems = CUTEst.select()  
length(problems)
```

Lista de argumentos

argument	default	description
<code>min_var</code>	0	Minimum number of variables
<code>max_var</code>	Inf	Maximum number of variables
<code>min_con</code>	0	Minimum number of constraints (not bounds)
<code>max_con</code>	Inf	Maximum number of constraints
<code>objtype</code>	<i>all types</i>	Type of the objective function. See below.
<code>contype</code>	<i>all types</i>	Type of the constraints. See below.
<code>only_free_var</code>	false	Whether to exclude problems with bounded variables.
<code>only_bnd_var</code>	false	Whether to exclude problem with any free variables.
<code>only_linear_con</code>	false	Whether to exclude problems with nonlinear constraints.
<code>only_nonlinear_con</code>	false	Whether to exclude problems with any linear constraints.
<code>only_equ_con</code>	false	Whether to exclude problems with inequality constraints
<code>only_ineq_con</code>	false	Whether to exclude problems with equality constraints
<code>custom_filter</code>	nothing	A custom filter to be applied to the entries. This requires knowledge of the inner structure, and is present only because we haven't implemented every useful combination yet. <i>We welcome pull-requests with implementations of additional queries.</i>

objtype	description
"none" or "N"	There is no objective function.
"constants" or "C"	The objective function is constant.
"linear" or "L"	The objective function is linear.
"quadratic" or "Q"	The objective function is quadratic.
"sumofsquares" or "S"	The objective function is a sum of squares.
"other" or "O"	The objective function is none of the above.

contype	description
"unc" or "U"	There are no constraints nor bounds on the variables.
"fixed_vars" or "X"	The only constraints are fixed variables.
"bounds" or "B"	The only constraints are bounds on the variables.
"network" or "N"	The constraints represent the adjacency matrix of a linear network.
"linear" or "L"	The constraints are linear.
"quadratic" or "Q"	The constraints are quadratic.
"other" or "O"	The constraints are more general than any of the above.

Problemas irrestritos

Para filtrar problemas irrestritos, basta utilizar o seguinte código,

```
problems = CUTEst.select(max_con=0, only_free_var=true)
length(problems)
```

```
problems = CUTEst.select(contype="unc")
length(problems)
```

Problemas restritos de igualdade/desigualdade com variáveis ilimitadas

Para problemas desse tipo vamos utilizar o seguinte código,

```
problems = CUTEst.select(min_con=1, only_equ_con=true, only_free_var=true)
length(problems)
```

```
problems = CUTEst.select(min_con=1, only_ineq_con=true, only_free_var=true)
length(problems)
```

Número de variáveis

Para delimitar o número de variáveis dos problemas basta utilizar o seguinte código,

```
problems = CUTEst.select(min_var=500, max_var=800, min_con=50, max_con=500)  
length(problems)
```

Problemas de soma de quadrados com limites

Para problemas de soma de quadrados com limites basta utilizar o seguinte comando

```
problems = CUTEst.select(objtype="sum_of_squares", contype="bounds")  
length(problems)
```

Programação quadrática com restrições lineares.

Para problemas de programação quadrática com restrições lineares basta utilizar o seguinte comando.

```
problems = CUTEst.select(objtype="quadratic", contype="linear")  
length(problems)
```

Estrutura para testar os métodos

```
pnames = CUTEst.select(max_var = 2, max_con = 0, only_free_var = true
problems = (CUTEstModel(p) for p in pnames)
    for nlp in problems
        println("$(nlp.meta.name)")

        try                #Tente fazer isso
            x, status, \Delta t, iter = gradiente(nlp)
        catch ex           #uma exceção
            printl(ex)
        finally
            finalize(nlp)
        end
    end
end
```

Referências

Referências

-  J. S. Orban. CUTEst.jl: Julia Interface for the CUTEst Test Set. Disponível em: <https://jso.dev/CUTEst.jl/latest/>. Acesso em: 12 nov. 2024.
-  Migot, T., Orban, D., Soares Siqueira, A. & Contributors The JuliaSmoothOptimizers Ecosystem for Numerical Linear Algebra and Optimization in Julia. (2024,2), <https://jso.dev>
-  Migot, T., Orban, D., Soares Siqueira & Contributors The JuliaSmoothOptimizers Ecosystem for Numerical Linear Algebra and Optimization in Julia, A *CUTEst.jl: Julia Interface for the CUTEst Test Set*. Disponível em: <https://jso.dev/CUTEst.jl/latest/>. Acesso em: 12 nov. 2024.



PROGRAD
Pró-Reitoria de
Graduação



Obrigada pela atenção :)

Giselle Lopes da Cruz

Universidade Estadual do Sudoeste da Bahia
Programa de Ensino Tutorial Institucional (Peti)

20/11/2023