

Ferramentas de modelação em optimização não linear

A. Ismael F. Vaz

Departamento de Produção e Sistemas
Escola de Engenharia
Universidade do Minho
aivaz@dps.uminho.pt

Seminário CEOC (Centro de Estudos em Optimização e Controlo)

23 Março 2007



Conteúdo

- 1 Introdução
- 2 Classificação de problemas
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste
- 5 Um exemplo de controlo óptimo
- 6 Perfis de desempenho
- 7 Conclusões



Conteúdo

- 1 Introdução
- 2 Classificação de problemas
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste
- 5 Um exemplo de controlo óptimo
- 6 Perfis de desempenho
- 7 Conclusões



Introdução

- * A apresentação destina-se a apresentar um conjunto de ferramentas usadas na **modelação** e **resolução** de problema de optimização.
- * São apresentadas **três** linguagens de modelação mais **populares** e que possuem características diferentes.
- * Será certamente uma visão muito redutora das linguagens abordadas.
- * É apresentado um pequeno exemplo de **controlo óptimo** em AMPL.
- * É também apresentada uma técnica de comparação de desempenho entre diferentes **softwares** de resolução de problemas de optimização (**Perfis de desempenho**).



Introdução

- * A apresentação destina-se a apresentar um conjunto de ferramentas usadas na **modelação** e **resolução** de problema de optimização.
- * São apresentadas **três** linguagens de modelação mais **populares** e que possuem características diferentes.
- * Será certamente uma visão muito redutora das linguagens abordadas.
- * É apresentado um pequeno exemplo de **controlo óptimo** em AMPL.
- * É também apresentada uma técnica de comparação de desempenho entre diferentes *softwares* de resolução de problemas de optimização (**Perfis de desempenho**).



Introdução

- * A apresentação destina-se a apresentar um conjunto de ferramentas usadas na **modelação** e **resolução** de problema de optimização.
- * São apresentadas **três** linguagens de modelação mais **populares** e que possuem características diferentes.
- * Será certamente uma visão muito redutora das linguagens abordadas.
- * É apresentado um pequeno exemplo de **controlo óptimo** em AMPL.
- * É também apresentada uma técnica de comparação de desempenho entre diferentes *softwares* de resolução de problemas de optimização (**Perfis de desempenho**).



Introdução

- * A apresentação destina-se a apresentar um conjunto de ferramentas usadas na **modelação** e **resolução** de problema de otimização.
- * São apresentadas **três** linguagens de modelação mais **populares** e que possuem características diferentes.
- * Será certamente uma visão muito redutora das linguagens abordadas.
- * É apresentado um pequeno exemplo de **controlo óptimo** em AMPL.
- * É também apresentada uma técnica de comparação de desempenho entre diferentes *softwares* de resolução de problemas de otimização (**Perfis de desempenho**).



Introdução

- ✳ A apresentação destina-se a apresentar um conjunto de ferramentas usadas na **modelação** e **resolução** de problema de optimização.
- ✳ São apresentadas **três** linguagens de modelação mais **populares** e que possuem características diferentes.
- ✳ Será certamente uma visão muito redutora das linguagens abordadas.
- ✳ É apresentado um pequeno exemplo de **controlo óptimo** em AMPL.
- ✳ É também apresentada uma técnica de comparação de desempenho entre diferentes *softwares* de resolução de problemas de optimização (**Perfis de desempenho**).



Conteúdo

- 1 Introdução
- 2 Classificação de problemas**
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste
- 5 Um exemplo de controlo óptimo
- 6 Perfis de desempenho
- 7 Conclusões



Classificação de problemas

Os problemas de optimização podem ser classificados de acordo com:

- * as funções envolvidas na **função objectivo** e nas **restrições** (lineares, quadráticas, não lineares, etc...)
- * o tipo de **variáveis** usadas (inteiras, binárias, discretas, contínuas, etc...)
- * o tipo de **restrições** consideradas (igualdade, desigualdade, infinitas, complementaridade, etc...)
- * o tipo de **solução** que se pretende obter (local ou global)
- * **diferenciabilidade** das funções envolvidas (optimização com ou sem derivadas)



Classificação de problemas

Os problemas de optimização podem ser classificados de acordo com:

- * as funções envolvidas na **função objectivo** e nas **restrições** (lineares, quadráticas, não lineares, etc...)
- * o tipo de **variáveis** usadas (inteiras, binárias, discretas, contínuas, etc...)
- * o tipo de **restrições** consideradas (igualdade, desigualdade, infinitas, complementaridade, etc...)
- * o tipo de **solução** que se pretende obter (local ou global)
- * **diferenciabilidade** das funções envolvidas (optimização com ou sem derivadas)



Classificação de problemas

Os problemas de optimização podem ser classificados de acordo com:

- ✳ as funções envolvidas na **função objectivo** e nas **restrições** (lineares, quadráticas, não lineares, etc...)
- ✳ o tipo de **variáveis** usadas (inteiras, binárias, discretas, contínuas, etc...)
- ✳ o tipo de **restrições** consideradas (igualdade, desigualdade, infinitas, complementaridade, etc...)
- ✳ o tipo de **solução** que se pretende obter (local ou global)
- ✳ **diferenciabilidade** das funções envolvidas (optimização com ou sem derivadas)



Classificação de problemas

Os problemas de optimização podem ser classificados de acordo com:

- ✱ as funções envolvidas na **função objectivo** e nas **restrições** (lineares, quadráticas, não lineares, etc...)
- ✱ o tipo de **variáveis** usadas (inteiras, binárias, discretas, contínuas, etc...)
- ✱ o tipo de **restrições** consideradas (igualdade, desigualdade, infinitas, complementaridade, etc...)
- ✱ o tipo de **solução** que se pretende obter (local ou global)
- ✱ **diferenciabilidade** das funções envolvidas (optimização com ou sem derivadas)



Classificação de problemas

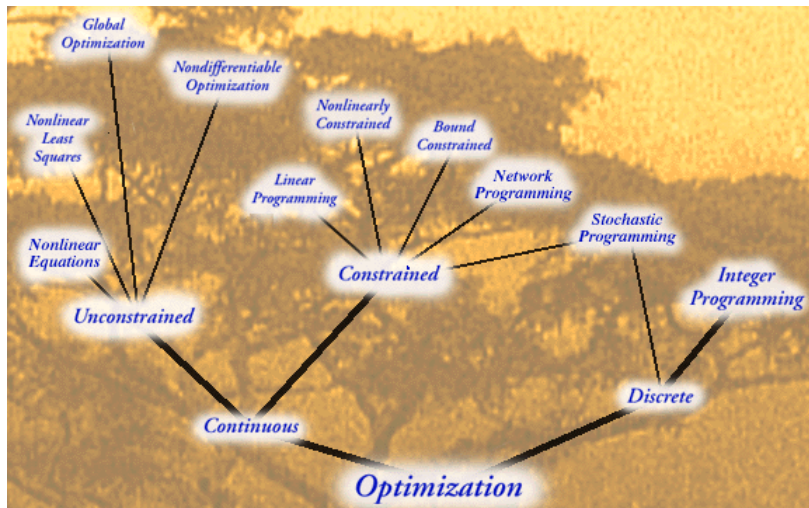
Os problemas de optimização podem ser classificados de acordo com:

- ✱ as funções envolvidas na **função objectivo** e nas **restrições** (lineares, quadráticas, não lineares, etc...)
- ✱ o tipo de **variáveis** usadas (inteiras, binárias, discretas, contínuas, etc...)
- ✱ o tipo de **restrições** consideradas (igualdade, desigualdade, infinitas, complementaridade, etc...)
- ✱ o tipo de **solução** que se pretende obter (local ou global)
- ✱ **diferenciabilidade** das funções envolvidas (optimização com ou sem derivadas)



Optimization tree

<http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/>



Importância da classificação

- ✳ **Não existe** software que resolva todos os tipos de problemas.
- ✳ O tipo de problema que se pretende resolver **condiciona** o software (*solver* e linguagem de modelação) a usar.



Importância da classificação

- ✳ **Não existe** software que resolva todos os tipos de problemas.
- ✳ O tipo de problema que se pretende resolver **condiciona** o software (*solver* e linguagem de modelação) a usar.



Conteúdo

- 1 Introdução
- 2 Classificação de problemas
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste
- 5 Um exemplo de controlo óptimo
- 6 Perfis de desempenho
- 7 Conclusões



AMPL

- * AMPL significa *A Mathematical Programming Language*.
- * O AMPL é um software comercial, mas existe uma versão *Student Edition* que é limitada (300 variáveis, 300 restrições).
- * O ambiente que compila a linguagem AMPL é constituído por um único executável (`ampl(.exe)`, dependendo do sistema operativo).
- * `www.ampl.com`



AMPL

- * AMPL significa *A Mathematical Programming Language*.
- * O AMPL é um software **comercial**, mas existe uma versão *Student Edition* que é limitada (300 variáveis, 300 restrições).
- * O ambiente que compila a linguagem AMPL é constituído por um único executável (`ampl(.exe)`, dependendo do sistema operativo).
- * `www.ampl.com`



AMPL

- * AMPL significa *A Mathematical Programming Language*.
- * O AMPL é um software **comercial**, mas existe uma versão *Student Edition* que é limitada (300 variáveis, 300 restrições).
- * O ambiente que compila a linguagem AMPL é constituído por um único executável (`ampl(.exe)`, dependendo do sistema operativo).
- * www.ampl.com

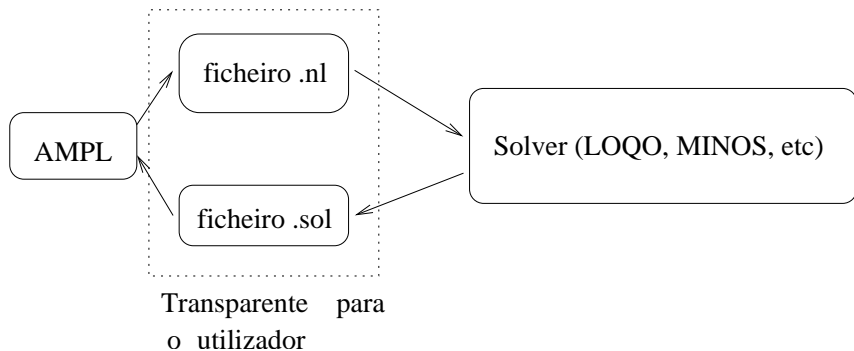


AMPL

- * AMPL significa *A Mathematical Programming Language*.
- * O AMPL é um software **comercial**, mas existe uma versão *Student Edition* que é limitada (300 variáveis, 300 restrições).
- * O ambiente que compila a linguagem AMPL é constituído por um único executável (`ampl(.exe)`, dependendo do sistema operativo).
- * `www.ampl.com`



Esquema de ligações



Alguns detalhes

- * O binário `amp1` é independente do *solver*.
- * É responsabilidade do *solver* obter o problema e apresentar a solução (existe um pacote de rotinas de domínio público que facilitam esta interface com o `amp1`).
- * A codificação do problema em AMPL é feita em **texto simples** (ascii). Por essa razão pode ser usado **qualquer** editor de texto (desde que o ficheiro seja gravado como texto simples).
- * Os ficheiros com o modelo do problema têm, usualmente, a extensão `.mod`.
- * Os dados do problema devem estar codificados em ficheiros separados para uma melhor organização. Estes ficheiros têm, usualmente a extensão `.dat`.



Alguns detalhes

- ✧ O binário `amp1` é independente do *solver*.
- ✧ É responsabilidade do *solver* obter o problema e apresentar a solução (existe um pacote de rotinas de domínio público que facilitam esta interface com o `amp1`).
- ✧ A codificação do problema em AMPL é feita em **texto simples** (ascii). Por essa razão pode ser usado **qualquer** editor de texto (desde que o ficheiro seja gravado como texto simples).
- ✧ Os ficheiros com o modelo do problema têm, usualmente, a extensão `.mod`.
- ✧ Os dados do problema devem estar codificados em ficheiros separados para uma melhor organização. Estes ficheiros têm, usualmente a extensão `.dat`.



Alguns detalhes

- ✳ O binário `amp1` é independente do *solver*.
- ✳ É responsabilidade do *solver* obter o problema e apresentar a solução (existe um pacote de rotinas de domínio público que facilitam esta interface com o `amp1`).
- ✳ A codificação do problema em AMPL é feita em **texto simples** (ascii). Por essa razão pode ser usado **qualquer** editor de texto (desde que o ficheiro seja gravado como texto simples).
- ✳ Os ficheiros com o modelo do problema têm, usualmente, a extensão `.mod`.
- ✳ Os dados do problema devem estar codificados em ficheiros separados para uma melhor organização. Estes ficheiros têm, usualmente a extensão `.dat`.



Alguns detalhes

- ✧ O binário `amp1` é independente do *solver*.
- ✧ É responsabilidade do *solver* obter o problema e apresentar a solução (existe um pacote de rotinas de domínio público que facilitam esta interface com o `amp1`).
- ✧ A codificação do problema em AMPL é feita em **texto simples** (ascii). Por essa razão pode ser usado **qualquer** editor de texto (desde que o ficheiro seja gravado como texto simples).
- ✧ Os ficheiros com o modelo do problema têm, usualmente, a extensão `.mod`.
- ✧ Os dados do problema devem estar codificados em ficheiros separados para uma melhor organização. Estes ficheiros têm, usualmente a extensão `.dat`.



Alguns detalhes

- ✧ O binário `ampl` é independente do *solver*.
- ✧ É responsabilidade do *solver* obter o problema e apresentar a solução (existe um pacote de rotinas de domínio público que facilitam esta interface com o `ampl`).
- ✧ A codificação do problema em AMPL é feita em **texto simples** (ascii). Por essa razão pode ser usado **qualquer** editor de texto (desde que o ficheiro seja gravado como texto simples).
- ✧ Os ficheiros com o modelo do problema têm, usualmente, a extensão `.mod`.
- ✧ Os dados do problema devem estar codificados em ficheiros separados para uma melhor organização. Estes ficheiros têm, usualmente a extensão `.dat`.



Exemplo em AMPL - hs014.mod

```
var x {1..2};

minimize obj:
  (x[1] - 2)^2 + (x[2]-1)^2;

subject to constr1:
  x[1]^2/4 + x[2]^2 <= 1;
subject to constr2:
  x[1] - 2*x[2] = -1;
```

```
let x[1] := 2;
let x[2] := 2;

option solver loqo;
option loqo_options 'verbose 2';
solve;
display x;
display obj;
display obj - 9 + 2.875*sqrt(7);
```



Comentários

- * Problema com **duas** variáveis, **uma** restrição de desigualdade e **uma** de igualdade.
- * Aproximação inicial $x = (2, 2)^T$.
- * O comando `solve` indica ao AMPL que se pretende **resolver** o problema.
- * O `ampl` escreve um ficheiro temporário de extensão `.nl` e executa o `solver` com os argumentos `stub -AMPL`. Se não se usar a opção `solver` o `ampl` executa o `solver` por defeito (minos).
- * O `ampl` suspende a sua execução até que o `solver` termine e produza um ficheiro de extensão `.sol`.



Comentários

- * Problema com **duas** variáveis, **uma** restrição de desigualdade e **uma** de igualdade.
- * Aproximação inicial $x = (2, 2)^T$.
- * O comando `solve` indica ao AMPL que se pretende **resolver** o problema.
- * O `ampl` escreve um ficheiro temporário de extensão `.nl` e executa o `solver` com os argumentos `stub -AMPL`. Se não se usar a opção `solver` o `ampl` executa o `solver` por defeito (minos).
- * O `ampl` suspende a sua execução até que o `solver` termine e produza um ficheiro de extensão `.sol`.



Comentários

- * Problema com **duas** variáveis, **uma** restrição de desigualdade e **uma** de igualdade.
- * Aproximação inicial $x = (2, 2)^T$.
- * O comando `solve` indica ao AMPL que se pretende **resolver** o problema.
- * O `ampl` escreve um ficheiro temporário de extensão `.nl` e executa o `solver` com os argumentos `stub -AMPL`. Se não se usar a opção `solver` o `ampl` executa o `solver` por defeito (menos).
- * O `ampl` suspende a sua execução até que o `solver` termine e produza um ficheiro de extensão `.sol`.



Comentários

- * Problema com **duas** variáveis, **uma** restrição de desigualdade e **uma** de igualdade.
- * Aproximação inicial $x = (2, 2)^T$.
- * O comando `solve` indica ao AMPL que se pretende **resolver** o problema.
- * O `ampl` escreve um ficheiro temporário de extensão `.nl` e executa o `solver` com os argumentos `stub -AMPL`. Se não se usar a opção `solver` o `ampl` executa o `solver` por defeito (menos).
- * O `ampl` suspende a sua execução até que o `solver` termine e produza um ficheiro de extensão `.sol`.



Comentários

- * Problema com **duas** variáveis, **uma** restrição de desigualdade e **uma** de igualdade.
- * Aproximação inicial $x = (2, 2)^T$.
- * O comando `solve` indica ao AMPL que se pretende **resolver** o problema.
- * O `ampl` escreve um ficheiro temporário de extensão `.nl` e executa o `solver` com os argumentos `stub -AMPL`. Se não se usar a opção `solver` o `ampl` executa o `solver` por defeito (minos).
- * O `ampl` suspende a sua execução até que o `solver` termine e produza um ficheiro de extensão `.sol`.



Exemplo em AMPL - diet1.mod

```
set NUTR ordered;
set FOOD ordered;
param cost {FOOD} >= 0;
param f_min {FOOD} >= 0, default 0;
param f_max {j in FOOD} >= f_min[j], default Infinity;
param n_min {NUTR} >= 0, default 0;
param n_max {i in NUTR} >= n_min[i], default Infinity;
param amt {NUTR,FOOD} >= 0;

var Buy {j in FOOD} integer >= f_min[j], <= f_max[j];

minimize Total_Cost:  sum {j in FOOD} cost[j] * Buy[j];

minimize Nutr_Amt {i in NUTR}: sum {j in FOOD} amt[i,j] * Buy[j];

subject to Diet {i in NUTR}:
    n_min[i] <= sum {j in FOOD} amt[i,j] * Buy[j] <= n_max[i];
```



Exemplo em AMPL - diet1.dat

```
param:  FOOD:                cost   f_min   f_max  :=
  "Quarter Pounder w/ Cheese"  1.84    .       .
  "McLean Deluxe w/ Cheese"    2.19    .       .
  "Big Mac"                    1.84    .       .
  "Filet-O-Fish"              1.44    .       .
  "McGrilled Chicken"         2.29    .       .
  "Fries, small"              .77     .       .
  "Sausage McMuffin"          1.29    .       .
  "1% Lowfat Milk"            .60     .       .
  "Orange Juice"              .72     .       . ;
```



Exemplo em AMPL - diet1.dat

```
param:  NUTR:  n_min  n_max :=  
        Cal      2000      .  
        Carbo    350      375  
        Protein  55        .  
        VitA     100        .  
        VitC     100        .  
        Calc     100        .  
        Iron     100        . ;
```



Exemplo em AMPL - diet1.dat

```
param amt (tr):
```

	Cal	Carbo	Protein	VitA	VitC	Calc	Iron :=
"Quarter Pounder w/ Cheese"	510	34	28	15	6	30	20
"McLean Deluxe w/ Cheese"	370	35	24	15	10	20	20
"Big Mac"	500	42	25	6	2	25	20
"Filet-O-Fish"	370	38	14	2	0	15	10
"McGrilled Chicken"	400	42	31	8	15	15	8
"Fries, small"	220	26	3	0	15	0	2
"Sausage McMuffin"	345	27	15	4	0	20	15
"1% Lowfat Milk"	110	12	9	10	4	30	0
"Orange Juice"	80	20	1	2	120	2	2 ;



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * Facilidade de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * Fase *presolve*.
- * Interface *Kestrel* (Ligação ao *NEOS server*).



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * **Facilidade** de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * Fase *presolve*.
- * Interface *Kestrel* (Ligação ao *NEOS server*).



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * **Facilidade** de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * Fase *presolve*.
- * Interface *Kestrel* (Ligação ao *NEOS server*).



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * **Facilidade** de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * *Fase resolve.*
- * *Interface Kestrel (Ligação ao NEOS server).*



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * **Facilidade** de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * Fase *presolve*.
- * Interface *Kestrel* (Ligação ao *NEOS server*).



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * **Facilidade** de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * *Fase *presolve*.*
- * *Interface *Kestrel* (Ligação ao *NEOS server*).*



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * **Facilidade** de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * Fase *presolve*.
- * Interface *Kestrel* (Ligação ao *NEOS server*).



Algumas características

- * **Diferenciação automática** (até segundas derivadas).
- * **Facilidade** de codificação dos problemas.
- * É uma linguagem de **modelação** e não de **programação**.
- * Possibilidade de funções externas através de uma DLL.
- * Possibilidade de ligações a bases de dados através da interface ODBC da Microsoft (Excel).
- * Interface gráfica AMPL plus (ILOG OPL Studio).
- * Fase *presolve*.
- * Interface *Kestrel* (Ligação ao *NEOS server*).



GAMS

- * GAMS significa *General Algebraic Modeling System*.
- * É um software comercial, mas existe uma versão de demonstração para *download* limitada no número de variáveis (300), número de restrições (300), etc... .
- * www.gams.com (e <http://www.gamsworld.org>)



GAMS

- * GAMS significa *General Algebraic Modeling System*.
- * É um software **comercial**, mas existe uma versão de demonstração para *download* limitada no número de variáveis (300), número de restrições (300), etc... .
- * www.gams.com (e <http://www.gamsworld.org>)



GAMS

- ✳ GAMS significa *General Algebraic Modeling System*.
- ✳ É um software **comercial**, mas existe uma versão de demonstração para *download* limitada no número de variáveis (300), número de restrições (300), etc... .
- ✳ `www.gams.com` (e `http://www.gamsworld.org`)



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Modelos suportados

- ✧ LP Linear Programming
- ✧ MIP Mixed-Integer Programming
- ✧ NLP Non-Linear Programming
- ✧ MCP Mixed Complementarity Problems
- ✧ MPEC Mathematical Programs with Equilibrium Constraints
- ✧ CNS Constrained Nonlinear Systems
- ✧ DNLP Non-Linear Programming with Discontinuous Derivatives
- ✧ MINLP Mixed-Integer Non-Linear Programming
- ✧ QCP Quadratically Constrained Programs
- ✧ MIQCP Mixed Integer Quadratically Constrained Programs



Um exemplo - HS62

```
variables obj; positive variables x1,x2,x3; equations objdef,eq1, eq1x;

objdef..  obj =e= -32.174*( 255.*log((x1+x2+x3+0.03)/(0.09*x1+x2+x3+0.03))
                        + 280.*log((x2+x3+0.03)/(0.07*x2+x3+0.03))
                        + 290.*log((x3+0.03)/(0.13*x3+0.03)));

eq1.. 20*sqr(x1+x2+x3-1) =e= 0;
eq1x..      x1+x2+x3-1 =e= 0;

model m / objdef, eq1 /
model mx / objdef, eq1x /;
* x1.l=1/3; x2.l=1/3; x3.l=1/3;

solve m us nlp min obj;
* solve mx us nlp min obj;

scalar global global solution / -0.262725145e5 /
diff optcr - relative distance from global;

diff = (global - obj.l)/global; display global, obj.l, diff;
```



Algumas propriedades

- * Quase todas as características do AMPL também são válidas para o GAMS (diferenciação automática, NEOS, Kestrel, etc).
- * Linguagem um pouco mais **complexa** (não necessariamente mais poderosa).
- * Menor facilidade em ligar um *solver*.
- * Aparentemente existe mais informação sobre o AMPL que sobre o GAMS (pelo menos na página da Internet).



Algumas propriedades

- * Quase todas as características do AMPL também são válidas para o GAMS (diferenciação automática, NEOS, Kestrel, etc).
- * Linguagem um pouco mais **complexa** (não necessariamente mais poderosa).
- * Menor facilidade em ligar um *solver*.
- * Aparentemente existe mais informação sobre o AMPL que sobre o GAMS (pelo menos na página da Internet).



Algumas propriedades

- * Quase todas as características do AMPL também são válidas para o GAMS (diferenciação automática, NEOS, Kestrel, etc).
- * Linguagem um pouco mais **complexa** (não necessariamente mais poderosa).
- * Menor facilidade em ligar um *solver*.
- * Aparentemente existe mais informação sobre o AMPL que sobre o GAMS (pelo menos na página da Internet).



Algumas propriedades

- * Quase todas as características do AMPL também são válidas para o GAMS (diferenciação automática, NEOS, Kestrel, etc).
- * Linguagem um pouco mais **complexa** (não necessariamente mais poderosa).
- * Menor facilidade em ligar um *solver*.
- * Aparentemente existe mais informação sobre o AMPL que sobre o GAMS (pelo menos na página da Internet).



CUTEr

- ✱ Constrained and Unconstrained Testing Environment (revisited).
- ✱ O CUTEr é uma actualização do CUTE.
- ✱ "CUTEr is a versatile testing environment for optimization and linear algebra solvers. The package contains a collection of test problems, along with Fortran 77, Fortran 90/95 and Matlab tools intended to help developers design, compare and improve new and existing solvers."
- ✱ Usa a linguagem **SIF** (*Standard Input Format*) para descrever os problemas teste.
- ✱ hsl.rl.ac.uk/cuter-www/



CUTEr

- ✧ Constrained and Unconstrained Testing Environment (revisited).
- ✧ O CUTEr é uma actualização do CUTE.
- ✧ "CUTEr is a versatile testing environment for optimization and linear algebra solvers. The package contains a collection of test problems, along with Fortran 77, Fortran 90/95 and Matlab tools intended to help developers design, compare and improve new and existing solvers."
- ✧ Usa a linguagem **SIF** (*Standard Input Format*) para descrever os problemas teste.
- ✧ hsl.rl.ac.uk/cuter-www/



CUTEr

- ✧ Constrained and Unconstrained Testing Environment (revisited).
- ✧ O CUTEr é uma actualização do CUTE.
- ✧ "CUTEr is a versatile testing environment for optimization and linear algebra solvers. The package contains a collection of test problems, along with Fortran 77, Fortran 90/95 and Matlab tools intended to help developers design, compare and improve new and existing solvers."
- ✧ Usa a linguagem **SIF** (*Standard Input Format*) para descrever os problemas teste.
- ✧ hsl.rl.ac.uk/cuter-www/



CUTEr

- ✱ Constrained and Unconstrained Testing Environment (revisited).
- ✱ O CUTEr é uma actualização do CUTE.
- ✱ "CUTEr is a versatile testing environment for optimization and linear algebra solvers. The package contains a collection of test problems, along with Fortran 77, Fortran 90/95 and Matlab tools intended to help developers design, compare and improve new and existing solvers."
- ✱ Usa a linguagem **SIF** (*Standard Input Format*) para descrever os problemas teste.
- ✱ hsl.rl.ac.uk/cuter-www/



CUTEr

- ✧ Constrained and Unconstrained Testing Environment (revisited).
- ✧ O CUTEr é uma actualização do CUTE.
- ✧ "CUTEr is a versatile testing environment for optimization and linear algebra solvers. The package contains a collection of test problems, along with Fortran 77, Fortran 90/95 and Matlab tools intended to help developers design, compare and improve new and existing solvers."
- ✧ Usa a linguagem **SIF** (*Standard Input Format*) para descrever os problemas teste.
- ✧ hsl.rl.ac.uk/cuter-www/



Funcionamento

- * Usa a ferramenta SIFdec para converter os problemas em SIF para Fortran.
- * o Solver é compilado junto com o problema.
- * Funcionamento distinto do AMPL e GAMS. Compilação para a resolução de cada problema.
- * "Ready-to-use interfaces to existing packages, such as MINOS, SNOPT, filterSQP, Knitro, and more, are provided."



Funcionamento

- ✳ Usa a ferramenta SIFdec para converter os problemas em SIF para Fortran.
- ✳ o Solver é compilado **junto** com o problema.
- ✳ Funcionamento distinto do AMPL e GAMS. Compilação para a resolução de **cada** problema.
- ✳ "Ready-to-use interfaces to existing packages, such as MINOS, SNOPT, filterSQP, Knitro, and more, are provided."



Funcionamento

- ✳ Usa a ferramenta SIFdec para converter os problemas em SIF para **Fortran**.
- ✳ o Solver é compilado **junto** com o problema.
- ✳ Funcionamento distinto do AMPL e GAMS. Compilação para a resolução de **cada** problema.
- ✳ "Ready-to-use interfaces to existing packages, such as MINOS, SNOPT, filterSQP, Knitro, and more, are provided."



Funcionamento

- * Usa a ferramenta SIFdec para converter os problemas em SIF para Fortran.
- * o Solver é compilado junto com o problema.
- * Funcionamento distinto do AMPL e GAMS. Compilação para a resolução de cada problema.
- * "Ready-to-use interfaces to existing packages, such as MINOS, SNOPT, filterSQP, Knitro, and more, are provided."



Um exemplo - HS14

```

NAME          HS14
VARIABLES
  X1
  X2
GROUPS
  N  OBJ
  G  CON1
  E  CON2      X1  1.0  X2  -2.0
CONSTANTS
  HS14      CON1      -1.0
  HS14      CON2      -1.0
BOUNDS
  FR HS14      'DEFAULT'
START POINT
  HS14      X1      2.0
  HS14      X2      2.0
ELEMENT TYPE
  EV SQ      V1

```

```

  EV SQ-2      V1
  EV SQ-1      V1
ELEMENT USES
  T  E1      SQ-2
  V  E1      V1
  T  E2      SQ-1
  V  E2      V1
  T  E3      SQ
  V  E3      V1
  T  E4      SQ
  V  E4      V1
GROUP USES
  E  OBJ      E1
  E  CON1     E3      -0.25
OBJECT BOUND
  LO HS14      0.0
*  Solution
*LO SOLTN      1.42322464
ENDATA

```

Um exemplo - HS14

```
*****
* SET UP THE FUNCTION *
* AND RANGE ROUTINES *
*****
```

```
ELEMENTS      HS14
```

```
TEMPORARIES
```

```
R  V1M2
```

```
R  V1M1
```

```
INDIVIDUALS
```

```
T  SQ-2
```

```
A  V1M2
```

```
V1 - 2.0
```

```
F          (V1M2)**2
G  V1          2.0*(V1M2)
H  V1          V1          2.0
T  SQ-1
A  V1M1          V1 - 1.0
F          (V1M1)**2
G  V1          2.0*(V1M1)
H  V1          V1          2.0
T  SQ
F          V1 * V1
G  V1          2.0 * V1
H  V1          V1          2.0
```

```
ENDATA
```



Características

- ✧ Linguagem **pouco natural**.
- ✧ Não aconselhável para descrição de modelos complexos, mas é frequentemente usado para testes numéricos de software.
- ✧ A ferramenta SIFDec possui **diferenciação automática**.
- ✧ Muitos dos problemas disponíveis para o AMPL e GAMS são oriundos da colecção de problemas do CUTE.
- ✧ O CUTEr é software de domínio público (não limitado). Problemas teste de grandes dimensões sem o recurso a software comercial.



Características

- ✧ Linguagem **pouco natural**.
- ✧ Não aconselhável para descrição de modelos complexos, mas é frequentemente usado para testes numéricos de software.
- ✧ A ferramenta SIFDec possui **diferenciação automática**.
- ✧ Muitos dos problemas disponíveis para o AMPL e GAMS são oriundos da colecção de problemas do CUTE.
- ✧ O CUTEr é software de domínio público (não limitado). Problemas teste de grandes dimensões sem o recurso a software comercial.



Características

- ✧ Linguagem **pouco natural**.
- ✧ Não aconselhável para descrição de modelos complexos, mas é frequentemente usado para testes numéricos de software.
- ✧ A ferramenta SIFDec possui **diferenciação automática**.
- ✧ Muitos dos problemas disponíveis para o AMPL e GAMS são oriundos da colecção de problemas do CUTE.
- ✧ O CUTEr é software de **domínio público** (não limitado). Problemas teste de grandes dimensões sem o recurso a software comercial.



Características

- ✧ Linguagem **pouco natural**.
- ✧ Não aconselhável para descrição de modelos complexos, mas é frequentemente usado para testes numéricos de software.
- ✧ A ferramenta SIFDec possui **diferenciação automática**.
- ✧ Muitos dos problemas disponíveis para o AMPL e GAMS são oriundos da colecção de problemas do CUTE.
- ✧ O CUTEr é software de **domínio público** (não limitado). Problemas teste de grandes dimensões sem o recurso a software comercial.



Características

- ✧ Linguagem **pouco natural**.
- ✧ Não aconselhável para descrição de modelos complexos, mas é frequentemente usado para testes numéricos de software.
- ✧ A ferramenta SIFDec possui **diferenciação automática**.
- ✧ Muitos dos problemas disponíveis para o AMPL e GAMS são oriundos da colecção de problemas do CUTE.
- ✧ O CUTEr é software de **domínio público** (não limitado). Problemas teste de grandes dimensões sem o recurso a software comercial.



Conteúdo

- 1 Introdução
- 2 Classificação de problemas
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste**
- 5 Um exemplo de controlo óptimo
- 6 Perfis de desempenho
- 7 Conclusões



Colecções de problemas teste

- * <http://www-unix.mcs.anl.gov/~more/cops/> – COPS – (Large scale hard problems)
- * <http://www-unix.mcs.anl.gov/~leyffer/MacMPEC/> – (MPEC) – Equilibrium constraints - Alguns obtidos de bi-nível e semi-infinita.
- * <http://www.norg.uminho.pt/aivaz> – SIPAMPL – SIP e Optimização global.
- * <http://www.netlib.org/ampl/models/> – NETLIB – Modelos diversos incluindo os do CUTE.
- * <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/> – Optimização não linear - CUTE e não CUTE - programação semidefinida.
- * <http://plato.asu.edu/sub/testcases.html> para uma lista enorme de colecções de problemas.
- * <http://plato.asu.edu/bench.html> ainda mais colecções e *Benchmarks for Optimization Software*.



Colecções de problemas teste

- * <http://www-unix.mcs.anl.gov/~more/cops/> – COPS – (Large scale hard problems)
- * <http://www-unix.mcs.anl.gov/~leyffer/MacMPEC/> – (MPEC) – Equilibrium constraints - Alguns obtidos de bi-nível e semi-infinita.
- * <http://www.norg.uminho.pt/aivaz> – SIPAMPL – SIP e Optimização global.
- * <http://www.netlib.org/ampl/models/> – NETLIB – Modelos diversos incluindo os do CUTE.
- * <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/> – Optimização não linear - CUTE e não CUTE - programação semidefinida.
- * <http://plato.asu.edu/sub/testcases.html> para uma lista enorme de colecções de problemas.
- * <http://plato.asu.edu/bench.html> ainda mais colecções e *Benchmarks for Optimization Software*.



Colecções de problemas teste

- * <http://www-unix.mcs.anl.gov/~more/cops/> – COPS – (Large scale hard problems)
- * <http://www-unix.mcs.anl.gov/~leyffer/MacMPEC/> – (MPEC) – Equilibrium constraints - Alguns obtidos de bi-nível e semi-infinita.
- * <http://www.norg.uminho.pt/aivaz> – SIPAMPL – SIP e Optimização global.
- * <http://www.netlib.org/ampl/models/> – NETLIB – Modelos diversos incluindo os do CUTE.
- * <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/> – Optimização não linear - CUTE e não CUTE - programação semidefinida.
- * <http://plato.asu.edu/sub/testcases.html> para uma lista enorme de colecções de problemas.
- * <http://plato.asu.edu/bench.html> ainda mais colecções e *Benchmarks for Optimization Software*.



Colecções de problemas teste

- * <http://www-unix.mcs.anl.gov/~more/cops/> – COPS – (Large scale hard problems)
- * <http://www-unix.mcs.anl.gov/~leyffer/MacMPEC/> – (MPEC) – Equilibrium constraints - Alguns obtidos de bi-nível e semi-infinita.
- * <http://www.norg.uminho.pt/aivaz> – SIPAMPL – SIP e Optimização global.
- * <http://www.netlib.org/ampl/models/> – NETLIB – Modelos diversos incluindo os do CUTE.
- * <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/> – Optimização não linear - CUTE e não CUTE - programação semidefinida.
- * <http://plato.asu.edu/sub/testcases.html> para uma lista enorme de colecções de problemas.
- * <http://plato.asu.edu/bench.html> ainda mais colecções e *Benchmarks for Optimization Software*.



Colecções de problemas teste

- * <http://www-unix.mcs.anl.gov/~more/cops/> – COPS – (Large scale hard problems)
- * <http://www-unix.mcs.anl.gov/~leyffer/MacMPEC/> – (MPEC) – Equilibrium constraints - Alguns obtidos de bi-nível e semi-infinita.
- * <http://www.norg.uminho.pt/aivaz> – SIPAMPL – SIP e Optimização global.
- * <http://www.netlib.org/ampl/models/> – NETLIB – Modelos diversos incluindo os do CUTE.
- * <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/> – Optimização não linear - CUTE e não CUTE - programação semidefinida.
- * <http://plato.asu.edu/sub/testcases.html> para uma lista enorme de colecções de problemas.
- * <http://plato.asu.edu/bench.html> ainda mais colecções e *Benchmarks for Optimization Software*.



Colecções de problemas teste

- * <http://www-unix.mcs.anl.gov/~more/cops/> – COPS – (Large scale hard problems)
- * <http://www-unix.mcs.anl.gov/~leyffer/MacMPEC/> – (MPEC) – Equilibrium constraints - Alguns obtidos de bi-nível e semi-infinita.
- * <http://www.norg.uminho.pt/aivaz> – SIPAMPL – SIP e Optimização global.
- * <http://www.netlib.org/ampl/models/> – NETLIB – Modelos diversos incluindo os do CUTE.
- * <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/> – Optimização não linear - CUTE e não CUTE - programação semidefinida.
- * <http://plato.asu.edu/sub/testcases.html> para uma lista enorme de colecções de problemas.
- * <http://plato.asu.edu/bench.html> ainda mais colecções e *Benchmarks for Optimization Software*.



Colecções de problemas teste

- * <http://www-unix.mcs.anl.gov/~more/cops/> – COPS – (Large scale hard problems)
- * <http://www-unix.mcs.anl.gov/~leyffer/MacMPEC/> – (MPEC) – Equilibrium constraints - Alguns obtidos de bi-nível e semi-infinita.
- * <http://www.norg.uminho.pt/aivaz> – SIPAMPL – SIP e Optimização global.
- * <http://www.netlib.org/ampl/models/> – NETLIB – Modelos diversos incluindo os do CUTE.
- * <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/> – Optimização não linear - CUTE e não CUTE - programação semidefinida.
- * <http://plato.asu.edu/sub/testcases.html> para uma lista enorme de colecções de problemas.
- * <http://plato.asu.edu/bench.html> ainda mais colecções e *Benchmarks for Optimization Software*.



Conteúdo

- 1 Introdução
- 2 Classificação de problemas
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste
- 5 Um exemplo de controlo óptimo**
- 6 Perfis de desempenho
- 7 Conclusões



O problema de controlo

- ✳ O problema de controlo óptimo é descrito por um conjunto de equações diferenciais $\dot{\chi} = h(\chi, u, t)$, $\chi(t^0) = \chi^0$, $t^0 \leq t \leq t^f$, onde χ representa as variáveis de estado e u as variáveis de controlo.
- ✳ O índice de desempenho J pode ser descrito da seguinte forma geral

$$J(t^f) = \varphi(\chi(t^f), t^f) + \int_{t^0}^{t^f} \phi(\chi, u, t) dt,$$

onde φ é o índice de desempenho das variáveis de estado no instante de tempo final t^f e ϕ é o desempenho integrado durante a operação.

- ✳ Podem ser impostas restrições adicionais que reflectem algumas limitações físicas do sistema.



O problema de controlo

- ✳ O problema de controlo óptimo é descrito por um conjunto de equações diferenciais $\dot{\chi} = h(\chi, u, t)$, $\chi(t^0) = \chi^0$, $t^0 \leq t \leq t^f$, onde χ representa as variáveis de estado e u as variáveis de controlo.
- ✳ O índice de **desempenho** J pode ser descrito da seguinte forma geral

$$J(t^f) = \varphi(\chi(t^f), t^f) + \int_{t^0}^{t^f} \phi(\chi, u, t) dt,$$

onde φ é o índice de desempenho das variáveis de estado no instante de tempo final t^f e ϕ é o desempenho integrado durante a operação.

- ✳ Podem ser impostas restrições adicionais que reflectem algumas limitações físicas do sistema.



O problema de controlo

- ✳ O problema de controlo óptimo é descrito por um conjunto de equações diferenciais $\dot{\chi} = h(\chi, u, t)$, $\chi(t^0) = \chi^0$, $t^0 \leq t \leq t^f$, onde χ representa as variáveis de estado e u as variáveis de controlo.
- ✳ O índice de **desempenho** J pode ser descrito da seguinte forma geral

$$J(t^f) = \varphi(\chi(t^f), t^f) + \int_{t^0}^{t^f} \phi(\chi, u, t) dt,$$

onde φ é o índice de desempenho das variáveis de estado no instante de tempo final t^f e ϕ é o desempenho integrado durante a operação.

- ✳ Podem ser impostas restrições adicionais que reflectem algumas limitações físicas do sistema.



O problema de controlo óptimo

O problema de maximização (P) pode ser descrito na seguinte forma geral

Problema (P)

$$\max J(t^f) \quad (1)$$

$$s.t. \quad \dot{\chi} = h(\chi, u, t) \quad (2)$$

$$\underline{\chi} \leq \chi(t) \leq \bar{\chi}, \quad (3)$$

$$\underline{u} \leq u(t) \leq \bar{u}, \quad (4)$$

$$\forall t \in [t^0, t^f] \quad (5)$$

Onde as restrições de estado (3) e de controlo (4) são entendidas como desigualdades componente a componente.



Um exemplo de controlo óptimo em AMPL

```
function ethanol;

param Cell_mass := 1;
param Substrate := 150;
param Product   := 0;
param Volume    := 10;
param n := 5;
param h{1..n} := 12.24;
var f{1..n+1};

maximize fx:
    ethanol(1, n, {i in 1..n} h[i], {i in 1..n+1} f[i], Cell_mass,
        Substrate, Product, Volume);

subject to hbounds {i in 1..n}:
    0.01<= h[i] <= 20;
subject to fbounds {i in 1..n+1}:
    0<= f[i] <= 12;
```



Conteúdo

- 1 Introdução
- 2 Classificação de problemas
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste
- 5 Um exemplo de controlo óptimo
- 6 Perfis de desempenho**
- 7 Conclusões



Problemas teste

Perfis usados para comparação do **PSwarm** com os diversos solver de otimização global sem uso de derivadas.

- ✱ 122 problemas retirados da literatura sobre otimização global.
- ✱ 12 problemas de grande dimensão (entre 100 e 300 variáveis). Os restantes são de pequena (< 10) e média dimensão (< 30).
- ✱ Problemas teste disponíveis em <http://www.norg.uminho.pt/aivaz> (em *software*).



Problemas teste

Perfis usados para comparação do **PSwarm** com os diversos solver de otimização global sem uso de derivadas.

- ✱ 122 problemas retirados da literatura sobre otimização global.
- ✱ 12 problemas de grande dimensão (entre 100 e 300 variáveis). Os restantes são de pequena (< 10) e média dimensão (< 30).
- ✱ Problemas teste disponíveis em <http://www.norg.uminho.pt/aivaz> (em *software*).



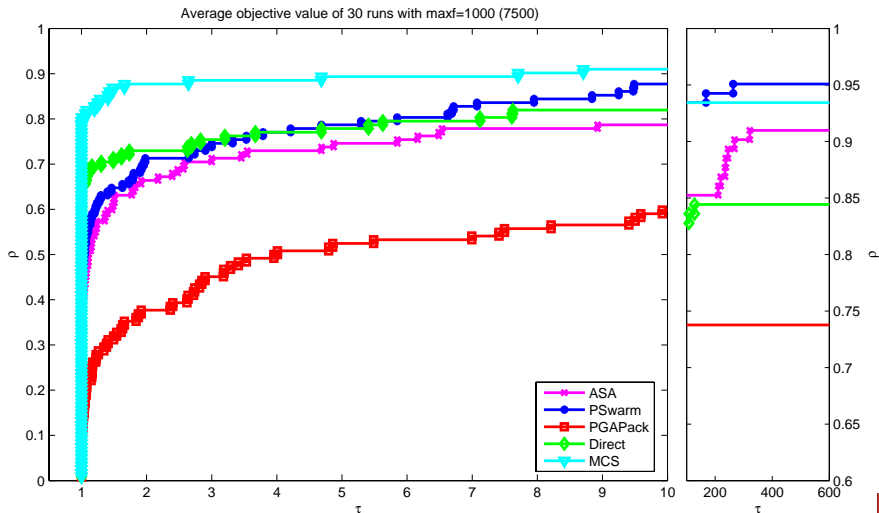
Problemas teste

Perfis usados para comparação do **PSwarm** com os diversos solver de otimização global sem uso de derivadas.

- ✱ 122 problemas retirados da literatura sobre otimização global.
- ✱ 12 problemas de grande dimensão (entre 100 e 300 variáveis). Os restantes são de pequena (< 10) e média dimensão (< 30).
- ✱ Problemas teste disponíveis em <http://www.norg.uminho.pt/aivaz> (em *software*).



Valor médio da função objectivo



Um exemplo em MATLAB

```

% Average objective function value of 30 runs
%ASA
T(:,1)=[1002      1002.066667 1002      1002.033333 1002      1002.066667
%PSwarm mu=nu=0.5, deltaf=5, popsize=20
T(:,2)=[1009.233333 239.9      249.2666667 251.5      254.5666667 512.8
%PGA
T(:,3)=[1009.7      1008.533333 1008.866667 1008.3      1008.166667 1009.3
%Direct
T(:,4)=[1245      1035      1079      1195      1021      1129      1003      1037
%Mcs
T(:,5)=[643 235 84 85 121 103 614 85 255 107 216 156 1000 1034
Titulo = 'Average objective evaluation of 30 runs with maxf=1000';

perf(T); % http://www-unix.mcs.anl.gov/~more/cops/

title(Titulo);
xlabel('\tau');
ylabel('\rho');
legend('ASA', 'PSwarm', 'PGAPack', 'Direct', 'MCS',4);

```



Conteúdo

- 1 Introdução
- 2 Classificação de problemas
- 3 Linguagens de modelação
 - AMPL
 - GAMS
 - CUTEr
- 4 Colecções de problemas teste
- 5 Um exemplo de controlo óptimo
- 6 Perfis de desempenho
- 7 Conclusões



Conclusões

- * AMPL e GAMS com uma linguagem **bastante acessível** e uma **grande variedade** de *solvers*.
- * O CUTEr é do domínio público, mas possui uma linguagem **menos natural**.
- * Todos possuem **ligação** ao MATLAB para que possa obter o problema a resolver a partir deste ambiente de trabalho.
- * **Perfis de desempenho** para comparação de diferentes *solvers*.



Conclusões

- * AMPL e GAMS com uma linguagem **bastante acessível** e uma **grande variedade** de *solvers*.
- * O CUTEr é do domínio público, mas possui uma linguagem **menos natural**.
- * Todos possuem **ligação** ao MATLAB para que possa obter o problema a resolver a partir deste ambiente de trabalho.
- * **Perfis de desempenho** para comparação de diferentes *solvers*.



Conclusões

- * AMPL e GAMS com uma linguagem **bastante acessível** e uma **grande variedade** de *solvers*.
- * O CUTER é do domínio público, mas possui uma linguagem **menos natural**.
- * Todos possuem **ligação** ao MATLAB para que possa obter o problema a resolver a partir deste ambiente de trabalho.
- * **Perfis de desempenho** para comparação de diferentes *solvers*.



Conclusões

- * AMPL e GAMS com uma linguagem **bastante acessível** e uma **grande variedade** de *solvers*.
- * O CUTER é do domínio público, mas possui uma linguagem **menos natural**.
- * Todos possuem **ligação** ao MATLAB para que possa obter o problema a resolver a partir deste ambiente de trabalho.
- * **Perfis de desempenho** para comparação de diferentes *solvers*.

