Johns Hopkins Engineering

Applied Machine Learning for Mechanical Engineers

Optimization, Part 2, A

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Introduction to Linear Optimization Programming Packages

- By the end of this lecture, you will be able to:
  - Address linear optimization problems such to be compatible with Python and MATLAB optimization packages
  - Describe “scipy.optimize.linprog” package in Python
  - Describe ‘linprog” package in MATLAB

Introduction to Linear Optimization Programming Packages

- Formal linear optimization problem with one objective function

$$\begin{gathered} minimize\ f(\boldsymbol{x}) \\ subject\ to \begin{cases} g_j(\boldsymbol{x}) \leq 0 & j \in \{1, 2, \dots, J\} \\ h_k(\boldsymbol{x}) = 0 & k \in \{1, 2, \dots, K\} \end{cases} \end{gathered} \qquad (2-1)$$

where $\boldsymbol{x} = [x_1, x_2, \dots, x_N]$ include the optimization variables (solution), $f(\boldsymbol{x})$, $g_j(\boldsymbol{x})$, and $h_k(\boldsymbol{x})$ are a linear objective function, linear inequality constraints, and linear equality constraints, respectively, and the rest is similar to Eq. 1-1.

Introduction to Linear Optimization Programming Packages

- Programming packages for linear programming

$$minimize\ \boldsymbol{C}.\boldsymbol{x}^T$$
$$subject\ to\ \begin{cases} \boldsymbol{G}.\boldsymbol{x} \leq \boldsymbol{A} \\ \boldsymbol{H}.\boldsymbol{x} = \boldsymbol{B} \\ \boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U} \end{cases} \qquad (2-2)$$

where $\boldsymbol{C}$ is the N-dimensional row vector of multipliers of N optimization variables in objective function, $\boldsymbol{G}$ is a J by N matrix of multipliers of N optimization variables in J inequality constraints, $\boldsymbol{A}$ is a J-dimensional column vector of equality constants, $\boldsymbol{H}$ is a K by N matrix of multipliers of N optimization variables in K inequality constraints, $\boldsymbol{B}$ is a K-dimensional column vector of equality constants, $\boldsymbol{L}$ is an N-dimensional row vector of lower bounds of N optimization variables, and $\boldsymbol{U}$ is an N-dimensional row vector of upper bounds of N optimization variables.

Introduction to Linear Optimization Programming Packages

- Programming packages for linear programming
  - Different annotations

$$
\begin{aligned}
\min_{x} \ & c^T x \\
\text{such that } & A_{ub} x \leq b_{ub}, \\
& A_{eq} x = b_{eq}, \\
& l \leq x \leq u,
\end{aligned}
$$

Figure 2-1 Annotations used to address linear optimization problems at
https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html

Introduction to Linear Optimization Programming Packages

- Programming packages for linear programming
  - Python: “scipy.optimize.linprog” at

    https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html

  - MATLAB: “linprog” at

    https://www.mathworks.com/help/optim/ug/linprog.html

Introduction to Linear Optimization Programming Packages

- Python: “scipy.optimize.linprog”

SciPy.org

SciPy.org | Docs | SciPy v1.4.1 Reference Guide | Optimization and Root Finding (scipy.optimize) | index | modules | next | previous

scipy.optimize.linprog¶

scipy.optimize.linprog(*c, A_ub=None, b_ub=None, A_eq=None, b_eq=None, bounds=None, method='interior-point', callback=None, options=None, x0=None*) [source]

Linear programming: minimize a linear objective function subject to linear equality and inequality constraints.

Linear programming solves problems of the following form:

$$\begin{aligned} \min_x \ & c^T x \\ \text{such that } & A_{ub} x \leq b_{ub}, \\ & A_{eq} x = b_{eq}, \\ & l \leq x \leq u, \end{aligned}$$

where x is a vector of decision variables; c, b_{ub}, b_{eq}, l, and u are vectors; and A_{ub} and A_{eq} are matrices.

Informally, that’s:

minimize:

Previous topic

root(method=’df-sane’)

Next topic

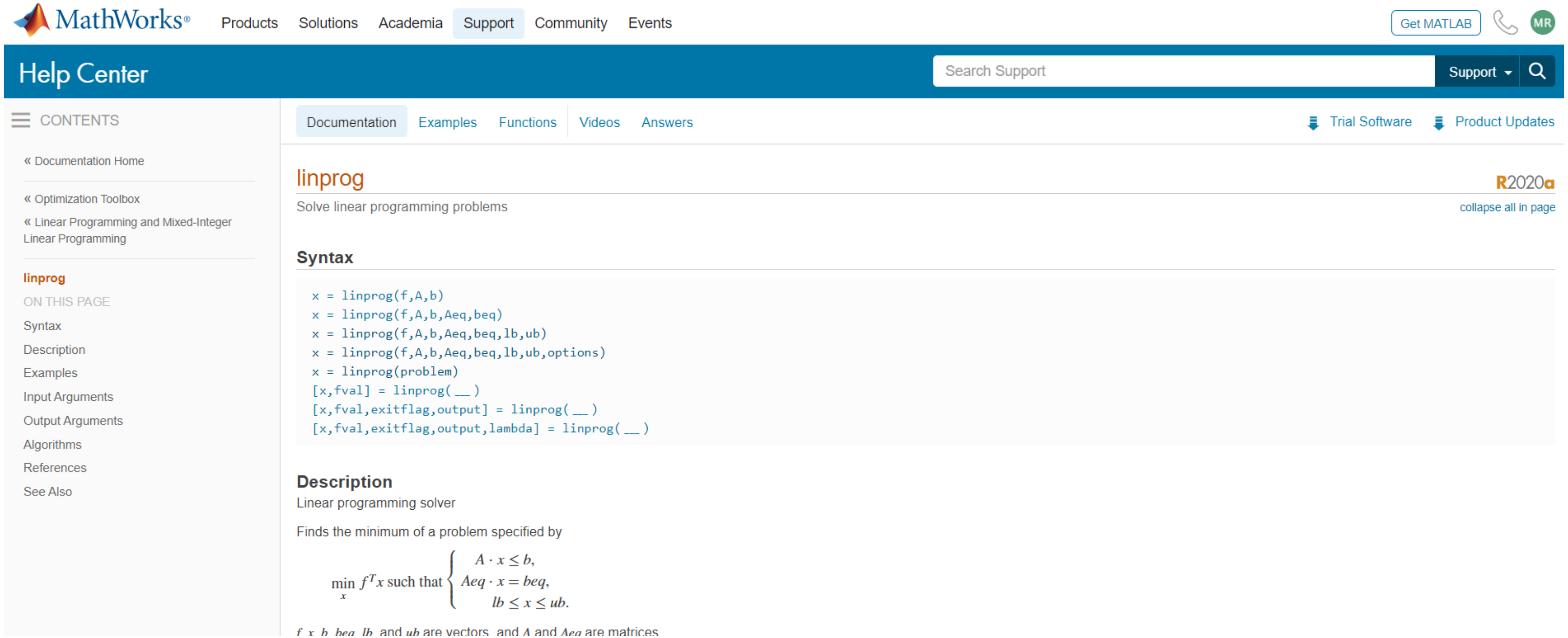
linprog(method=’simplex’)

Quick search

search

Introduction to Linear Optimization Programming Packages

- MATLAB: “linprog”


MathWorks® Products Solutions Academia Support Community Events Get MATLAB MR

Help Center Search Support Support

CONTENTS

« Documentation Home

« Optimization Toolbox

« Linear Programming and Mixed-Integer Linear Programming

linprog

ON THIS PAGE

Syntax

Description

Examples

Input Arguments

Output Arguments

Algorithms

References

See Also

Documentation Examples Functions Videos Answers Trial Software Product Updates

linprog R2020a

Solve linear programming problems

collapse all in page

Syntax

```
x = linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub)
x = linprog(f,A,b,Aeq,beq,lb,ub,options)
x = linprog(problem)
[x,fval] = linprog( __ )
[x,fval,exitflag,output] = linprog( __ )
[x,fval,exitflag,output,lambda] = linprog( __ )
```

Description

Linear programming solver

Finds the minimum of a problem specified by

$$\min_x f^T x \text{ such that } \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub. \end{cases}$$

f, x, b, beq, lb, and ub are vectors, and A and Aeq are matrices.

Introduction to Linear Optimization Programming Packages

- In this lecture, you learned about:

  - General formulations of linear optimization problems in Python and MATLAB optimization packages

  - “scipy.optimize.linprog” programming package in Python

  - ‘linprog” programming package in MATLAB

- In the next lecture, we will practice these programming packages in Python and MATLAB

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING