Solving Optimization Problems with MATLAB

Loren Shure
Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization
Optimization Problems

- Maximize Fuel Efficiency
- Minimize Risk
- Maximize Profits
Design Process

Initial Design Variables → System → Objectives met? → Optimal Design

Modify Design Variables

No → Yes
Why use Optimization?

Manually (trial-and-error or iteratively)
Why use Optimization?

Automatically (using optimization techniques)
Why use Optimization?

- Finding better (optimal) designs and decisions
- Faster design and decision evaluations
- Automate routine decisions
- Useful for trade-off analysis
- Non-intuitive designs may be found

Antenna Design Using Genetic Algorithm
http://ic.arc.nasa.gov/projects/esp/research/antenna.htm
Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization
Curve Fitting Demo

Given some data:

\[
t = [0 \ 0.3 \ 0.8 \ 1.1 \ 1.6 \ 2.3];
\]

\[
y = [0.82 \ 0.72 \ 0.63 \ 0.60 \ 0.55 \ 0.50];
\]

Fit a curve of the form:

\[
y(t) = c_1 + c_2 e^{-t}
\]
How to solve?

As a linear system of equations:

\[ y(t) = c_1 + c_2 e^{-t} \]

\[ y = \begin{bmatrix} 1 & e^{-t} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = Ec \]

\[ \begin{bmatrix} 0.82 \\ 0.72 \\ 0.63 \\ 0.60 \\ 0.55 \\ 0.50 \end{bmatrix} = \begin{bmatrix} 1 & e^0 \\ 1 & e^{-0.3} \\ 1 & e^{-0.8} \\ 1 & e^{-1.1} \\ 1 & e^{-1.6} \\ 1 & e^{-2.3} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \]

Can’t solve this exactly (6 eqns, 2 unknowns)

\[ y = Ec \]

\[ \min_c \| Ec - y \|_2^2 \]

An optimization problem!
Topics

- Introduction
- Least-squares minimization
- **Nonlinear optimization**
- Mixed-integer programming
- Global optimization
Nonlinear Optimization

\[ \min_{x} \log \left( 1 + \left( x_1 - \frac{4}{3} \right)^2 + 3 \left( x_1 + x_2 - x_1^3 \right)^2 \right) \]
Nonlinear Optimization - Modeling Gantry Crane

- Determine acceleration profile that minimizes payload swing

Constraints:
- \( t_f \geq t_{p1} + t_{p2} \)
- \( 1 \text{s} \leq t_{p1} \leq 20 \text{s} \)
- \( 1 \text{s} \leq t_{p2} \leq 20 \text{s} \)
- \( 4 \text{s} \leq t_f \leq 25 \text{s} \)
Symbolic Math Toolbox
Functions for analytical computations

- Conveniently manage & document symbolic computations in Live Editor
  - Math notation, embedded text, graphics
  - Share work as pdf or html

- Perform exact computations using familiar MATLAB syntax in MATLAB

- Integrate with numeric computing – MATLAB, Simulink and Simscape language

- Perform Variable-precision arithmetic

Integration
\[
\int \left( e^{ax^2} \right) \,dx
\]
\[
\frac{\sqrt{n} \, \text{erf}(\sqrt{n} \, x)}{2 \, \sqrt{\pi}}
\]

Transforms
\[
\text{diff}(x^2 + \sin(y+x),x,y)
\]
\[
\cos(x) - x \, y \, \sin(x \, y)
\]

Solving equations
\[
\left\{ \begin{array}{l}
\frac{\left( \sqrt{\pi} \, \text{erf}(\sqrt{n} \, x) \right)}{2 \, \sqrt{\pi}} - m \, e^{-\frac{\left( \sqrt{\pi} \, \text{erf}(\sqrt{n} \, x) \right)}{2 \, \sqrt{\pi}}}
\end{array} \right.
\]
\[
\sqrt{B^2 - 4 \, km}
\]

Simplification
\[
f := \frac{4 \, \sqrt{70} \, x \, e^{-x} \, (x^2 - 1) \, \sqrt{x^2 + 13 \, x^3 - 42 \, x^2 - 84)}{315 \, (2 \, x + \sin(2 \, x) + 2)}
\]
Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization
Mixed-Integer Programming

- Many things exist in discrete amounts:
  - Shares of stock
  - Number of cars a factory produces
  - Number of cows on a farm

- Often have binary decisions:
  - On/off
  - Buy/don’t buy

- Mixed-integer linear programming:
  - Solve optimization problem while enforcing that certain variables need to be integer
Continuous and integer variables

\[ x_1 \in [0, 100] \quad x_2 \in \{1, 2, 3, 4, 5\} \]

Linear objective and constraints

\[
\min_x \quad -x_1 - 2x_2
\]

such that

\[
\begin{align*}
x_1 + 4x_2 & \leq 20 \\
x_1 + x_2 & = 10
\end{align*}
\]
Traveling Salesman Problem

Problem
- How to find the shortest path through a series of points?

Solution
- Calculate distances between all combinations of points
- Solve an optimization problem where variables correspond to trips between two points
Topics

- Introduction
- Least-squares minimization
- Nonlinear optimization
- Mixed-integer programming
- Global optimization
Example Global Optimization Problems

Why does \texttt{fmincon} have a hard time finding the function minimum?

\begin{align*}
\text{Starting at } 0 & \quad \text{Starting at } 1 & \quad \text{Starting at } 3 \\
\text{Starting at } 6 & \quad \text{Starting at } 8 & \quad \text{Starting at } 10
\end{align*}
Example Global Optimization Problems

Why didn’t \texttt{fminunc} find the maximum efficiency?
Example Global Optimization Problems

Why didn’t nonlinear regression find a good fit?

c = b_1 e^{-b_1 t} + b_2 e^{-b_2 t} + b_3 e^{-b_3 t}

t

c

0 20 40 60 80 100 120
Global Optimization

**Goal:**
Want to find the **lowest/largest** value of the nonlinear function that has **many local minima/maxima**

**Problem:**
Traditional solvers often return one of the local minima (not the global)

**Solution:**
A solver that locates globally optimal solutions
Global Optimization Solvers Covered Today

- Multi Start
- Global Search
- Simulated Annealing
- Pattern Search
- Particle Swarm
- Genetic Algorithm
MultiStart Demo – Nonlinear Regression

**lsqcurvefit solution**

**MultiStart solution**
MULTISTART
What is MultiStart?

- Run a local solver from each set of start points
- Option to filter starting points based on feasibility
- Supports parallel computing
MultiStart Demo – Peaks Function

Final x = [0.2283 -1.6255]
GLOBAL SEARCH
What is GlobalSearch?

- Multistart heuristic algorithm
- Calls \texttt{fmincon} from multiple start points to try and find a global minimum
- Filters/removes non-promising start points
GlobalSearch Overview

Schematic Problem

Peaks function
Three minima
Green, $z = -0.065$
Red, $z = -3.05$
Blue, $z = -6.55$
GlobalSearch Overview – Stage 0

Run from specified x0
GlobalSearch Overview – Stage 1
GlobalSearch Overview – Stage 1

![Graph showing the GlobalSearch Overview for Stage 1 with various points and labels.](Diagram.png)
GlobalSearch Overview – Stage 1
GlobalSearch Overview – Stage 2
Global Search Overview – Stage 2
GlobalSearch Overview – Stage 2
GlobalSearch Overview – Stage 2
GlobalSearch Overview – Stage 2

Current penalty threshold value: 4
GlobalSearch Overview – Stage 2

Current penalty threshold value : 4
GlobalSearch Overview – Stage 2

Expand basin of attraction if minimum already found

Basins can overlap
GlobalSearch Demo – Peaks Function

Final x = [0.2283 -1.6255]
SIMULATED ANNEALING
What is Simulated Annealing?

- A probabilistic metaheuristic approach based upon the physical process of annealing in metallurgy.

- Controlled cooling of a metal allows atoms to realign from a random higher energy state to an ordered crystalline (globally) lower energy state.
Simulated Annealing Overview – Iteration 1

*Run from specified x₀*
Simulated Annealing Overview – Iteration 1

Possible New Points: Standard Normal N(0,1) * Temperature

Temperature = 1
Simulated Annealing Overview – Iteration 1

Temperature = 1

\[ P_{accept} = \frac{1}{1 + e^{(x_{new} - x_{old})/T}} = 0.11 \]
Simulated Annealing Overview – Iteration 1

Temperature = 1
Simulated Annealing Overview – Iteration 1

Temperature = 1
Simulated Annealing Overview – Iteration 2

Temperature = 1
Simulated Annealing Overview – Iteration 2

Temperature = 0.75
Simulated Annealing Overview – Iteration N-1

Temperature = 0.1
Simulated Annealing Overview – Iteration N

Reannealing
Simulated Annealing Overview – Iteration N

Reannealing

\[ p_{accept} = \frac{1}{1 + e^{-(x_1^2 + y_1^2)/T}} = 0.27 \]
Simulated Annealing Overview – Iteration N

Reannealing

Temperature = 1
Simulated Annealing Overview – Iteration N+1

Temperature = 0.75
Simulated Annealing Overview – Iteration N+1

Temperature = 0.75
Simulated Annealing Overview – Iteration …

Temperature = 0.75
Simulated Annealing – Peaks Function

Final x = [-1.2770 0.1984]
PATTERN SEARCH (DIRECT SEARCH)
What is Pattern Search?

- An approach that uses a pattern of search directions around the existing points
- Expands/contracts around the current point when a solution is not found
- Does not rely on gradients: works on smooth and nonsmooth problems
Pattern Search Overview – Iteration 1

*Run from specified x0*
Pattern Search Overview – Iteration 1

Apply pattern vector, poll new points for improvement

Mesh size = 1
Pattern vectors = [1,0], [0,1], [-1,0], [0,-1]

First poll successful

\[ P_{\text{new}} = \text{mesh size} \times \text{pattern vector} + x_0 \]

1.6 = 1 * [1,0] + x_0

Complete Poll (not default)
Pattern Search Overview – Iteration 2

Mesh size = 2
Pattern vectors = [1,0], [0,1], [-1,0], [0,-1]
Pattern Search Overview – Iteration 3

Mesh size = 4
Pattern vectors = [1,0], [0,1], [-1,0], [0,-1]

x

y

-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3
Pattern Search Overview – Iteration 4

Mesh size $= 4 \times 0.5 = 2$
Pattern vectors $= [1,0], [0,1], [-1,0], [0,-1]$

MathWorks
Pattern Search Overview – Iteration N

Continue expansion/contraction until convergence…
Pattern Search – Peaks Function

Final x = [0.2283 -1.6255]
Pattern Search Climbs Mount Washington
PARTICLE SWARM
What is Particle Swarm Optimization?

- A collection of particles move throughout the region
- Particles have velocity and are affected by the other particles in the swarm
- Does not rely on gradients: works on smooth and nonsmooth problems
Particle Swarm Overview – Iteration 1

Initialize particle locations and velocities, evaluate all locations
Particle Swarm Overview – Iteration N

*Update velocities for each particle*
Particle Swarm Overview – Iteration N

Update velocities for each particle
Particle Swarm Overview – Iteration N

Move particles based on new velocities
Particle Swarm Overview – Iteration N

Continue swarming until convergence
Particle Swarm – Peaks Function

Final $x = [0.2283 \ -1.6255]$
GENETIC ALGORITHM
What is a Genetic Algorithm?

- Uses concepts from *evolutionary biology*
- Start with an initial generation of candidate solutions that are tested against the objective function
- Subsequent generations evolve from the 1st through *selection, crossover* and *mutation*
How Evolution Works – Binary Case

- **Selection**
  - *Retain* the best performing bit strings from one generation to the next. *Favor these for reproduction*
    - parent1 = [1 0 1 0 0 1 1 0 0 0]
    - parent2 = [1 0 0 1 0 0 1 0 1 0]

- **Crossover**
  - parent1 = [1 0 1 0 0 1 1 0 0 0]
  - parent2 = [1 0 0 1 0 0 1 0 1 0]
  - child = [1 0 0 0 0 1 1 0 1 0]

- **Mutation**
  - parent = [1 0 1 0 0 1 1 0 0 0]
  - child = [0 1 0 1 0 1 0 0 0 1]
Genetic Algorithm – Iteration 1

Evaluate initial population
Genetic Algorithm – Iteration 1

Select a few good solutions for reproduction
Genetic Algorithm – Iteration 2

*Generate new population and evaluate*
Genetic Algorithm – Iteration 2
Genetic Algorithm – Iteration 3
Genetic Algorithm – Iteration 3
Genetic Algorithm – Iteration N

Continue process until stopping criteria are met

Solution found
Genetic Algorithm – Peaks Function

Final x = [0.2283 -1.6255]
Genetic Algorithm – Integer Constraints

Mixed Integer Optimization

$$\min_{x} f(x)$$

s.t. some \( x \) are integers

Examples

- Only certain sizes of components available
- Can only purchase whole shares of stock
Application: Circuit Component Selection

- 6 components to size
- Only certain sizes available
- Objective:
  - Match Voltage vs. Temperature curve

Thermistors: Resistance varies nonlinearly with temperature

\[ R_{TH} = \frac{R_{TH,Nom}}{e^{\beta (T - T_{Nom}) / (T_{Nom} + T) \beta}} \]
Global Optimization Toolbox Solvers

- **GlobalSearch, MultiStart**
  - Well suited for smooth objective and constraints
  - Return the location of local and global minima

- **ga, gamultiobj, simulannealbnd, particleswarm**
  - Many function evaluations to sample the search space
  - Work on both smooth and nonsmooth problems

- **patternsearch**
  - Fewer function evaluations than **ga, simulannealbnd, particleswarm**
  - Does not rely on gradient calculation like **GlobalSearch and MultiStart**
  - Works on both smooth and nonsmooth problems
Optimization Toolbox Solvers

- **fmincon, fminbnd, fminunc, fgoalattain, fminimax**
  - Nonlinear constraints and objectives
  - Gradient-based methods for smooth objectives and constraints

- **quadprog, linprog**
  - Linear constraints and quadratic or linear objective, respectively

- **intlinprog**
  - Linear constraints and objective and integer variables

- **lsqlin, lsqnonneg**
  - Constrained linear least squares

- **lsqnonlin, lsqcurvefit**
  - Nonlinear least squares

- **fsolve**
  - Nonlinear equations
Speeding-up with Parallel Computing

- Global Optimization solvers that support Parallel Computing:
  - ga, gamultiobj: Members of population evaluated in parallel at each iteration
  - patternsearch: Poll points evaluated in parallel at each iteration
  - particleswarm: Population evaluated in parallel at each iteration
  - MultiStart: Start points evaluated in parallel

- Optimization solvers that support Parallel Computing:
  - fmincon: parallel evaluation of objective function for finite differences
  - fminunc, fminimax, fgoalattain, fsolve, lsqcurvefit, lsqnonlin: same as fmincon

- Parallel Computing can also be used in the Objective Function
  - parfor
Parallel Computing Toolbox for the Desktop

- Speed up parallel applications
- Take advantage of GPUs
- Prototype code for your cluster
Scale Up to Clusters and Clouds
Learn More about Optimization with MATLAB

Recorded webinar: Mixed Integer Linear Programming in MATLAB

Recorded webinar: Optimization in MATLAB for Financial Applications

MATLAB Digest: Using Symbolic Gradients for Optimization

MATLAB Digest: Improving Optimization Performance with Parallel Computing

Optimization Toolbox Web demo: Finding an Optimal Path using MATLAB and Optimization Toolbox
Key Takeaways

- Solve a wide variety of optimization problems in MATLAB
  - Linear and Nonlinear
  - Continuous and mixed-integer
  - Smooth and Nonsmooth

- Find better solutions to multiple minima and non-smooth problems using global optimization

- Use symbolic math for setting up problems and automatically calculating gradients

- Using parallel computing to speed up optimization problems
Questions?