Nonlinear Optimization in AMPL with LGO

Presented by
János D. Pintér
Lecture Topics

- Introduction
- Continuous Nonlinear Optimization Model
- AMPL-LGO Solver Engine
- Illustrative Nonlinear Optimization Applications
- Conclusions and Illustrative References

- Comments and questions welcome! Further information available
Introducing Myself (Quickly)

• Education and professional background
  
  M.Sc., Eötvös Loránd University (ELTE, Budapest): applied mathematics / operations research
  
  Ph.D., Moscow State University: stochastic optimization
  
  D.Sc., Hungarian Academy of Sciences: global optimization

• Hungarian-Canadian; living and working in the USA since 2016

• Lifelong interest in modeling and solving decision problems inspired by the real world

• Teaching management science / operations research / industrial and system engineering courses

• Presented training courses, and worked as a scientist / consultant / lecturer in cca. 40 countries

• Optimization software developer, doing research and teaching with AMPL
Introduction

• Author of a research monograph, a university textbook, a tutorial book and an e-book; editor of 7 books (as of 2020); further work on two book projects in progress

• Author/co-author of 200+ articles, book chapters, technical reports,…

• Professional society officer (past or present): Hung. Acad. Sci., INFORMS, EUROPT

• Editorial board member (present): *J. of Global Optimization*, *Operations Research Forum*, *SpringerBriefs in Optimization* (and several other journals in the past)

• Professional summary information: 
  https://engineering.lehigh.edu/faculty/janos-d-pinter (link to change in September 2020)
  https://scholar.google.com/citations?user=iHrfmDEAAAAB

“May you live interesting times!” Indeed, almost never had a boring moment... ☺
Optimization Paradigm

• Optimization models help to find justifiable (reasonable) quantitative decisions in many real-world settings, when the “best” decision is often far from obvious...

• Decision variables model the key unknown factors of the problem studied

• Bound constraints express lower and upper variable bounds on the decision variables

• General (function) constraints express physical, logical, resource, financial, legal, etc. limitations that constrain the feasible (admissible) values of the decision variables

• Objective function expresses the chosen primary goal such as minimal total cost, minimal completion time, maximal profit, and so on

• Unless specifically noted otherwise, we will assume that
  
  There is only one objective function (else multi-criteria optimization needed)
  
  All model input data (parameters) are known with certainty (else stochastic optimization needed)
  
  Both assumptions are approximations in many real-world contexts, used to simplify the analysis
Optimization Model Development

General framework

Optimize system performance

\[
\begin{align*}
\text{min } & \quad f(x) \\
\text{s.t. } & \quad x = (x_1, \ldots, x_n) \\
\text{Subject to } & \quad g(x) \leq 0 \\
\text{Constraints (functions) } & \quad g_j(x) \leq 0, \ j = 1, \ldots, m \\
\text{Bound constraints } & \quad x_l \leq x \leq x_u
\end{align*}
\]

\[f: \mathbb{R}^n \rightarrow \mathbb{R}\]

\[g: \mathbb{R}^n \rightarrow \mathbb{R}^m\]

This concise model statement can be flexibly specified to describe decision-making scenarios, using corresponding model forms such as LP, ILP, MILP, NLO, MINLP, SP,…

A vast range of optimization applications in business, engineering, sciences
Continuous Nonlinear Optimization Model

- Optimization model: $\min f(x)$ s.t. $x \in D := \{ x_l \leq x \leq x_u, g(x) \leq 0 \}$
- We will assume that at least some of the functions $f$ and $g_j, j = 1, \ldots, m$ are nonlinear
- Key analytical assumptions
  - Bounds $x_l, x_u$ are finite and known
  - Feasible set $D$ is non-empty
  - All model functions $f$ and $g$ (component-wise) are continuous

- These analytical conditions are sufficient to guarantee the existence of the global solution set $X^*$
- Local optimization (LO): we assume to have access to a high quality initial solution guess which can be sufficiently improved locally, to arrive at a good or provably optimal local solution; global optimality is guaranteed e.g., by model convexity ($f$ and $g_j, j = 1, \ldots, m$ are all convex) when applicable
- Global optimization (GO): we assume that the model could have multiple optima of different quality, and we aim for the best of the local optima, the global solution $x^*$ (or global solution set $X^*$)
Global Optimization Challenges

• In this presentation, we focus on the GO class of continuous nonlinear optimization models

• Note in passing that all combinatorial optimization models formally belong to the GO class

• Finding the solution set $X^*$, or even finding one element $x^*$ of $X^*$ using purely analytical tools is often not possible: even the usage of local optimality conditions could become unmanageably cumbersome...

• Finding $X^*$ numerically also remains a tough challenge, in absence of provable model convexity

• To establish global optimality of a candidate solution $x^*$, theoretically “complete” information or numerically “exhaustive” global scope search is required

• On the next three slides, we illustrate the GO challenge by visual examples
An Illustrative Collection of *Box-constrained* Models

These are very small problems: $n = 2, m = 0$

However, some are already non-trivial GO challenges...

Numerical difficulties could rapidly increase as $n, m$ become larger...
Some Hard(er) GO Test Problems

What can be expected from GO software? Global scope search is key to numerical success...
GO Challenges Implied by Feasible Sets


Again, what can we expect from software when solving models which have difficult feasible sets? Traditional local optimization approaches could (and often will) fail in cases like 2 or 3; global search needed
Nonlinear Models: Modeling and Solver Difficulties

- No definite outcome, when solving new, “unusual”, or notoriously hard models
- It may be difficult to find even a reasonable initial solution: e.g., nonlinear equations
- It is often difficult to satisfy nonlinear (especially equality) constraints
- It could be difficult to maintain feasibility in non-convex models
- More complicated theory, many non-equivalent approaches (implemented in solver engines) to tackle hard nonlinear optimization problems
- Different starting solutions (“initial guesses”) can lead to different solutions reported by solver engines: this is typically due to non-convex model structure – and it often happens when we try to use local scope solvers to handle hard GO problems...
AMPL for Optimization Model Development

• AMPL is a sophisticated tool that supports the entire optimization modeling lifecycle: model development, testing, deployment, and maintenance

• By using a high-level model representation that describes optimization problems in the same ways that (knowledgeable) people think about them, AMPL promotes consistent and rapid model development with reliable results

• Comprehensive information: the AMPL book by Fourer, Gay, and Kernighan, *AMPL – A Modeling Language for Mathematical Programming*

• The AMPL book provides a detailed introductory tutorial, and discussions of all basic and advanced features; the book can be accessed at [www.ampl.com](http://www.ampl.com)

• For a more concise, but detailed technical description, consult the posted article of the AMPL authors (*Management Science*, 1990)
LGO Solver Suite for Global-Local Nonlinear Optimization

- LGO (Lipschitz Global Optimizer) can numerically handle the general class of continuous nonlinear optimization problems introduced above, with specific emphasis on solving GO models.
- Only direct model function value evaluations are required, without the need for higher order (derivative, Hessian, ...) information: this feature is particularly useful for “black box” systems optimization arising e.g., in interdisciplinary R&D.
- LGO uses a combination of global and local optimization algorithms: experimental design, branch-and-bound, randomized search, and constrained local optimization.
- LGO is one of the theoretically sound and numerically efficient GO solvers, as demonstrated by in-depth benchmarking studies, conducted by myself with colleagues and by others.
- LGO can be linked to model development systems and to integrated computing systems.
- Next, we highlight a set of GO examples solved numerically by using AMPL-LGO.
Global Optimization: Illustrative Applications

• Nonlinear transportation model (here we illustrate mod, dat, run, txt file system based development)
• Solving systems of nonlinear equations
• Model fitting to data (model calibration, inverse modeling)
• Data classification (clustering; only highlighted here for brevity)
• Potential energy models; example: electron positions on a prescribed surface
• Object packing and configuration design: largest small polygon, circles, ellipses, ovals, tetris-like items, with real-world engineering and scientific applications (only briefly discussed here)
• Optimization of “black box” systems (illustrated here by two non-AMPL implementation examples)
• Articles, books, web links available upon request
• All AMPL-LGO runtimes reported are based on runs performed on a several years old PC: LGO is mostly used in global search mode, leading to longer runs (and often returning better results*) compared to local search based solvers
• * YOU are encouraged to try and compare various nonlinear solvers linked to AMPL on GO models
Nonlinear Transportation Problem: Model File

Code based on a model discussed in the AMPL book by Fourer et al.

# A lumber company has wood supplies at given locations (set ORIG) that need to be transported to # markets (set DEST). Our goal is to optimize shipments among all possible supply-demand pairs, by # minimizing the total cost of shipments. This model extends the standard transportation problem, # by using an often more realistic nonlinear (concave) cost function. The model is nonconvex, and # it has an unknow number of local optima (see the illustrative results saved in the run file).

set ORIG;  # origins
set DEST;  # destinations

param supply {ORIG} >= 0;  # amount of wood supply available at origins
param demand {DEST} >= 0;  # amount of wood required at destinations

param cost1 {ORIG, DEST};  # fixed shipment costs per year [10^3 $ / 10^6 feet]
param cost2 {ORIG, DEST};  # fixed shipment costs per year [10^3 $ / 10^6 feet]

var Ship {i in ORIG, j in DEST} >= 0;  # units to ship

minimize Total_Cost: sum {i in ORIG, j in DEST} (cost1[i,j] + cost2[i,j] * Ship[i,j]) / (1 + Ship[i,j]) * Ship[i,j];

# Basic LP model objective: Cost[i,j] * Ship[i,j];
# Nonlinear Cost_Relation {i in ORIG, j in DEST}:
# Cost[i,j] = (cost1[i,j] + cost2[i,j] * Ship[i,j]) / (1 + Ship[i,j]);

subject to Supply {i in ORIG}: sum {j in DEST} Ship[i,j] <= supply[i];
subject to Demand {j in DEST}: sum {i in ORIG} Ship[i,j] >= demand[j];
Nonlinear Transportation Problem: Data File

# Nonlinear Transportation Problem
# Illustrative data file NLT.dat

data;

param: ORIG: supply :=
          01 1400  02  2600  03  2900 ;

param: DEST: demand :=
       D1 900   D2  1200  D3  600
       D4 400   D5  1700  D6 1100
       D7 1000 ;

param cost1 :
            D1  D2  D3  D4  D5  D6  D7 :=
       01  39  14  11  14  16  82  80
       02  27   9  12  39  26  65  17
       03  24  14  17  13  28  99  20 ;

param cost2 :
            D1  D2  D3  D4  D5  D6  D7 :=
       01  5   10  10  5  8  6  10
       02  3   8   7  6  2  9  3
       03  8   6   6  5  3  2  9 ;
# Nonlinear Transportation Problem
# Script file NLT.run

# Reset AMPL
reset;

# Load model
model NLT.mod;

# Load data
data NLT.dat;

# Nonconvex model: global scope search recommended
option solver lgo;

# Selecting local or global search mode in LGO
# Local search option
option lgo_options 'opmode = 0';
# Runtime = 0.015625 seconds Total_Cost = 37029.76714

# Global search options: opmode = 1 or 2 or 3
# Increased global search effort from default 5000 steps set automatically for this model size
option lgo_options 'opmode = 3 g_maxfct = 100000';
# Runtime = 0.578125 seconds Total_Cost = 26625.51004
Nonlinear Transportation Problem: Result File

Total_Cost = 26625.51004

:        _varname    _var    :=
1    "Ship['O1', 'D1']"    900
2    "Ship['O1', 'D2']"    100
3    "Ship['O1', 'D3']"    0
4    "Ship['O1', 'D4']"    400
5    "Ship['O1', 'D5']"    0
6    "Ship['O1', 'D6']"    0
7    "Ship['O1', 'D7']"    0
8    "Ship['O2', 'D1']"    0
9    "Ship['O2', 'D2']"    0
10   "Ship['O2', 'D3']"    0
11   "Ship['O2', 'D4']"    0
12   "Ship['O2', 'D5']"   1600
13   "Ship['O2', 'D6']"    0
14   "Ship['O2', 'D7']"   1000
15   "Ship['O3', 'D1']"    0
16   "Ship['O3', 'D2']"   1100
17   "Ship['O3', 'D3']"    600
18   "Ship['O3', 'D4']"    0
19   "Ship['O3', 'D5']"   100
20   "Ship['O3', 'D6']"   1100
21   "Ship['O3', 'D7']"    0;

:       _conname    _con.slack    :=
1    "Supply['O1']"   1.407443051e-10
2    "Supply['O2']"   1.264197635e-10
3    "Supply['O3']"   1.768967195e-10
4    "Demand['D1']"  -8.753886505e-11
5    "Demand['D2']"  -2.705746738e-11
6    "Demand['D3']"  -5.479705781e-11
7    "Demand['D4']"   -4.388311936e-11
8    "Demand['D5']"  -5.593392418e-11
9    "Demand['D6']"  -8.776623872e-11
10   "Demand['D7']"  -8.753886505e-11;
Systems of Nonlinear Equations: Example 1

- Core AMPL code of a small-scale example

```AMPL
# Decision variables
var x >= -1 <= 2;
var y >= -2 <= 1;

# Auxiliary objective: equation1^2 + equation2^2
# Since it could happen that there is no solution...
minimize error: (x - y + sin(2*x) - cos(y))^2 +
(4*x - exp(-y) + 5*sin(6*x-y) + 3*cos(3*y))^2;

# Constraints
# subject to
equation1: x - y + sin(2*x) - cos(y) = 0;
equation2: 4*x - exp(-y) + 5*sin(6*x-y) + 3*cos(3*y) = 0;
```

• Numerical solution found by AMPL-LGO (runtime very close to 0 sec):
  
  $x = 0.01475897554, \ y = -0.7124741692,$
  
  $\text{error} = 7.211482899e-27, \ \text{equation1} = -4.107825191e-15, \ \text{equation2} = -8.479328351e-14$

• There could be other solutions; it is possible to conduct systematic numerical search for these
Systems of Nonlinear Equations: Example 2

- Circuit Design (CD) Problem: A well-known GO test challenge

    # Following the classical study of Ebers and Moll (1954), a bipolar transistor can be modeled by an
    # electrical circuit. This model leads to a square system of highly nonlinear equations in 9 variables
    # that has been studied by numerous researchers over several decades.

    # Parameters: only a partial list is shown
    param one:=1.; param rthou:=0.001; param g11:=0.485; ... param cad8:=126.3450748162-2.6985*10^-6;

    # Decision variables and bound constraints
    var x1 >= 0 <= 10;
    var x2 >= 0 <= 10;
    var x3 >= 0 <= 10;
    var x4 >= 0 <= 10;
    var x5 >= 0 <= 10;
    var x6 >= 0 <= 10;
    var x7 >= 0 <= 10;
    var x8 >= 0 <= 10;
    var x9 >= 0 <= 10;

    # Objective function (here we tacitly assume that the system of equations has a solution[s]...)
    minimize dummy_objfct: 0;
# Constraints (all constraints are nonlinear)

subject to

con1 : \((1-x_1x_2)x_3(\exp(x_5(g_{11}-g_{31}x_7r_\text{hou}-g_{51}x_8r_\text{hou}))-1) - g_{51} + g_{41}x_2 + \text{cad}_1 = 0;\)

con2 : \((1-x_1x_2)x_3(\exp(x_5(g_{12}-g_{32}x_7r_\text{hou}-g_{52}x_8r_\text{hou}))-1) - g_{52} + g_{42}x_2 + \text{cad}_2 = 0;\)

con3 : \((1-x_1x_2)x_3(\exp(x_5(g_{13}-g_{33}x_7r_\text{hou}-g_{53}x_8r_\text{hou}))-1) - g_{53} + g_{43}x_2 + \text{cad}_3 = 0;\)

con4 : \((1-x_1x_2)x_3(\exp(x_5(g_{14}-g_{34}x_7r_\text{hou}-g_{54}x_8r_\text{hou}))-1) - g_{54} + g_{44}x_2 + \text{cad}_4 = 0;\)

con5 : \((1-x_1x_2)x_4(\exp(x_6(g_{11}-g_{21}-g_{31}x_7r_\text{hou}+g_{41}x_9r_\text{hou}))-1) - g_{51} + g_{41}x_2 + \text{cad}_5 = 0;\)

con6 : \((1-x_1x_2)x_4(\exp(x_6(g_{12}-g_{22}-g_{32}x_7r_\text{hou}+g_{42}x_9r_\text{hou}))-1) - g_{52} + g_{42}x_2 + \text{cad}_6 = 0;\)

con7 : \((1-x_1x_2)x_4(\exp(x_6(g_{13}-g_{23}-g_{33}x_7r_\text{hou}+g_{43}x_9r_\text{hou}))-1) - g_{53} + g_{43}x_2 + \text{cad}_7 = 0;\)

con8 : \((1-x_1x_2)x_4(\exp(x_6(g_{14}-g_{24}-g_{34}x_7r_\text{hou}+g_{44}x_9r_\text{hou}))-1) - g_{54} + g_{44}x_2 + \text{cad}_8 = 0;\)

con9 : \(x_1x_3x_2x_4 = 0;\)
Systems of Nonlinear Equations: Example 2 Solution

The solution found is identical to the published one with high numerical precision.

The verification of this solution using rigorous methods takes substantial time and computer resources (references available).
Globally Optimized Model Fitting to Data

An example developed in Mathematica: \(a + \sin[b \cdot (\pi t) + c] + \cos[d \cdot (3\pi t) + e] + \sin[f \cdot (5\pi t) + g] + \xi\)

Model Calibration in AMPL: An Example

# Problem statement: Given the input data vector x[K] and output data vector y[K]), find the optimized
# parameters of a postulated model type (structure), by minimizing the least squares error between
# model output and observations.

# To run this model, enter in the command window
# model MC.mod;

# reset AMPL
reset;

# Number of data points (time steps) in this example
param Tmax := 250;
set T = 1..Tmax;

# Equidistant input values
param x{T};
let {t in T} x[t] := t/Tmax;

# Preset optimal parameter values
param a0 := 1;
param b0 := 2;
param c0 := 3;
param d0 := 4;
param e0 := 5;
# Noise term standard deviation parameter used in normal distribution based model Normal(0, stdev);
param stdev = 0.1;
# To guarantee reproducible (identical) run results, set random seed parameter to a fixed value
option randseed 1;

# Calculated model output values
param y{T};

# Assign data set based on a postulated nonlinear model + noise term
let {t in T} y[t] := (x[t] - c0)^2 + (x[t] - d0)^2 + (x[t] - e0)^2 + sqrt(a0 + e0*x[t]) + 7*cos(3*b0*x[t] + c0) - 12*sin(2*a0*x[t]+c0) + log(1 + a0*x[t]) + Normal(0,stdev);

# Decision variables to optimize
var a >= 0 <= 6;
var b >= 0 <= 6;
var c >= 0 <= 6;
var d >= 0 <= 6;
var e >= 0 <= 6;

# Least squares error objective
minimize LSError: sum {t in T} (y[t] - ((x[t] - c)^2 + (x[t] - d)^2 +(x[t] - e)^2 + sqrt(a + e*x[t]) + 7*cos(3*b*x[t] + c) - 12*sin(2*a*x[t]+c) + log(1 + a*x[t])))^2;
Model Calibration in AMPL: An Example

```AMPL
option solver lgo;
option lgo_options 'g_maxfct = 100000 maxnosuc = 100000 l_maxfct = 100000';
solve;
display sqrt(LSError)/Tmax;
display _varname, _var;
display sqrt(LSError)/Tmax > MC.txt;
display _varname, _var >> MC.txt;
```

# Result obtained (retrieved from MC.txt file)
# Recalling the noise model, the solution (1, 2, 3, 4, 5) can be recovered only approximately

```
sqrt(LSError)/Tmax = 0.006486728897
```

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>d</td>
</tr>
<tr>
<td>5</td>
<td>e</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>_var</th>
<th>1.995145143</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.002629644</td>
<td></td>
</tr>
<tr>
<td>3.004339888</td>
<td></td>
</tr>
<tr>
<td>3.956755991</td>
<td></td>
</tr>
<tr>
<td>5.024104229</td>
<td></td>
</tr>
</tbody>
</table>
Optimized Calibration of Artificial Neural Networks

For concise ANN background, consult, e.g., https://en.wikipedia.org/wiki/Artificial_neural_network

Data Classification (Clustering) by Global Optimization

• Objective: find the most homogeneous grouping of a given data set (black dots in the lhs figure). This can be done numerically, by **globally optimizing the position of assigned cluster centers** (medoids) $c_k \, k = 1,...,K$: see the red dots in the rhs figure. For an arbitrary candidate medoid configuration, one can use the **nearest neighbor** rule to assign the data to the cluster centers $c_k$. DC quality is defined as the sum of all distances between the data and the associated $c_k$.

• Example: 400 three-dimensional points classified into 4 clusters; GO is key to success...

*Mathematica* + LGO used
Optimized Electron Configurations on the Surface of an Ellipsoid

• This example is largely based on the technical report *The Lightning AMPL Tutorial: A Guide for Nonlinear Optimization Users* by Dominique Orban (HEC-GERAD Montréal & École Polytechnique de Montréal, Canada)

• Consider a set of electrons that are physically constrained to be positioned on the surface of a conducting body, here an ellipsoid: we want to find their “natural configuration” (arrangement)

• Many similar problems arise in computational physics, chemistry, and biology models

• Such problems are frequently analyzed by postulating a potential energy function: then one can minimize this function to obtain the corresponding optimized object configuration

• Similar point or object arrangement problems often require global optimization; local search will typically fail – unless we are able to provide an insightful initial solution (based on expertise, *yet often still not guaranteed to lead to the unknown best configuration...*)
Electrons on an Ellipsoid: Model Development

• We assume that the ellipsoid center is at the origin 0∈R³; the ellipsoid is defined by its half-axes rx, ry and rz: these are model parameters

• This is a scalable model: the number N of electrons can be chosen as a model parameter

• Decision variables are the electron positions (x[i], y[i], z[i]) for i = 1..N

• Constraints express the fact that all electrons must be located on the ellipsoid surface:

\[(x[i]/r_x)^2 + (y[i]/r_y)^2 + (z[i]/r_z)^2 = 1\] Here x[i] corresponds to xᵢ, and so on: see function U(.) below

• Objective: The potential energy function used here is the \textit{Coulomb potential U(.)}, defined by the sum of reciprocal pairwise distances of all electrons as follows (ignoring a physical constant):

\[
U(x, y, z) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^{-1/2}
\]
AMPL Model File

model; # Opens up a model file: this command has to be in the model file or in the call
param N > 0, integer ; # Number of electrons: its value will be defined in the data file
param rx > 0; # Half axis of ellipse in direction x: value is given in the data file
param ry > 0; # Half axis of ellipse in direction y: value is given in the data file
param rz > 0; # Half axis of ellipse in direction z: value is given in the data file
var x {1..N} >= -rx <= rx; # x-coordinates of the electrons, with tight bounds
var y {1..N} >= -ry <= ry; # y-coordinates of the electrons, with tight bounds
var z {1..N} >= -rz <= rz; # z-coordinates of the electrons, with tight bounds
minimize CoulombPotential : # Objective function
  sum {i in 1..N-1} sum {j in i+1..N} 1/sqrt ( (x[i]-x[j])^2 + (y[i]-y[j])^2 + (z[i]-z[j])^2 );
subject to # All electrons are located on the surface of the ellipsoid
  EllipsoidConstraint {i in 1..N}: x[i]^2 /rx^2 + y[i]^2 /ry^2 + z[i]^2 /rz^2 = 1;
  # RangeConstraint {i in 1..N}: -0.5*rz <= z[i] <= 0.5*rz; # An example of optional tighter bounds
Optimized Object Packings: Further Examples

Optimized Circle Packing for $n=25$

Embedding circle contains circles with radii $r_k = k - 0.5$, $k=1,...,25$

- General problem: given a set of objects, find a given type of container with minimal size which contains all these objects in a non-overlapping arrangement
- In practice, further criteria (e.g., balance conditions) need to be also considered

Credits: Collaboration with Frank Kampas, Ignacio Castillo, Giorgio Fasano, Tatiana Romanova et al.; articles and book chapters available
Optimized Object Packings: Further Examples

Credits: Collaboration with Ignacio Castillo and Frank Kampas
“Black Box” Optimization

• **Motivation:** I have been working for a long time as an independent researcher / software developer / consultant / lecturer and trainer

• **Interdisciplinary research often leads to “black box” nonlinear optimization**

• The person(s) in charge of optimization don’t necessarily have access to the descriptive models with parameters to optimize... certain details and model properties may be unknown

Some application areas based on my own work experience (list continued on next two slides)

• advanced engineering design (acoustics, automotive, lasers, optics, and other areas)

• biotechnology

• chemical process analysis

• computational physics and chemistry
“Black Box” Optimization

Application areas (continued)
- defence
- environmental systems analysis and management
- financial modeling and optimization
- industrial product design
- model calibration
- oil and gas production
- process control
- radiation therapy planning
- risk analysis and management
“Black Box” Optimization

Application areas (continued)

• robot equipment design
• scientific modelling
• space engineering
  and many other areas...

• Credits: Collaboration with Roger Cooke, Mustafa Çağlayan, Larry Deschaine et al., Giorgio Fasano et al., Mark Gammon et al., Zoltán Horváth et al., Glenn Isenor, Thomas Mason et al., Maplesoft’s engineering application development team, Grigoris Pantoleontos et al., Christopher Purcell et al., Jouko Tervo et al., and a large number of other colleagues

• Related publications are available upon request;
• To illustrate, here two “black box” optimization applications are highlighted
“Black Box” Optimization: Intensity Modulated Radiotherapy

Descriptive model development by Jouko Tervo et al., U. of Kuopio

Objective: targeted dose to treated region, while – ideally – minimal radiation to surrounding tissue and organs at risk

See illustrative figure that shows globally optimized dose distribution; much better than earlier locally optimized dose distribution plans

Joint article and book chapter available
“Black Box” Optimization: Suspension System Design

• This case study was developed in collaboration with Maplesoft

• Goal: Given a required (target) behavior of a double wishbone suspension system – defined in terms of the displacement curves for bumps on the road – we want to find its optimal design point settings, known as “hard points”, that result in an optimized smooth ride

• “Black box” descriptive model: DynaFlex Pro software by MotionPro used to model this system; the resulting inverse problem is then solved by the Global Optimization Toolbox (LGO linked to Maple)

• Citing the application director of Maplesoft at the time of this study, global optimization helped to find a “stunning design”…

Concluding Notes

• Facing the vast universe of nonlinear systems and processes, we have an unlimited source of important, motivating (and often hard) modeling and optimization problems.

• Frequently, interdisciplinary research is required.

• The future of creative thinking – integrating concepts and tools from business analytics, operations research, systems engineering, and computational optimization – is bright!

• Continuing expansion of examples collected to teach optimization: new models welcome!

• Interested in collaborative R&D, training and consulting opportunities.

• Articles and further information are available upon request.

• Current (Lehigh University) personal webpage, with a professional summary and links to books https://engineering.lehigh.edu/faculty/janos-d-pinter

• Google Scholar http://scholar.google.ca/citations?user=iHrfmDEAAAJ&hl=en&oi=ao
Illustrative References

JDP: Global Optimization, LGO Software Versions, Benchmarking


• Benchmarking nonlinear optimization software in technical computing environments. *TOP* 21 (2013), 33-162. (Joint work with F.J. Kampas)


Illustrative References

Edited Books on NLP and GO Applications

Illustrative References

Some Challenging Global Optimization Applications


Illustrative References

Further Challenging Global Optimization Applications


Thanks for your attention!

Questions and comments welcome
janos.d.pinter@gmail.com