Optimization Techniques in CST STUDIO SUITE

Vratislav Sokol, CST
Optimization algorithms in CST STUDIO SUITE™

Classic Powell
Interpolated Quasi-Newton
Trust Region Framework
Nelder-Mead Simplex
Genetic Algorithm
Particle Swarm Optimization

General suggestions for optimizer setting

Examples
Waveguide corner
Dual-band matching circuit network
Planar filter tuning
Antenna array side lobes suppression
Optimizer Window Overview

Solver Selection
- Simulation type: Transient Solver

Optimizer Choice
- Algorithm: Trust Region Framework
- Reset min/max: 10% of initial value
- Domain accuracy: 1e-4

Automatically choose parameter boundaries

Parameter space definition

Termination criterion
Goal Function Overview (1)

An arbitrary number of goals can be defined. The optimizer will try to satisfy all goals.

You can choose to optimize the sum of all goals or the maximum of all goals.

Goals are 0D, and can be derived from any 1D or 0D Result Template
Goal Function Overview (2)

A range of the 1D result can be defined for goal value calculation.

The weight allows you to give priority to a goal over others.

Possible operators are: >, <, =, move min/max.

minimize and maximize operators exist for S-parameter results.
Optimizer - Goal Visualisation
Optimizer - Result Plots

also in DS
Local vs. Global Optimizers

- Classic Powell
- Interpolated Quasi Newton
- Trust Region Framework
- Nelder-Mead Simplex Algorithm
- Particle Swarm Optimization
- Genetic Algorithm

Local

Initial parameters already give a good estimate of the optimum, parameter ranges are small

Global

Initial parameters give a poor estimate of the optimum, parameter ranges are large
Example 1: Waveguide Corner

**Goal** - Minimize $S_{11}$

- Classic Powell
- Quasi Newton
- Simplex (Nelder-Mead)
- Genetic Algorithm
- Particle Swarm
- Trust Region Framework
A local optimizer that robustly finds an optimum within the given parameter bounds. Sometimes, many iterations are necessary when closing in on the optimum. This algorithm is suitable for one-variable problems.

Optimization terminates if two consecutive goal values $g_1$ and $g_2$ yield

$$\frac{2(g_1 - g_2)}{\|g_1\| + \|g_2\|} < \text{Accuracy}$$
Interpolated Quasi Newton

The optimizer allows a restart of the algorithm within an automatically chosen smaller parameter range. This range is determined by the previous pass.

A Search algorithm for expensive problems: The parameter space is sampled in each variable direction. EM simulations are only performed for these discrete parameter space points. A model is created from these evaluations and used for optimization. During the search, the model is updated regularly by real evaluations.
Trust Region Framework

If a normalized variation of the parameters becomes smaller than this value, the optimization terminates.

A fast and accurate optimizer that converges robustly and finds an optimum within the given parameter bounds using a low number of evaluations. It is suitable for 3D EM optimization.
Trust Region Framework Algorithm (1)

• Choose initial point \( x_0 \)

• Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:
• Go to the minimizer (predicted optimum) of the model inside the trust-region

• Verify: Does the error decrease?
  • If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
  • If 'just' true, keep trust region radius and calculate new model
  • If not true, reduce size of trust region.
Trust Region Framework Algorithm (2)

- Choose initial point $x_0$
- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

**Repeat:**
- Go to the minimizer (predicted optimum) of the model inside the trust-region

**Verify:** Does the error decrease?
- If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
- If 'just' true, keep trust region radius and calculate new model
- If not true, reduce size of trust region.
Trust Region Framework Algorithm (3)

- Choose initial point $x_0$

- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:
- Go to the minimizer (predicted optimum) of the model inside the trust-region

- Verify: Does the error decrease?
  - If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
  - If 'just' true, keep trust region radius and calculate new model
  - If not true, reduce size of trust region.
Trust Region Framework Algorithm (4)

- Choose initial point $x_0$

- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:
- Go to the minimizer (predicted optimum) of the model inside the trust-region

- Verify: Does the error decrease?
  - If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
  - If 'just' true, keep trust region radius and calculate new model
  - If not true, reduce size of trust region.
Trust Region Framework Algorithm (5)

- Choose initial point $x_0$

- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:
- Go to the minimizer (predicted optimum) of the model inside the trust-region

Verify: Does the error decrease?
- If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
- If 'just' true, keep trust region radius and calculate new model
- If not true, reduce size of trust region.
Trust Region Framework Algorithm (6)

... “The algorithm will be converged once the trust region radius or distance to the next predicted optimum becomes smaller than the specified domain accuracy.”
Global Optimizer Overview

**Nelder Mead**
An optimizer for more complex problem domains with good convergence behavior:
Uses relatively few evaluations if the problem has a low number of parameters (i.e., less than 5).

**Particle Swarm**
A global optimizer that uses a higher number of evaluations to explore the search space, also suited for larger numbers of parameters (hint: use distributed computing).

**Genetic Algorithm**
A global optimizer that uses a high number of evaluations to explore the search space, suited for large numbers of parameters or very complex problem domains (hint: use distributed computing).
General Suggestions

1. Try to use a concise parameterization.
2. Try to keep the number of goal functions low.
3. Monitor parameter changes throughout optimization to gain insight into convergence behavior.
4. Sometimes, re-formulating your goal function makes the difference (e.g., min vs. move min).
5. You can use coarse parameter sweeps to determine good initial values and boundaries, and to support the right choice of optimization algorithm.
6. If possible, use face constraints together with sensitivities in combination with the trust region optimizer.
Mobile Phone Antenna

Goal: Best impedance matching in bands 890-960 MHz and 1710-1880 MHz.
Optimisation in CST DS (1)

![CST DS Diagram](Image)
Optimisation in CST DS (2)

Algorithm: Nelder Mead Simplex Algorithm
Number of evaluations: 349
(solver: 349, interpolation: 0)
Initial goal function value = 29.6303278834
Best goal function value = 0
Last goal function value = 0

Best parameters so far:
C1 = 1.8976
C2 = 1.87995
C3 = 1.3601
L1 = 7.73246

Optimizer Goals Real Part

Sum of all goals
0_1D Result_magdB(S1,1)
1_1D Result_magdB(S1,1)
Trust Region Framework + Sensitivity
TRF + Sensitivity: Results

Initial
Best so far
Post-Processing Optimisation

- Radiation pattern of 8x1 antenna array is constructed from the farfield of one element by applying so called array factor using template based post-processing (TBPP).

- In the second step the side lobe level is minimized using pure TBPP optimisation.
Optimisation of TBPP Steps

Optimise amplitudes and phases of element excitations as a post-processing step to minimise vertical plane side-lobe levels.

vertical plane directivity

original $a_n=1 \Rightarrow \text{SLL} = -13.7 \text{ dB}$
Optimisation of Side Lobe Levels

Optimise amplitudes and phases of element excitations as a post-processing step to minimise vertical plane side-lobe levels.

vertical plane directivity

original $a_n=1 \Rightarrow SLL = -13.7$ dB
optimised $a_n \Rightarrow SLL = -20$ dB
## Optimisation of Side Lobe Levels

<table>
<thead>
<tr>
<th>Optimisation Comparison</th>
<th>Vertical Plane Directivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time per 3D simulation</td>
<td>5 min × 30 sec</td>
</tr>
<tr>
<td>Number of 3D simulations</td>
<td>40 × 8</td>
</tr>
<tr>
<td>Time per TBPP eval.</td>
<td>30 sec × 30 sec</td>
</tr>
<tr>
<td>Optimisation steps</td>
<td>40 × 40</td>
</tr>
<tr>
<td>Total simulation time</td>
<td>200 min + 60 min</td>
</tr>
</tbody>
</table>

Original $a_n = 1 \Rightarrow SLL = -13.7$ dB

Optimised $a_n \Rightarrow SLL = -20$ dB
1. CST STUDIO SUITE 2011 offers a complete portfolio of optimization methods for various application.

2. A new “Trust Region Framework” algorithm is very efficient tool for a direct 3D EM optimization especially in conjunction with the sensitivity analysis.

3. New visualization of goals and parameter values

4. Post-processing optimization without a need of any EM or circuit solver

5. A new “Minimax” goal function definition is now available.