

A Brief Introduction to Numerical Methods for Constrained Optimization

CEE 629. System Identification

Duke University, Spring 2019

1 Constrained Optimization

In a constrained optimization problem we are asked to find values for n *optimization parameters* or *design variables*, $\mathbf{x} = [x_1 \ x_2 \ x_3 \ \cdots \ x_n]^T$, that minimizes a specified objective function $J = f(\mathbf{x})$ such that a set of inequalities are satisfied. The suitability of the optimized solution often depends on more than the value of the objective function. Many problems must also meet a range of other criteria. These criteria are expressed as inequality constraints, and depend upon the set of optimization parameters. As long as the inequalities are met the criteria are satisfied. The inequalities constrain the best value of the objective. Note that any inequality $a \leq b$ can be written $a - b \leq 0$ or $a/b - 1 \leq 0$. In general, we will write a set of m inequality constraints as $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$, or $g_1(\mathbf{x}) \leq 0, g_2(\mathbf{x}) \leq 0, \dots, g_m(\mathbf{x}) \leq 0$.

With this notation, a constrained optimization problem may be written:

$$\begin{array}{ll} \underset{x_1, x_2, \dots, x_n}{\text{minimize}} & J = f(x_1, x_2, \dots, x_n), \quad \text{such that} \\ & \begin{array}{l} g_1(x_1, x_2, \dots, x_n) \leq 0 \\ g_2(x_1, x_2, \dots, x_n) \leq 0 \\ \vdots \\ g_m(x_1, x_2, \dots, x_n) \leq 0 \end{array} \end{array} \quad (1)$$

Constraining parameter values to lie within a range of reasonable values,

$$\begin{array}{l} x_{1,\min} \leq x_1 \leq x_{1,\max} \\ x_{2,\min} \leq x_2 \leq x_{2,\max} \\ \vdots \\ x_{n,\min} \leq x_n \leq x_{n,\max} \end{array} \quad (2)$$

prevents evaluations of the objective function and the constraint functions with parameter values that have no physical significance. Parameter value constraints may be included into the set of general inequality constraints.

$$\begin{array}{l} 1 - x_i/x_{i,\min} \leq 0 \\ x_i/x_{i,\max} - 1 \leq 0 \end{array} \quad (3)$$

In certain rare cases, we can write actual equations for $J = f(\mathbf{x})$ and $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$, and use calculus to find an equation for the constrained optimum. In the vast majority of practical problems, however, the equations are simply much too complicated for this approach. In such cases computer-aided *analysis* can automate the evaluation of the objective f and constraints \mathbf{g} of a particular trial solution \mathbf{x} . Further, computer-aided *optimization* allows one to rapidly determine feasible and possibly “optimal” solutions.

2 Overview of Numerical Methods for Constrained Optimization

Within an iteration of a constrained optimization algorithm, the vector of optimization parameters \mathbf{x} is updated to $\mathbf{x} + \mathbf{h}$, where \mathbf{h} is a change in parameters that reduces $f(\mathbf{x})$, if all $g_j(\mathbf{x}) \leq 0$, or reduces positive values of the constraints-in-violation, the set of positive-valued constraints $g_j(\mathbf{x})$. This typically involves steps such as:

1. initialize the optimization parameters $\mathbf{x}^{(k)}$, ($k = 0$) and set $n_{\text{feval}}=0$
2. evaluate the objective function $f(\mathbf{x}^{(k)})$ and the constraints $\mathbf{g}(\mathbf{x}^{(k)})$; $n_{\text{feval}} := n_{\text{feval}} + 1$.
3. check convergence criteria. if converged then stop, otherwise,
4. establish the (unit) direction vector of the parameter update \mathbf{h} , ($|\mathbf{h}| = 1$), that is, the *search direction*
5. establish a distance α along the search direction for which $f(\mathbf{x}^{(k)} + \alpha\mathbf{h})$ is sufficiently less than $f(\mathbf{x}^{(k)})$.
6. update the optimization parameters, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha\mathbf{h}^{(k)}$, set $k = k + 1$ and return to step 2.

There are two general classes of numerical optimization methods:

- **Search methods**

- are useful when a good (admissible) initial guess is hard to know
- are useful for problems in which $f(\mathbf{x})$ is not smooth (gradients are noisy)
- are useful when the feasible space is discontinuous
- do not require an evaluation of gradient vectors $[\partial f(\mathbf{x})/\partial \mathbf{x}]$
- can be slow to converge, many evaluations of $f(\mathbf{x})$
- require penalty methods to enforce constraints
- examples of search methods:
 - * Optimized Step Size Random Search (OSSRS)
 - * Symmetric Perturbation Stochastic Approximation (SPSA)
 - * Genetic Algorithms
 - * Nelder Mead Algorithm (NMA)

- **Gradient methods**

- converge to a minimum that is near the initial guess $\mathbf{x}^{(0)}$
- work best for smooth expressions $f(\mathbf{x})$ and smooth constraints $\mathbf{g}(\mathbf{x})$.
- require gradient vectors $[\partial f(\mathbf{x})/\partial \mathbf{x}]$, which may be evaluated using finite differences if an analytical gradient is not provided

- require the Hessian matrix $[\partial^2 f(\mathbf{x})/\partial \mathbf{x}^2]$ which is usually approximated from the gradient and updated recursively after each gradient evaluation
- computes the Hessian matrix, which is very useful for sensitivity analysis
- examples of gradient methods:
 - * Newton's Method
 - * Sequential Quadratic Programming (SQP)

3 Penalty Functions for Constraints

Search methods for constrained optimization incorporate penalty functions in order to satisfy the constraints. By augmenting the objective $f(\mathbf{x})$ with a positive-valued penalty function that increases monotonically with the values of constraint violations, the constrained optimization problem is transformed into an unconstrained optimization problem. Penalty functions can follow a linear, power-law, or hyperbolic relation to the severity of the penalty violation. For linear and power-law penalty functions, the objective is augmented as follows:

$$f^{\text{aug}}(\mathbf{x}) = f(\mathbf{x}) + \mathcal{P} \left[\sum_{j=1}^m \langle g_j(\mathbf{x}) \rangle \right]^q \quad (4)$$

where \mathcal{P} is a positive valued *penalty factor*. The *Macaulay bracket* $\langle g \rangle$ is a *ramp function* for which $\langle g \rangle = g$ if $g > 0$ and is zero otherwise. The *penalty exponent* q is positive-valued and can help with convergence to feasible solutions. (If $q = 1$, the penalty function is linear in the constraint violations.)

4 Convergence Criteria

If one or more of the following conditions are satisfied, the solution is considered to be *sufficiently accurate*. The desired accuracy is defined by user-specified positive-valued tolerances: ϵ_1 for the accuracy of the parameter values, ϵ_2 for the accuracy of the objective function, and ϵ_3 for allowable constraint violations.

- convergence in the optimization parameters : $\max(|h_i|/|x_i|) < \epsilon_1$
- convergence in the objective function : $|f(\mathbf{x} + \mathbf{h})|/|f(\mathbf{x})| < \epsilon_2$
- feasibility of the constraints : $\max(g_j(\mathbf{x})) < \epsilon_3$
- maximum allowable function evaluations : $n_{\text{feval}} \geq N_{\text{max}}$

This last condition terminates the iteration after a specified number of objective function evaluations. It is a termination criterion, not a convergence criterion.

References

- [1] Paul T. Boggs, and Jon W. Tolle, “[Sequential Quadratic Programming](#),” *Acta Numerica*, (1995): 1-51.
- [2] J.A. Nelder and R. Mead, “[A simplex method for function minimization](#),” *Computer Journal*, 7(4) (1965): 308-313.
- [3] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.
- [4] S.S. Rao, *Optimization, Theory and Applications*, John Wiley and Sons, 1984.
- [5] James C. Spall, “[Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation](#),” *IEEE Transactions on Automatic Control*, 37(3) (1992): 332-341.
- [6] James C. Spall, “[An Overview of the Simultaneous Perturbation Method for Efficient Optimization](#),” *Johns Hopkins APL Technical Digest*, 19(4) (1998): 482-492.
- [7] B.V. Sheela, “[An Optimized Step Size Random Search \(OSSRS\)](#),” *Computer Methods in Applied Mechanics and Engineering*, 19 (1979): 99-106.