A Parallel, Asynchronous Method for Derivative-Free Nonlinear Programs

Joshua D. Griffin and Tamara G. Kolda

Sandia National Laboratories, Livermore, CA 94551, USA
{jgriffi, tgkolda}@sandia.gov
http://csmr.ca.sandia.gov/~{jgriffi, tgkolda}

Derivative-free optimization algorithms are needed to solve real-world engineering problems that have computationally expensive and noisy objective function and constraint evaluations. In particular, we are focused on problems that involve running cumbersome simulation codes with run times measured in hours. In such cases, attempts to compute derivatives can prove futile because analytical derivatives are typically unavailable and noise limits the accuracy of numerical approximations. Furthermore, the objective and constraint functions may be inherently nonsmooth, i.e., because the underlying model is nonsmooth.

Generating Set Search (GSS) methods [7] are particularly well-suited to such unwieldy optimization problems. GSS methods are a generalization of pattern search that derives its search directions from the generators of the ϵ-tangent cone of the linear constraints, i.e., a generating set. GSS methods offer several advantages:

- Because search direction are based upon the local geometry of the feasible region defined by the linear constraints, and not the objective or nonlinear constraint functions, they are well-suited for problems with noise.
- The function evaluations can be performed asynchronously in parallel [5, 10, 6].
- If the underlying objective function and constraints are smooth, GSS methods can bound the first-order optimality conditions in terms of step size.
- They can easily accommodate undefined points within the feasible region, i.e, points where the simulation unexpectedly fails.

The focus of this talk will be on the addition of constraint-handling abilities to APPSPACK, which is a C++ implementation of an asynchronous parallel GSS algorithm [4]. APPSPACK is a publicly available derivative-free software package whose handling of both linear and nonlinear constraints is based upon rigorous convergence theory. Specifically, we have added the ability to handle linear constraints using conforming search directions and nonlinear constraints using an augmented Lagrangian algorithm.

We are interested in solving a nonlinear program of the form

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) = d \\ & \ell \le Ax \le u. \end{aligned} \tag{1}$$

Here $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $c : \mathbb{R}^n \to \mathbb{R}^p$ denotes the nonlinear equality constraints, and A denotes an $n \times m$ linear constraint matrix. We allow

A. Iglesias and N. Takayama (Eds.): ICMS 2006, LNCS 4151, pp. 260–262, 2006.

for both linear equality and inequality constraints. We assume that evaluating $f(x)$ and $c(x)$ is expensive and derivatives are unavailable.

Consider first the problem of linear constraints. APPSPACK's linear constraint support is based on [11, 8]. We generate a core set of search directions, as outlined in [11], from generators of tangent cones corresponding to nearby constraints. The solid lines in 1 show the "conforming" search directions to the nearby boundary (as defined by the ϵ-ball that is drawn). The dashed lines indicate additional directions that we might choose to add to further accelerate the search. This choice of search directions guarantees that we have at least one feasible descent direction, if one exists. If the nearby constraints are nondegenerate, finding the search directions is a straightforward linear algebra problem; otherwise, more sophisticated machinery is required and we use `cddlib` [3] to find the appropriate generators.

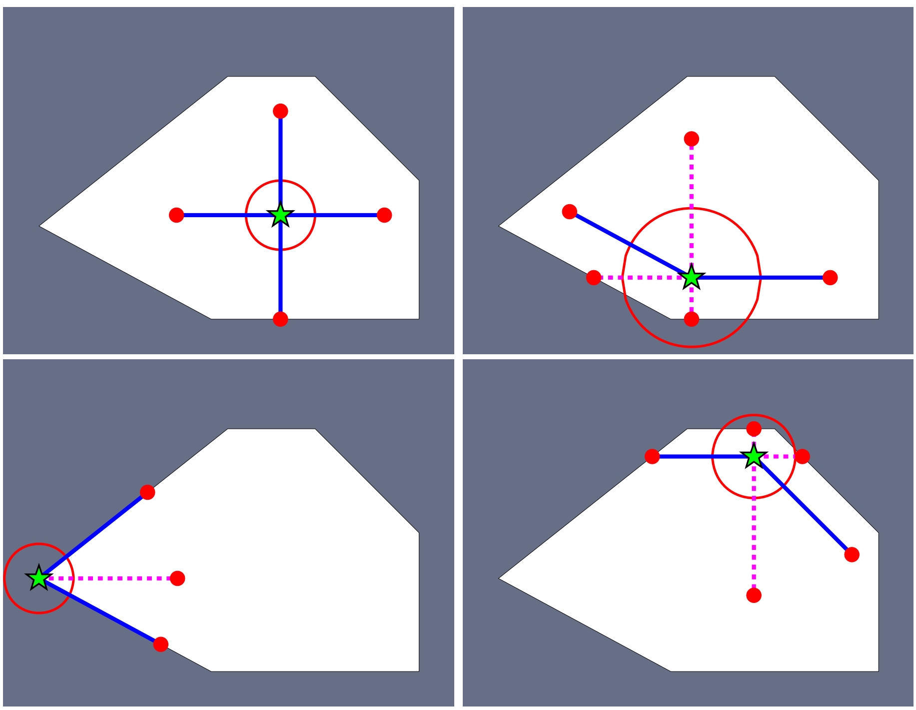

Fig. 1. At each iteration an ϵ-ball is formed about the current point to determine nearby constraints and corresponding search directions added. Extra directions (denoted by dashes) are generally added to facilitate movement toward and away from the boundary.

APPSPACK handles nonlinear equality constraints using an augmented Lagrangian method proposed in [2, 1] and extended to the directive-free case in [12, 9]. Remarkably, the derivative-free variant retains the same theoretical convergence properties as the original derivative-based augmented Lagrangian approach. This algorithm can be divided into inner and outer iterations; inner iterations being devoted to approximately solving the linearly-constrained subproblem

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & \phi_k(x) \equiv f(x) - \lambda_k^T c(x) + \frac{1}{2\mu_k}\|c(x)\|^2 \\ \text{subject to} & \ell \leq Ax \leq u \end{array} \tag{2}$$

for fixed λ_k and μ_k, while outer iteration are used to assess optimality and update the subproblem specific parameters.

This talk will begin with a brief background of GSS methods and follow with a description of how linear and nonlinear constraints are handled in APPSPACK. Theory and numerical results will be provided.

References

1. A. Conn, N. Gould, A. Sartenaer, and P. Toint, *Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints*, SIAM J. Optimiz., 6 (1996), pp. 674–703.
2. A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.
3. K. Fukuda, *cdd and cddplus homepage*. From McGill University, Montreal, Canada Web page: `http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html`, 2005.
4. G. A. Gray and T. G. Kolda, *Algorithm 8xx: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization*, ACM T. Math. Software. To appear.
5. P. D. Hough, T. G. Kolda, and V. J. Torczon, *Asynchronous parallel pattern search for nonlinear optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 134–156.
6. T. G. Kolda, *Revisiting asynchronous parallel pattern search for nonlinear optimization*, SIAM J. Optimiz., 16 (2005), pp. 563–586.
7. T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.
8. ———, *Stationarity results for generating set search for linearly constrained optimization*, Tech. Rep. SAND2003-8550, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, Oct. 2003.
9. ———, *Convergence properties of an augmented Lagrangian direct search algorithm for optimization with a combination of general equality and linear constraints*. In preparation, 2005.
10. T. G. Kolda and V. Torczon, *On the convergence of asynchronous parallel pattern search*, SIAM J. Optimiz., 14 (2004), pp. 939–964.
11. R. M. Lewis, A. Shepherd, and V. Torczon, *Implementing generating set search methods for linearly constrained minimization*, Tech. Rep. WM-CS-2005-01, Department of Computer Science, College of William & Mary, Williamsburg, Virginia, July 2005.
12. R. M. Lewis and V. Torczon, *A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, SIAM J. Optimiz., 12 (2002), pp. 1075–1089.