MANAGING APPROXIMATION MODELS IN OPTIMIZATION

J. E. Dennis, Jr. *
Department of Computational and Applied Mathematics
Rice University, P. O. Box 1892
Houston, Texas 77251-1892

Virginia Torczon †
Department of Computer Science
College of William and Mary
Williamsburg, Virginia 23187-8795

Abstract

A standard engineering practice is the use of approximation models in place of expensive simulations to drive an optimal design process based on nonlinear programming algorithms. The use of approximation techniques is intended to reduce the number of detailed, costly analyses required during optimization while maintaining the salient features of the design problem.

The question we address is how to manage the interplay between the optimization and the fidelity of the approximation models to ensure that the process converges to a solution of the original design problem. Using well-established notions from the literature on trust-region methods and a powerful global convergence theory for pattern search methods, we can ensure that the optimization process converges to a solution of the original design problem.

Key Words: Approximation Models, Nonlinear Programming, Nonlinear Optimization, Global Convergence, Trust Region, Pattern Search.
AMS(MOS) subject classification: 65K05, 49D37

Introduction

The desire is to find x^* to minimize $f(x, y(x))$, where x represents the control variables, $y(x)$ represents the state equations, and f is the design objective. We assume that the control variables x are subject to some constraints. In addition, we assume:

- $y(x)$ is never available, but computational procedures are available to compute or estimate some $M^*(x) \approx y(x)$;
- models $M^\alpha(x) \approx M^*(x)$ can be built with increasing "accuracy" at additional computational cost.

The task is to minimize $f(x, M^*(x))$ by approximately optimizing appropriately chosen $f(x, M^\alpha(x))$.

The Art of Model Management

Suppose we want to solve the following general minimization problem:

$$\min_z \phi(z), \text{ subject to } z \in \mathcal{B},$$

where $\mathcal{B}$ denotes the feasible region for the optimization problem.

Then given a putative solution $z_k \approx z^*$ we can follow the general strategy:

- If convergence, then exit; otherwise, continue.
- Build a global model G of ϕ on $\mathcal{B}$.
- Build a local model m of G at z_k.
- From m obtain a z_{k+}^{trial} that improves G.
- If z_{k+}^{trial} improves ϕ, then $z_{k+1} = z_{k+}^{trial}$; otherwise, either backtrack on the amount of optimization applied to G to obtain a more conservative choice of z_{k+}^{trial}, or refine the global model G and repeat the process.

This simple strategy leads the the following framework for using approximation models in optimization.

‡Member AIAA. The work of the first author was supported by the Air Force Office of Scientific Research under F49620–95–1–0210, and IBM under ?????.

†The work of the second author was supported by the Air Force Office of Scientific Research under grant F49620–95–1–0210.

Proposed Optimization Framework Using Approximation Models

Given M^α, M^*, $x_0 \approx x^*$, $y_0^* = M^*(x_0)$:
For $k = 0, 1, \cdots$, do

1. If convergence, then exit; otherwise, continue.

2. Apply an optimization algorithm to the approximate problem to find an x_{k+} for which $f(x_{k+}, y_{k+}^\alpha)$ satisfies an appropriate decrease condition for $f(x, M^\alpha(x))$ from x_k. (This could mean something like "do a complete optimization for the problem defined by $f(x, M^\alpha(x))$" or "take some (fixed) number of optimization iteration on the problem defined by $f(x, M^\alpha(x))$."

   Compute $pred_k \equiv f(x_k, y_k^*) - f(x_{k+}, y_{k+}^\alpha)$. This is the amount of reduction that the approximate problem defined by $f(x, M^\alpha(x))$ *predicts* if the trial solution x_{k+} is applied to the true problem $f(x, M^*(x))$.

3. Compute $y_{k+}^* = M^*(x_{k+})$ using either a detailed analysis or adaptive heuristics.

   Computer $ared_k = f(x_k, y_k^*) - f(x_{k+}, y_{k+}^*)$. This is the amount of reduction realized by the trial step x_{k+} when applied to the *actual* problem $f(x, M^*(x))$

4. If $\frac{ared_k}{pred_k} \leq 0$, then (improvement was predicted but not achieved)

   Set $x_{k+1} = x_k$ and $y_{k+1}^* = y_k^*$.
   (Reject the step.)

   Get a more faithful model M^α anchored at x_k.
   (Refine the model.)

   Allow less optimization on the approximate problem at the next iteration.

   Else, if $0 < \frac{ared_k}{pred_k} \leq 10^{-4}$, then (much more improvement was predicted than achieved)

   Set $x_{k+1} = x_{k+}$ and $y_{k+1}^* = y_{k+}^*$.
   (Accept the step.)

   Update the current M^α to interpolate to y_{k+1}.

   Allow less optimization on the approximate problem at the next iteration.

   Else, if $10^{-4} < \frac{ared_k}{pred_k} \leq 0.5$, then (the prediction was satisfactory)

   Set $x_{k+1} = x_{k+}$ and $y_{k+1}^* = y_{k+}^*$.
   (Accept the step.)

   Update the current M^α to interpolate to y_{k+1}.

   Allow the same optimization on the approximate problem at the next iteration.

   Else, $0.5 < \frac{ared_k}{pred_k}$ (the prediction was excellent, or more decrease was obtained than predicted)

   Set $x_{k+1} = x_{k+}$ and $y_{k+1}^* = y_{k+}^*$.
   (Accept the step.)

   Update the current M^α to interpolate to y_{k+1}.
   Consider using a *less* accurate approximation model.

   Allow more optimization on the approximate problem at the next iteration.

5. Return to Step 1.

To finish the specification for the general algorithm we must determine:

- how to find a trial step x_{k+},
- what constitutes an appropriate decrease condition for $f(x, M^\alpha(x))$ from x_k,
- how to update the amount of optimization to be done on the approximate problem in Step 2, and
- how to incorporate adaptive heuristics to estimate y_{k+}^*.

We use well-established notions from the literature on trust-region methods [1] and a powerful global convergence theory for pattern search methods [2] to manage this interplay between optimization and the fidelity of the approximation models. A careful use of these techniques insures that the process converges under the mild condition that at any given point at which a full simulation has been run, the approximation model agrees with the objective function of the optimization problem [3].

Computational testing is underway and will be the subject of future reports.

References

[1] J.E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[2] Virginia Torczon. On the convergence of pattern search algorithms. To appear in SIAM Journal on Optimization, February 1997.

[3] J.E. Dennis and Virginia Torczon. Managing approximation models in optimization. In Natalia Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State-of-the-Art*, SIAM, Philadelphia, 1996.