Experience with Approximations in the Trust-Region Parallel Direct Search Algorithm

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

May 25, 2009

Talk Outline

Motivation:

- Challenges of Simulation-Based Optimization

Algorithms:

- Trust-Region Parallel Direct Search (TRPDS)
- mTRPDS
- Speculative Gradient

Our Contribution:

- Use of generalized approximation models in TRPDS

Numerical Experiments:

- Comparisons with TRPDS and Speculative Gradients

Conclusions

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

Simulation-Based Optimization

Simulation-Based Optimization: Coupling optimization software with simulations.

Examples:

- Identify parameters for new model such that simulation and experimental results most closely match.
- Identify parameters to determine optimal device design.

Challenges of Simulation-Based Optimization

1. Analytic gradients might not be available (use finite-differences)
2. Function evaluations can be expensive (dominant cost)

Our Goal: To reduce the number of function evaluations by leveraging parallelism and approximation models.

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

Parallelization of Optimization Algorithms

Focus on variant of trust-region algorithms:

- Trust-Region Parallel Direct Search (TRPDS) algorithm (Hough, Meza, 2002)

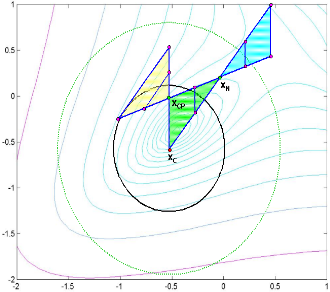
Our Contribution:

- Employ approximation models to further reduce computational cost.

Competing trust-region variant:

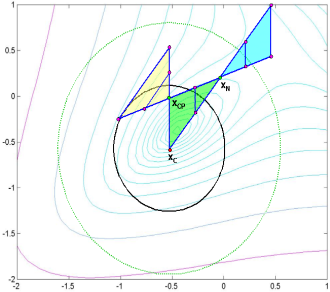
- Speculative gradient technique (Byrd, Schnabel, Shultz, 1988)

Trust-Region Parallel Direct Search (TRPDS) Algorithm



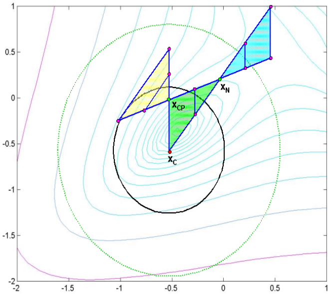
Trust-Region Parallel Direct Search (TRPDS) Algorithm

An Iteration of TRPDS



Trust-Region Parallel Direct Search (TRPDS) Algorithm

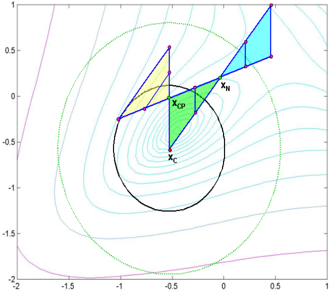
An Iteration of TRPDS



1. Minimize quadratic model

Trust-Region Parallel Direct Search (TRPDS) Algorithm

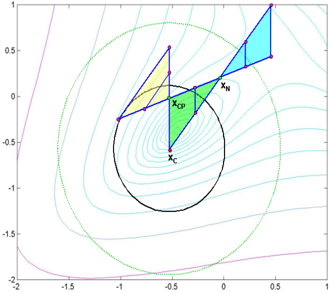
An Iteration of TRPDS



1. Minimize quadratic model
2. Form simplex

Trust-Region Parallel Direct Search (TRPDS) Algorithm

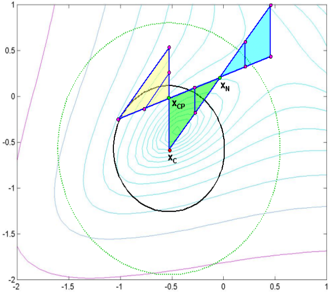
An Iteration of TRPDS



1. Minimize quadratic model
2. Form simplex
3. Concurrently evaluate $f(x)$ at simplex points; choose lowest value.

Trust-Region Parallel Direct Search (TRPDS) Algorithm

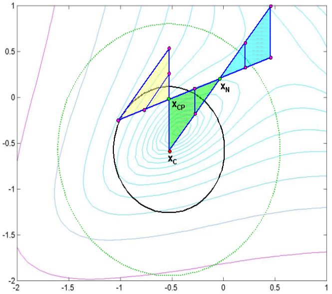
An Iteration of TRPDS



1. Minimize quadratic model
2. Form simplex
3. Concurrently evaluate $f(x)$ at simplex points; choose lowest value.
4. Check sufficient decrease

Trust-Region Parallel Direct Search (TRPDS) Algorithm

An Iteration of TRPDS



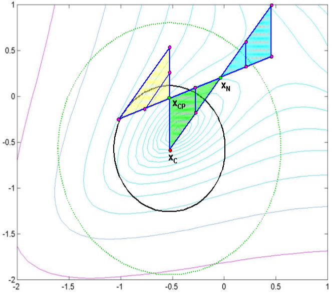
1. Minimize quadratic model
2. Form simplex
3. Concurrently evaluate $f(x)$ at simplex points; choose lowest value.
4. Check sufficient decrease
5. Accept/Reject trial iterate

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

Trust-Region Parallel Direct Search (TRPDS) Algorithm

An Iteration of TRPDS



1. Minimize quadratic model
2. Form simplex
3. Concurrently evaluate $f(x)$ at simplex points; choose lowest value.
4. Check sufficient decrease
5. Accept/Reject trial iterate
6. Update trust region

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

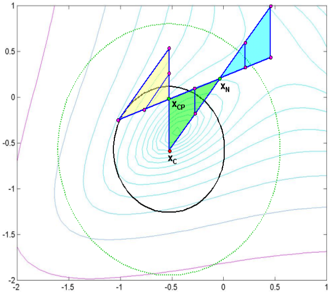
International Conference on Computational Science
Baton Rouge, LA

Trust-Region Parallel Direct Search (TRPDS) Algorithm

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

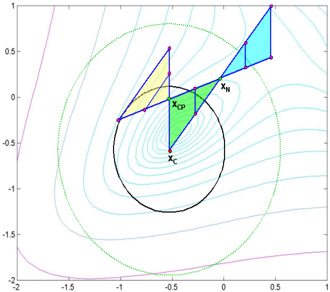
An Iteration of TRPDS



1. Minimize quadratic model
2. Form simplex
3. Concurrently evaluate $f(x)$ at simplex points; choose lowest value.
4. Check sufficient decrease
5. Accept/Reject trial iterate
6. Update trust region
7. Goto 1

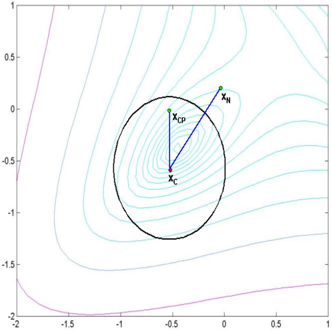
mTRPDS Algorithm

An Iteration of mTRPDS



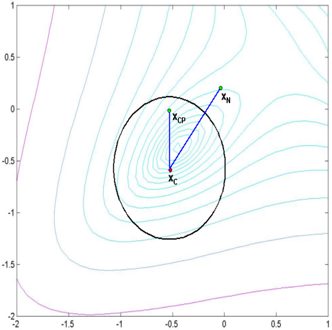
1. Minimize quadratic model
2. Form simplex
3. Use PDS to find the j lowest model values Evaluate $f(x)$ concurrently at those points. Choose lowest value.
4. Check sufficient decrease
5. Accept/Reject trial iterate
6. Update trust region
7. Goto 1

An Iteration of Speculative Gradient



Speculative Gradient Algorithm

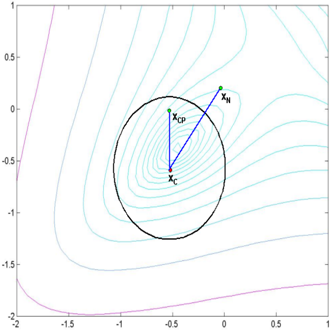
An Iteration of Speculative Gradient



1. Minimize quadratic model over trust region

Speculative Gradient Algorithm

An Iteration of Speculative Gradient



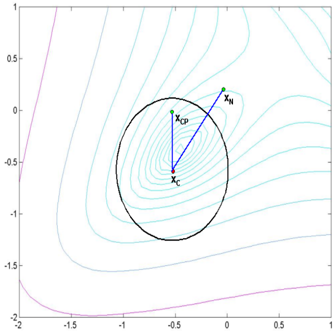
1. Minimize quadratic model over trust region
2. Processor 0 evaluates trial iterate; Remaining processors evaluate gradient

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

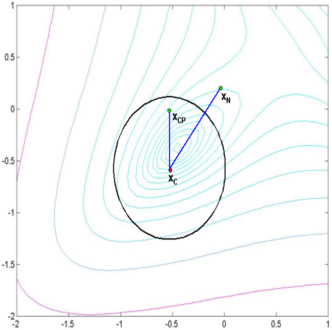
Speculative Gradient Algorithm

An Iteration of Speculative Gradient



1. Minimize quadratic model over trust region
2. Processor 0 evaluates trial iterate; Remaining processors evaluate gradient
3. Check sufficient decrease

An Iteration of Speculative Gradient



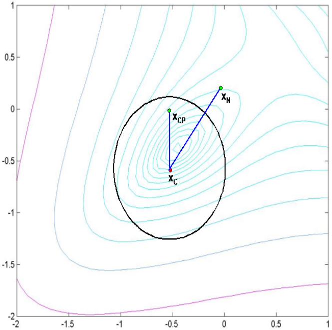
1. Minimize quadratic model over trust region
2. Processor 0 evaluates trial iterate; Remaining processors evaluate gradient
3. Check sufficient decrease
4. Accept/Reject trial iterate

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

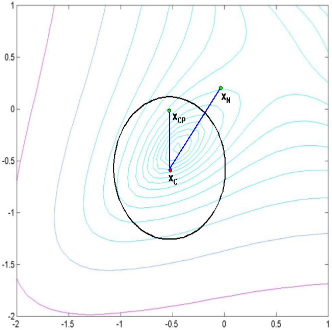
Speculative Gradient Algorithm

An Iteration of Speculative Gradient



1. Minimize quadratic model over trust region
2. Processor 0 evaluates trial iterate; Remaining processors evaluate gradient
3. Check sufficient decrease
4. Accept/Reject trial iterate
5. Update trust region

An Iteration of Speculative Gradient



1. Minimize quadratic model over trust region
2. Processor 0 evaluates trial iterate; Remaining processors evaluate gradient
3. Check sufficient decrease
4. Accept/Reject trial iterate
5. Update trust region
6. Goto 1

Generalized Trust-Region Framework

- The framework is for managing the use of approximation models (Alexandrov, Dennis, Lewis, Torczon, 1998).
- An *approximation model* $a_k(\mathbf{x}_k)$ is a less expensive representation of $f(\mathbf{x}_k)$.
- The trust-region method with generalized approximation models converges globally whenever:
  1. $a_k(\mathbf{x}_k) = f(\mathbf{x}_k)$
  2. $\nabla a_k(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)$.
- Steps must be computed such that the sequence of iterates produced satisfies the fraction of Cauchy decrease (FCD) condition $\Rightarrow$ Flexibility!

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

Generalized Trust-Region Framework

Model management framework:

- At iteration k, an approximation model, $m_k(\mathbf{x}_k)$, to the objective function, $f(\mathbf{x}_k)$, is built. Then, the following PDS subproblem is solved approximately:

$$\begin{gathered}\min m_k(\mathbf{x}_k + \mathbf{s}) \\ s.t.\ \|\mathbf{s}\|_2 \leq 2\delta_k.\end{gathered}$$

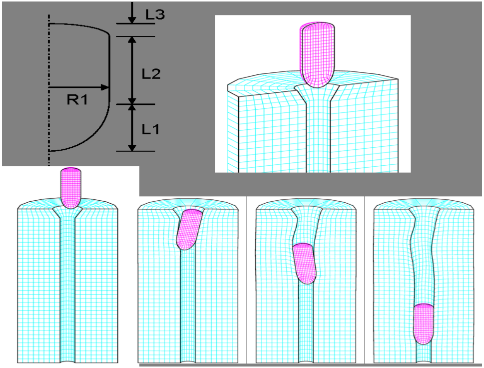
- In all cases, we use a quadratic model for a_k (to check FCD). Note FCD is satisfied by construction for Speculative Gradient.

- We use various approximation models for m_k to determine the step: $m_k =$ truth (TRPDS), generalized approximation model (mTRPDS), and quadratic model (Speculative Gradient).

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

Numerical Experiments

Case Study: Earth Penetrator Design Problem



Goal: To find section lengths that will optimize mission performance. The earth penetrator radius is held fixed, while the lengths are varied independently.

Optimal Design Problems (1)

Problem 1: Minimize maximum acceleration subject to bounds on length parameters

$$\begin{aligned} \min_{\mathbf{L} \in \mathbb{R}^3} \quad & F(\mathbf{L}) = \max\,(\text{acceleration}) \\ \text{s.t.} \quad & l_i \leq L_i \leq u_i, i = 1 \ldots 3, \end{aligned} \tag{1}$$

where $\mathbf{L}$ is the vector containing the three unknown length parameters, L_i, and l_i and u_i are the lower and upper bounds, respectively.

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

Optimal Design Problems (2)

Problem 2: Maximize penetration depth subject to bounds on length parameters.

$$\begin{aligned} \min_{\mathbf{L} \in \mathbb{R}^3} \quad & F(\mathbf{L}) = -(\text{depth of penetration}) \qquad (2) \\ \text{s.t.} \quad & l_i \le L_i \le u_i, i = 1 \ldots 3. \end{aligned}$$

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA

Optimization Experiments

Solve optimal design problems (1) and (2) via:

1. *m*TRPDS
2. TRPDS
3. Trust-Region Speculative Gradient

***m*TRPDS Approximations:**

1. Alter mesh discretization
2. Alter amount of event time simulated
3. Use Taylor series to construct quadratic model of function

Employed: Central finite difference gradient, BFGS approximation to Hessian

Compared: Timing results

Optimization Experiments

Penetrator design simulation: 3 variables, so 7 simultaneous $f(x)$ evaluations to compute function and gradient. Used 16 processors per simulation + 1 processor for optimization process. **Total: 113 processors**

m**TRPDS experiments:** Used 113 processors.
Ideal settings: search pattern size of 7, $j = 7$.

Optimization algorithms: Implemented in OPT++.

Approximation Models Employed

Key	Algorithm	Model	Time for Single Model Execution
1	SpecGrad	Truth (Mesh640k, Time25ms)	2 – 3 hours
2	TRPDS	Truth (Mesh640k, Time25ms)	2 – 3 hours
3	*m*TRPDS	QuadraticModel	negligible
4	*m*TRPDS	Mesh10k	0.8 – 1.3 hours
5	*m*TRPDS	Mesh80k	1.3 – 1.8 hours
6	*m*TRPDS	Time6.25ms	1.1 – 1.6 hours
7	*m*TRPDS	Time12.5ms	1.7 – 2.3 hours

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

Wall Clock Time: Problem # 1

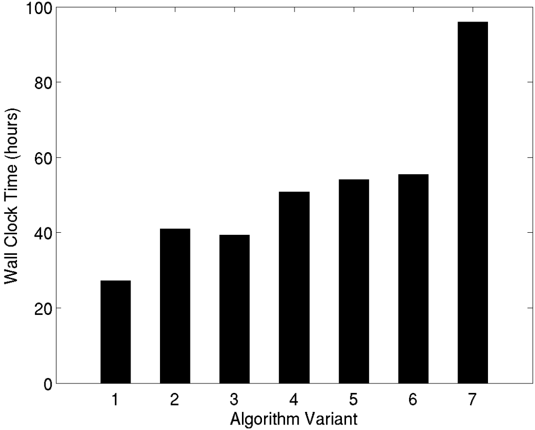


Figure: Wall clock time required to achieve a 0.1% change in the function

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

Average Time Per Iteration: Problem # 1

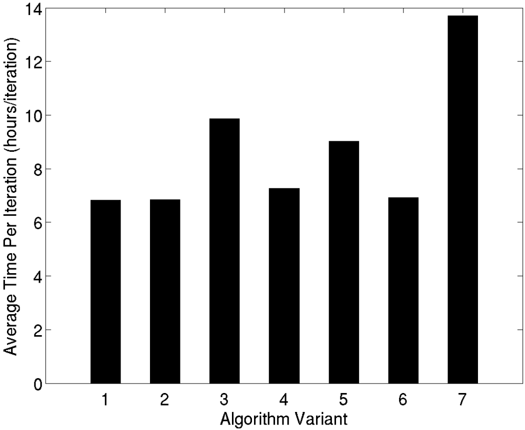


Figure: Average time per iteration

Problem # 1 Results

- Primary difference in wall clock times is due to different number of iterations required.
- However, several algorithm-model combinations have comparable average times per iteration.
- TRPDS-based algorithms start out same as speculative gradient algorithm. However, they move towards solutions with lower function values. Thus, take longer.
- Further characterization of problem features on algorithmic performance is needed.
- Require computationally expensive, physics-based test problems. Hard to come by.

Wall Clock Time: Problem # 2

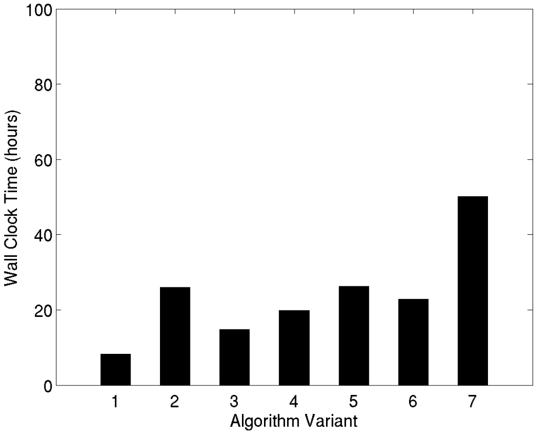


Figure: Wall clock time required to achieve a 0.1% change in the function

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

Average Time Per Iteration: Problem # 2

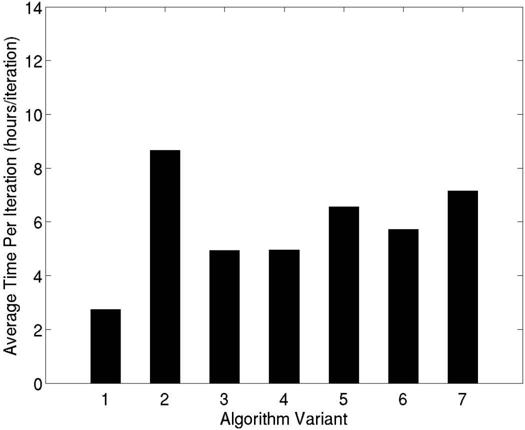


Figure: Average time per iteration.

Problem # 2 Results

Algorithms took approximately same number of iterations. Variations in average wall clock time per iteration.

Suggestions for improving computational efficiency:

- Approximation models track truth fairly well. Reduce j and incorporate computation of speculative finite-difference gradients for those j points.
- Best might be $j = 0$. Would reduce number of truth evaluations per iteration, hence reducing total time.
- PDS sometimes performs extraneous approximation evaluations. Need dynamic scheme for managing amount of PDS work.

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

Convergence Patterns

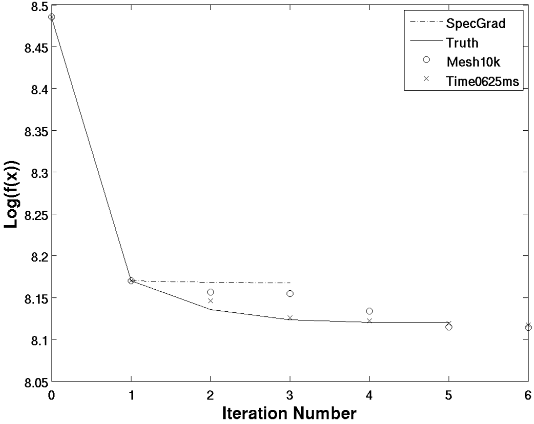


Figure: Problem # 1: TRPDS-based algorithms move to a solution with a lower function value, thereby taking more iterations.

Conclusions

1. Extended TRPDS algorithm to include use of approximation model to solve PDS subproblem. Parallelism used.
2. Performed numerical experiments on two earth penetrator optimal design problems. Compared with TRPDS and speculative gradient trust-region method.
3. Made suggestions for improving efficiency.

Longer-Term Research

1. Develop meaningful numerical stopping criteria for optimization algorithms.
2. Characterization of effects of problem characteristics on algorithm performance.

Acknowledgments

Funding:

- Department of Energy
- National Physical Science Consortium

People:

- John Dennis, Juan Meza, Virginia Torczon (suggestions on TRPDS-based algorithms)
- Monica Martinez-Canales, Genetha Gray, Laura Swiler, Mike Chiesa, Randy Settgast (assistance with earth penetrator application)

Earth Penetrator Simulation Details

Mechanical deformation of penetrator upon impact: Presto, 3D explicit transient dynamics code; Lagrangian finite elements (Koteras, Guillerud, Crane, Hales, Reinert, 2007)

Contact algorithms: ACME library

Penetrator: Modeled as homogeneous elastic solid.

Target: Modeled with Mohr-Coulomb constitutive model.

Parametric meshes: Generated using CUBIT. Elements are eight-node hex elements.

Time step: Chosen to satisfy Courant stability condition.

Simulations: Performed on Linux cluster.

Suzanne Shontz (Penn State)
Victoria Howle (Texas Tech)
Patricia Hough (Sandia National Labs)

International Conference on Computational Science
Baton Rouge, LA