An Asynchronous and Fault-Tolerant Algorithm for Parallel Direct Search Optimization

Patricia D. Hough
Tamara G. Kolda
Sandia Nationals Labs, Livermore

Virginia Torczon
College of William & Mary

Research Support

Applied Mathematical Sciences Research Program,
U.S. Department of Energy.

Sandia National Laboratories.

National Science Foundation.

What is Direct Search?

$$\boxed{\min f(x) \quad x \in \Re^n}$$

A direct search method is a derivative-free optimization method further classified by the fact that it does not 'in its heart' develop an approximate gradient. (cf. Wright, 1996)

When is It Appropriate?

- Calculation of f dominates the cost of an iteration.
- Gradients of f cannot be calculated or approximated.
- The number of dimensions is small.

Outline

- Parallel Direct Search (PDS)
- Asynchronous PDS
- Fault-Tolerance
- Conclusions & Future Work

Parallel Direct Search (PDS)

Initialization:

- Select a template: $\{d_1, \ldots, d_p\}$.
- $\delta_0 \leftarrow 1$.
- Select starting point x_0 and evaluate $f(x_0)$.

Iteration:

1. Evaluate $f(x_k + \delta_k d_i)$ for $i = 1, \ldots, p$ in parallel.

— Synchronization Point —

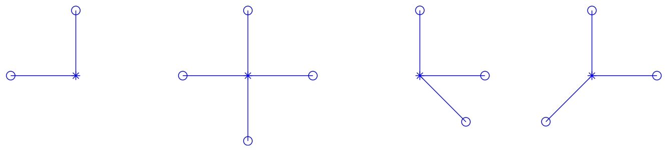
2. If $f(x_k + \delta_k d_i) < f(x_k)$ for some i then set $x_{k+1} = x_k + \delta_k d_i$ and $\delta_{k+1} = \delta_k$. Else set $x_{k+1} = x_k$ and $\delta_{k+1} = \frac{1}{2}\delta_k$.
3. If $\delta_{k+1} <$ `tol`, exit. Else, $k \leftarrow k + 1$ and repeat.

Positive Spanning Set

A set of vectors $\{d_1, \ldots, d_p\}$ positively spans $\Re^n$ if any vector $x \in \Re^n$ can be written as

$$x = \alpha_1 d_1 + \cdots + \alpha_p d_p, \quad \alpha_i \geq 0 \quad \forall\, i.$$

That is, any vector can be written as a *positive* linear combination of the vectors.



Suppose $\{d_1, \cdots, d_r\}$, $d_i \neq 0$, linearly spans $\Re^n$. TFAE:

1. $\{d_1, \cdots, d_r\}$ positively spans $\Re^n$.
2. $\forall\, x \neq 0$, $\exists\, i$ such that $x^T d_i > 0$.
3. $\forall\, i \in \{1, \cdots, r\}$, $-d_i$ is in the convex cone spanned by the remaining d_i.

Why do we need Asynchronous PDS?

- Heterogeneous computing environments.
- Varying loads on nodes.
- Function evaluations can takes different amounts of time.
- Easier to introduce fault-tolerance.
- Can do it at little extra cost!

Task Relationship

Master-Slave or *Peer-to-Peer*?

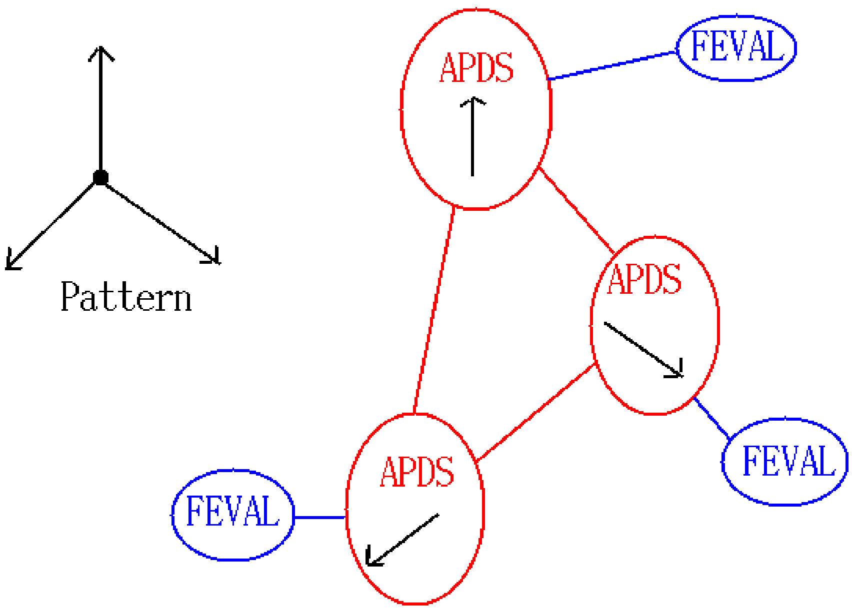
Master-Slave Scenario:

- Master assigns function evaluations to Slaves.
- All messages are to or from the Master.
- Master cannot fail. (Problem)
- Any Slave can fail, easy recovery.

Peer-to-Peer Scenario:

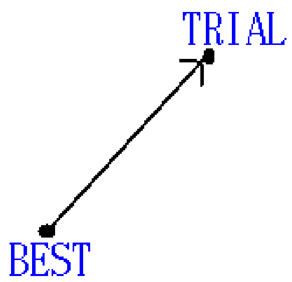
- Each Peer handles one search direction.
- Each process determines what to do next on its own.
- New information is broadcast to all Peers when necessary.
- Any Peer can fail, easy recovery.

APDS Daemons



Local Collection of Points

- BEST - Point with the best known function value. Usually the point used to general the TRIAL point, but may be different if a better incoming point received during function evaluation.
- TRIAL - Point sent to function evaluator. Generated from current BEST point.
- INCOMING - Potential new best point received from another APDS daemon.



- coordinates
- f-value
- δ
- convergence
- unique id

MESSAGE: RETURN FROM f-EVAL

1. If f(TRIAL) $<$ f(BEST) then
   (a) BEST $\leftarrow$ TRIAL
   (b) Broadcast BEST to other processors.
   (c) Create new TRIAL point and spawn new f-eval.
2. Else if BEST $\neq$ TRIAL GENERATOR then create new TRIAL point and spawn new f-eval.
3. Else $\delta = \delta/2$. Is $\delta >= $ TOL?

Yes. Create new TRIAL point and spawn new f-eval.

No. Broadcast convergence message and wait.

SIAM Conference on Optimization May 10, 1999

MESSAGE: NEW MINIMUM

If f(INCOMING) $<$ f(BEST) then

1. BEST $\leftarrow$ INCOMING.
2. If δ(INCOMING) $>=$ δ(TRIAL) then
   (a) Break current f-eval spawn.
   (b) Create new TRIAL point and spawn new f-eval.

MESSAGE: CONVERGENCE

1. Check that INCOMING satisifies conditions of a new minimum.
2. If INCOMING matches BEST, merge convergence information.
3. If the set of converged direction vectors forms a positive spanning set then print solution and shutdown remaining processes. *Should be agreement on who performs this check.*

Checking Positive Basis

Given a set of vectors $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$, how do we check to see if $\mathcal{V}$ is a positive spanning set?

1. Verify that $\mathcal{V}$ is a spanning set using, e.g., a QR factorization.
2. Let $V = [v_1 v_2 \cdots v_m]$ denote the matrix whose columns are the vectors in $\mathcal{V}$. Let e denote the vector of all ones. Solve the LP:

$$\max \quad t \quad \text{s.t.} \quad Vx = 0, \quad x \geq te, \quad t \leq 1. \qquad (*)$$

- $(x, t) = (0, 0)$ is feasible.
- The solution to $(*)$ is 1 iff V is a positive spanning set. Otherwise, the solution is 0.

Solution due to Steve E. Wright at Miami Univ in Ohio.

Fault-Tolerance

- We can afford to lose tasks as long as we maintain a template that positively spans $\Re^n$.
- We can restart jobs when we no longer have a positive basis.
- No check-pointing required!!
- Progression towards a solution should continue as long as one or more APDS processes is alive. Does *not* depend on the survival of any particular daemon.
- PVM is not entirely fault-tolerant — it depends on the survival of the *PVM master daemon.*

FT Situations

- Failure in function evaluations — Just restart.
- APDS daemon failure — Restart if necessary to form a positive basis. *Should be agreement on who performs this restart.*
- New task — Update communication information.
- Host failure — Log that the host cannot be used anymore.

Convergence Analysis for Positive Basis Direct Search

Hypothesis on Exploratory Moves (HEM): No contraction until all elements in a *base template* are checked.

Theorem: Suppose $L(x_0) \equiv \{x : f(x) \leq f(x_0)\}$ is compact and f is continuously differentiable on an open neighborhood Ω of $L(x_0)$. Let $\{x_k\}$ be the sequence of iterates produced by the positive spanning set direct search method satisfying HEM. Then

$$\liminf_{k \to +\infty} \|g(x_k)\| = 0$$

Strong Hypothesis on Exploratory Moves (SHEM): The least element in a *base template* must be chosen.

Theorem: Add that the search satisfies the SHEM. Then

$$\lim_{k \to +\infty} \|g(x_k)\| = 0$$

Lewis & Torczon, 1996

Convergence Analysis for APDS

Forthcoming...

Basis Problem: We can easily guarantee that all of the points at length, say, 1 are checked, but we cannot guarantee that an *even better* point than any of those won't be found that as the result of a contraction along one leg.

Some hope? "On the Global Convergence on Derivative Free Methods for Unconstrained Optimization" by Lucidi and Sciandrone.

Review

- Introduced an asynchronous PDS algorithm,
- Added fault-tolerant to APDS algorithm.
- Verifying a positive basis?
- Convergence analysis and numerical results forthcoming.

Future Work

- Constraints (bounds, linear, nonlinear).
- Incorporate fuzzy gradient information.
- Process migration.
- Dynamic basis changing.
- Move to better communication architecture then PVM.
- Work-arounds for "Curse of Dimensionality".