Optimal Placement of Tsunami Warning Buoys using Mesh Adaptive Direct Searches

Charles Audet, Gilles Couture, École Polytechnique de Montréal

John Dennis, Rice University

LtCol Mark Abramson, AFIT

Frank Gonzalez, Hal Mofjeld
NOAA Pacific Marine Environmental Lab(PMEL)

Vasily Titov, Mick Spillane, University of Washington

January 2008

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous,
and Ω is any subset of $\mathbb{R}^n$

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous, and Ω is any subset of $\mathbb{R}^n$ and:

- evaluation of f and of the functions defining Ω are usually the result of a computer code (a black box)

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous,
and Ω is any subset of $\mathbb{R}^n$ and:

- evaluation of f and of the functions defining Ω are usually the result of a computer code (a black box)
- the functions are nonsmooth, with some '`if`'s and '`goto`'s

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous,
and Ω is any subset of $\mathbb{R}^n$ and:

- evaluation of f and of the functions defining Ω are usually the result of a computer code (a black box)
- the functions are nonsmooth, with some 'if's and 'goto's
- the functions are expensive black boxes - secs, mins, days

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \text{minimize} \quad f(x)$$
$$\text{subject to} \quad x \in \Omega,$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous,
and Ω is any subset of $\mathbb{R}^n$ and:

- evaluation of f and of the functions defining Ω are usually the result of a computer code (a black box)
- the functions are nonsmooth, with some 'if's and 'goto's
- the functions are expensive black boxes - secs, mins, days
- the functions may fail unexpectedly even for $x \in \Omega$

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \text{minimize} \quad f(x) \qquad \text{subject to} \quad x \in \Omega,$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ may be discontinuous,
and Ω is any subset of $\mathbb{R}^n$ and:

- evaluation of f and of the functions defining Ω are usually the result of a computer code (a black box)
- the functions are nonsmooth, with some 'if's and 'goto's
- the functions are expensive black boxes - secs, mins, days
- the functions may fail unexpectedly even for $x \in \Omega$
- only a few correct digits are ensured

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous,
and Ω is any subset of $\mathbb{R}^n$ and:

- evaluation of f and of the functions defining Ω are usually the result of a computer code (a black box)
- the functions are nonsmooth, with some 'if's and 'goto's
- the functions are expensive black boxes - secs, mins, days
- the functions may fail unexpectedly even for $x \in \Omega$
- only a few correct digits are ensured
- accurate approximation of derivatives is problematic

Avant-propos

My main research interest is nonsmooth optimization:

$$(NLP) \qquad \text{minimize} \quad f(x)$$
$$\text{subject to} \quad x \in \Omega,$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ may be discontinuous,
and Ω is any subset of $\mathbb{R}^n$ and:

- evaluation of f and of the functions defining Ω are usually the result of a computer code (a black box)
- the functions are nonsmooth, with some '`if`'s and '`goto`'s
- the functions are expensive black boxes - secs, mins, days
- the functions may fail unexpectedly even for $x \in \Omega$
- only a few correct digits are ensured
- accurate approximation of derivatives is problematic
- the constraints defining Ω may be nonlinear, nonconvex, nonsmooth and may simply return '`yes/no`'.

Presentation Outline

1. Tsunamy warning buoys

Presentation Outline

1. Tsunamy warning buoys

2. Buoy placement optimization
   - Initiating the collaboration
   - The building blocks of an optimization model
   - Playing with model formulations

Presentation Outline

1. Tsunamy warning buoys

2. Buoy placement optimization
   - Initiating the collaboration
   - The building blocks of an optimization model
   - Playing with model formulations

3. A direct search algorithm
   - The Mesh Adaptive Direct Search algorithm
   - Summary of convergence analysis

Presentation Outline

1. Tsunamy warning buoys
2. Buoy placement optimization
   - Initiating the collaboration
   - The building blocks of an optimization model
   - Playing with model formulations
3. A direct search algorithm
   - The Mesh Adaptive Direct Search algorithm
   - Summary of convergence analysis
4. Conclusions and plans

Presentation Outline

1. Tsunamy warning buoys
2. Buoy placement optimization
   - Initiating the collaboration
   - The building blocks of an optimization model
   - Playing with model formulations
3. A direct search algorithm
   - The Mesh Adaptive Direct Search algorithm
   - Summary of convergence analysis
4. Conclusions and plans

Controlling tsunami risk

- A tsunami is a long wave. The most dangerous are caused by magnitude ≥ 7.5 earthquakes on the ocean floor. There is evidence that underwater landslides and volcanic eruptions have caused tsunamis.

Controlling tsunami risk

- A tsunami is a long wave. The most dangerous are caused by magnitude ≥ 7.5 earthquakes on the ocean floor. There is evidence that underwater landslides and volcanic eruptions have caused tsunamis.
- *Education is important*: A December 04 tsunami in the Indian Ocean killed hundreds of thousands because of a lack of education and a lack of warning.

Controlling tsunami risk

- A tsunami is a long wave. The most dangerous are caused by magnitude ≥ 7.5 earthquakes on the ocean floor. There is evidence that underwater landslides and volcanic eruptions have caused tsunamis.
- *Education is important*: A December 04 tsunami in the Indian Ocean killed hundreds of thousands because of a lack of education and a lack of warning.
- *Detection is important*: A 3 meter tsunami hitting the Los Angeles docks without warning could disrupt the US economy.

Controlling tsunami risk

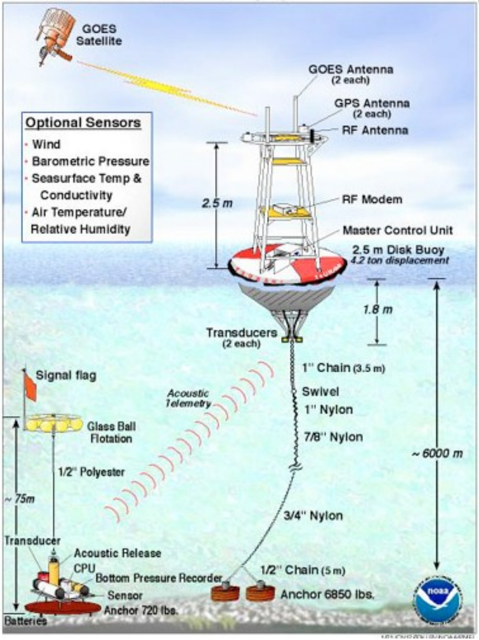
- A tsunami is a long wave. The most dangerous are caused by magnitude ≥ 7.5 earthquakes on the ocean floor. There is evidence that underwater landslides and volcanic eruptions have caused tsunamis.
- *Education is important*: A December 04 tsunami in the Indian Ocean killed hundreds of thousands because of a lack of education and a lack of warning.
- *Detection is important*: A 3 meter tsunami hitting the Los Angeles docks without warning could disrupt the US economy.
- *Accurate prediction is important*: A tsunami was correctly predicted to hit Hawaii in 1994. The total evacuation cost about 60million$US.

Controlling tsunami risk

- A tsunami is a long wave. The most dangerous are caused by magnitude ≥ 7.5 earthquakes on the ocean floor. There is evidence that underwater landslides and volcanic eruptions have caused tsunamis.
- *Education is important*: A December 04 tsunami in the Indian Ocean killed hundreds of thousands because of a lack of education and a lack of warning.
- *Detection is important*: A 3 meter tsunami hitting the Los Angeles docks without warning could disrupt the US economy.
- *Accurate prediction is important*: A tsunami was correctly predicted to hit Hawaii in 1994. The total evacuation cost about 60million$US. The 18inch tsunami arrived at the predicted time and the "I survived the tsunami" T-shirts went on sale at Hilo Hattie's soon after.

DART mooring system

- *Deep ocean Assessment and Reporting of Tsunamis* (DART) buoys are sensors on the ocean floor with a communication connection to a surface buoy. The tsunami amplitude they detect feeds prediction.
- DART buoys cost about 250,000$US + the cost of deployment and maintenance.



Tsunami reporting responsibility within NOAA (National Oceanic and Athmospheric Administration)

This is my personal understanding of the NOAA structure: there are surely subtleties I am missing, but for the purposes of this talk

- PMEL (Pacific Marine Environmental Lab) developed the buoys and recommends where they are deployed.
- NDBC (National Data Buoy Center) manufactures, deploys, and maintains the buoys
- PMEL monitors the buoy data and provides forecasts to the National Weather Service (NWS).
- NWS issues warnings and alerts to the public.

Tsunami reporting responsibility within NOAA (National Oceanic and Athmospheric Administration)

This is my personal understanding of the NOAA structure: there are surely subtleties I am missing, but for the purposes of this talk

- PMEL (Pacific Marine Environmental Lab) developed the buoys and recommends where they are deployed.
- NDBC (National Data Buoy Center) manufactures, deploys, and maintains the buoys
- PMEL monitors the buoy data and provides forecasts to the National Weather Service (NWS).
- NWS issues warnings and alerts to the public.

A budget for 35-40 buoys was given to PMEL. They quickly realized that positioning them in the vast Pacific involved optimization, and contacted members of the optimization community.

Presentation Outline

1. Tsunamy warning buoys
2. Buoy placement optimization
   - Initiating the collaboration
   - The building blocks of an optimization model
   - Playing with model formulations
3. A direct search algorithm
   - The Mesh Adaptive Direct Search algorithm
   - Summary of convergence analysis
4. Conclusions and plans

The challenge

Two groups are involved: the optimization group (us), and the PMEL tsunami scientis.

The challenge

Two groups are involved: the optimization group (us), and the PMEL tsunami scientis.

- Initially, the optimization group knows nothing about DART placement.
- Initially, PMEL does not know much about optimization.

The challenge

Two groups are involved: the optimization group (us), and the PMEL tsunami scientis.

- Initially, the optimization group knows nothing about DART placement.
- Initially, PMEL does not know much about optimization.
- Major difficulty: different technical languages
  - How does the underwater landscape affect the detection amplitude of the DART buoy ?
  - What is it that they really wish to optimize ? What are the constraint ? The objective function ?

The challenge

Two groups are involved: the optimization group (us), and the PMEL tsunami scientis.

- Initially, the optimization group knows nothing about DART placement.
- Initially, PMEL does not know much about optimization.
- Major difficulty: different technical languages
  - How does the underwater landscape affect the detection amplitude of the DART buoy ?
  - What is it that they really wish to optimize ? What are the constraint ? The objective function ?

John Dennis spent two months at the PMEL headquarters learning about the problem, and teaching them notions of optimization.

PMEL's perspective

- Numerical optimization is the process of using an algorithm to minimize or maximize a function subject to equality or inequality constraints.

PMEL's perspective

- Numerical optimization is the process of using an algorithm to minimize or maximize a function subject to equality or inequality constraints.
- The idea is to model DART array placement as a numerical optimization problem.

PMEL's perspective

- Numerical optimization is the process of using an algorithm to minimize or maximize a function subject to equality or inequality constraints.
- The idea is to model DART array placement as a numerical optimization problem.
- Optimization modeling requires specifying appropriate decision variables, objective function, and constraints so that the formalism models the real-world problem adequately and provides a solvable problem.

PMEL's perspective

- Numerical optimization is the process of using an algorithm to minimize or maximize a function subject to equality or inequality constraints.
- The idea is to model DART array placement as a numerical optimization problem.
- Optimization modeling requires specifying appropriate decision variables, objective function, and constraints so that the formalism models the real-world problem adequately and provides a solvable problem.
- Modeling is inherently interdisciplinary, and it is not easy.

Optimization format for NOMAD

NOMAD is our derivative-free nonlinear programming algorithm. It has been used successfully on many real-world problems.

Optimization format for NOMAD

NOMAD is our derivative-free nonlinear programming algorithm. It has been used successfully on many real-world problems.

NOMAD wants a problem in the form:

$$\min_{x \in \Omega} \quad f(x)$$

Optimization format for NOMAD

NOMAD is our derivative-free nonlinear programming algorithm. It has been used successfully on many real-world problems.

NOMAD wants a problem in the form:

$$\min_{x \in \Omega} \quad f(x)$$

where $\Omega \equiv \{x \in X : C(x) \leq 0\} \subset \mathbb{R}^n$.
The constraints are partitioned into two groups.

- X contains the *closed* constraints.
- $C(x) \leq 0$ are called the *open* constraints.

Open, closed and hidden constraints

Consider the toy problem:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^2} & x_1^2 - \sqrt{x_2} \\ \text{s.t.} & -x_1^2 + x_2^2 \leq 1 \\ & x_2 \geq 0 \end{array}$$

Open, closed and hidden constraints

Consider the toy problem:

$$\begin{array}{rl} \min\limits_{x \in \mathbb{R}^2} & x_1^2 - \sqrt{x_2} \\ \text{s.t.} & -x_1^2 + x_2^2 \leq 1 \\ & x_2 \geq 0 \end{array}$$

- Closed constraints *must* be satisfied at every trial vector of decision variables in order for the functions to evaluate.

Open, closed and hidden constraints

Consider the toy problem:

$$\begin{array}{rl} \min\limits_{x \in \mathbb{R}^2} & x_1^2 - \sqrt{x_2} \\ \text{s.t.} & -x_1^2 + x_2^2 \leq 1 \\ & x_2 \geq 0 \end{array}$$

- Closed constraints *must* be satisfied at every trial vector of decision variables in order for the functions to evaluate. Here $x_2 \geq 0$ is a closed constraint, because if it is violated, the objective function will fail.

Open, closed and hidden constraints

Consider the toy problem:

$$\begin{array}{rl} \min\limits_{x \in \mathbb{R}^2} & x_1^2 - \sqrt{x_2} \\ \text{s.t.} & -x_1^2 + x_2^2 \leq 1 \\ & x_2 \geq 0 \end{array}$$

- Closed constraints *must* be satisfied at every trial vector of decision variables in order for the functions to evaluate. Here $x_2 \geq 0$ is a closed constraint, because if it is violated, the objective function will fail.
- Open constraints must be satisfied at the solution, but an optimization algorithm may use some trial points that violate it. Here $-x_1^2 + x_2^2 \leq 1$ is an open constraint.

Open, closed and hidden constraints

Consider the toy problem:

$$\begin{array}{rl} \min\limits_{x \in \mathbb{R}^2} & x_1^2 - \ln(x_2) \\ \text{s.t.} & -x_1^2 + x_2^2 \leq 1 \\ & x_2 \geq 0 \end{array}$$

- Closed constraints *must* be satisfied at every trial vector of decision variables in order for the functions to evaluate. Here $x_2 \geq 0$ is a closed constraint, because if it is violated, the objective function will fail.
- Open constraints must be satisfied at the solution, but an optimization algorithm may use some trial points that violate it. Here $-x_1^2 + x_2^2 \leq 1$ is an open constraint.
- Lets change the objective. $x_2 \neq 0$ is now an hidden constraint. f is set to ∞ when $x \in \Omega$ but x fails to satisfy an hidden contraint.

Open, closed and hidden constraints

Consider the toy problem:

$$\begin{array}{rl} \min\limits_{x \in \mathbb{R}^2} & x_1^2 - \ln(x_2) \\ \text{s.t.} & -x_1^2 + x_2^2 \leq 1 \\ & x_2 \geq 0 \end{array}$$

- Closed constraints *must* be satisfied at every trial vector of decision variables in order for the functions to evaluate. Here $x_2 \geq 0$ is a closed constraint, because if it is violated, the objective function will fail.
- Open constraints must be satisfied at the solution, but an optimization algorithm may use some trial points that violate it. Here $-x_1^2 + x_2^2 \leq 1$ is an open constraint.
- Lets change the objective. $x_2 \neq 0$ is now an hidden constraint. f is set to ∞ when $x \in \Omega$ but x fails to satisfy an hidden contraint.

  DART placement has nasty closed and hidden constraints.

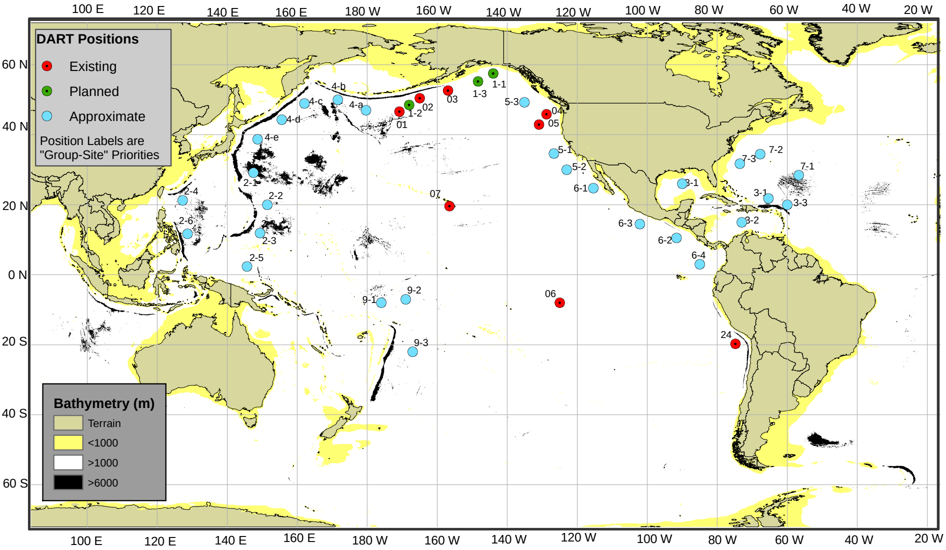
The optimization group's perspective

- The PMEL scientists possess a lot of data on tsunamis but it is not organized in the form of an optimization problem.

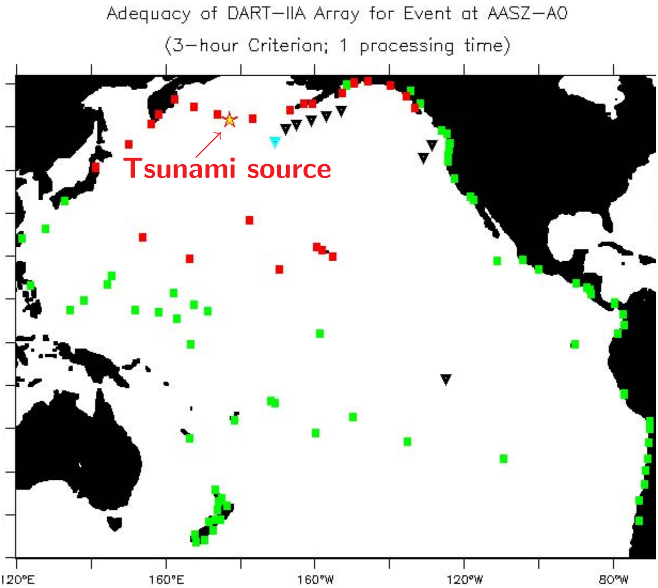
The optimization group's perspective

- The PMEL scientists possess a lot of data on tsunamis but it is not organized in the form of an optimization problem.
- The followings slides represent examples of the raw data.

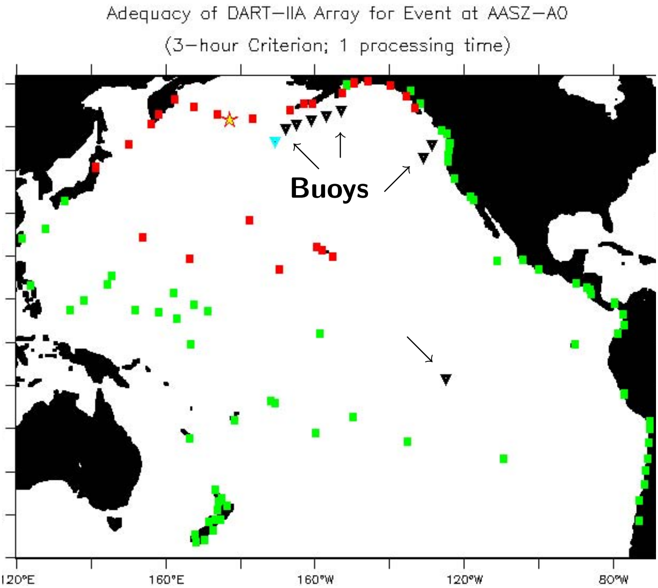
Preliminary placement by a panel of experts



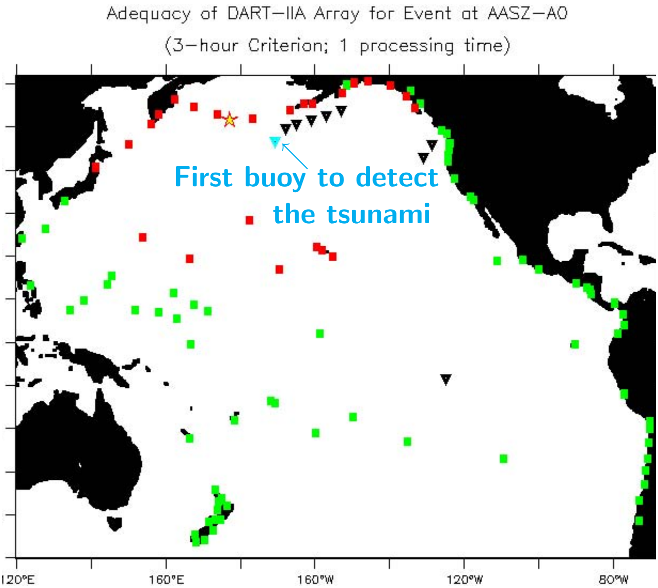
PMEL scientists can forecast arrival time given the source



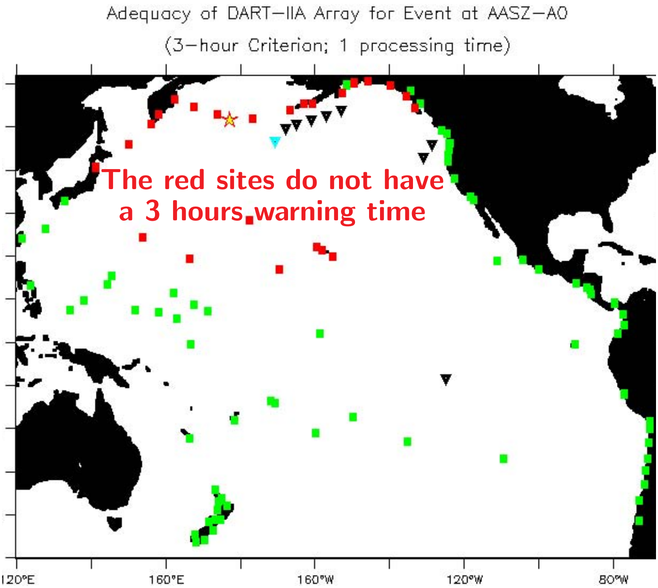
PMEL scientists can forecast arrival time given the source



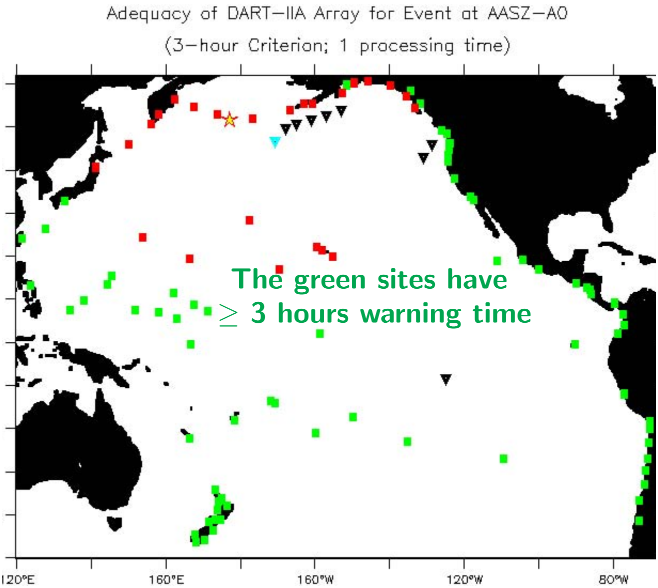
PMEL scientists can forecast arrival time given the source



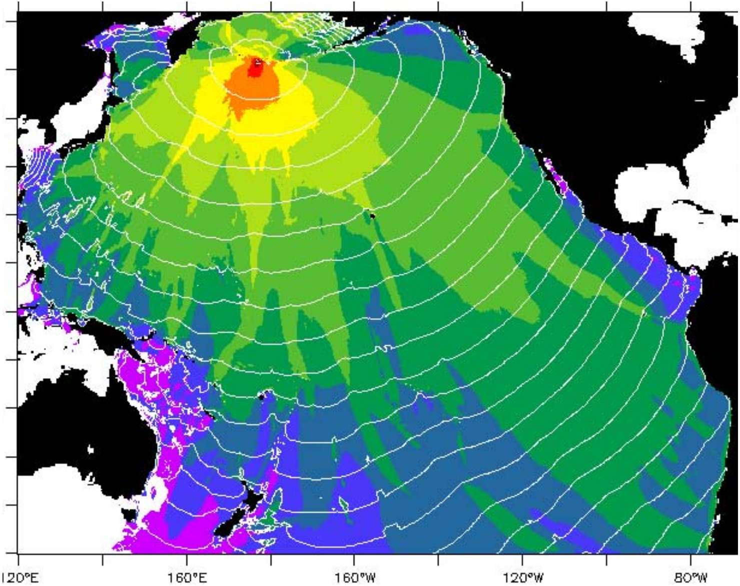
PMEL scientists can forecast arrival time given the source



PMEL scientists can forecast arrival time given the source



PMEL scientists can predict intensity given the source



Level sets of the intensity of a tsunami wave and of travel time.

A non-smooth problem

Source – http://nctr.pmel.noaa.gov/Mov/andr1.mov

Presentation Outline

1. Tsunamy warning buoys
2. Buoy placement optimization
   - Initiating the collaboration
   - The building blocks of an optimization model
   - Playing with model formulations
3. A direct search algorithm
   - The Mesh Adaptive Direct Search algorithm
   - Summary of convergence analysis
4. Conclusions and plans

The building blocks of an optimization model

How can this information be used to construct an optimization model, suitable for our software NOMAD.

The building blocks of an optimization model

How can this information be used to construct an optimization model, suitable for our software NOMAD.

- The preliminary placement can obviously serve as a starting point for our method.
- Travel time of the wave can be turned into a function
- Intensity of the wave can be turned into a function
- Warning time can be turned into a function
- ...

Building blocks (computer codes that return various function values) can be elaborated.

What do PMEL scientists want DART to do?

- Detect tsunamis within 1 hour $\Rightarrow$ put the buoys close to the sources - call this timely detection.

What do PMEL scientists want DART to do?

- Detect tsunamis within 1 hour ⇒ put the buoys close to the sources - call this timely detection.
- Avoid data corruption from earthquake ⇒ but not too close to the source - call this not too close.

What do PMEL scientists want DART to do?

- Detect tsunamis within 1 hour $\Rightarrow$ put the buoys close to the sources - call this timely detection.
- Avoid data corruption from earthquake $\Rightarrow$ but not too close to the source - call this not too close.
- Avoid weak signals $\Rightarrow$ put buoys in the main tsunami beams - call this sufficient detection amplitude.

What do PMEL scientists want DART to do?

- Detect tsunamis within 1 hour $\Rightarrow$ put the buoys close to the sources - call this timely detection.
- Avoid data corruption from earthquake $\Rightarrow$ but not too close to the source - call this not too close.
- Avoid weak signals $\Rightarrow$ put buoys in the main tsunami beams - call this sufficient detection amplitude.
- Avoid unsuitable bottom conditions $\Rightarrow$ weird yes/no optimization constraint - call this bottom conditions.

What do PMEL scientists want DART to do?

- Detect tsunamis within 1 hour $\Rightarrow$ put the buoys close to the sources - call this timely detection.
- Avoid data corruption from earthquake $\Rightarrow$ but not too close to the source - call this not too close.
- Avoid weak signals $\Rightarrow$ put buoys in the main tsunami beams - call this sufficient detection amplitude.
- Avoid unsuitable bottom conditions $\Rightarrow$ weird yes/no optimization constraint - call this bottom conditions.
- Have multiple buoys able to achieve these goals for each source $\Rightarrow$ another strange nondifferentiable optimization constraint - call this sensor coverage.

What do PMEL scientists want DART to do?

- Detect tsunamis within 1 hour $\Rightarrow$ put the buoys close to the sources - call this timely detection.
- Avoid data corruption from earthquake $\Rightarrow$ but not too close to the source - call this not too close.
- Avoid weak signals $\Rightarrow$ put buoys in the main tsunami beams - call this sufficient detection amplitude.
- Avoid unsuitable bottom conditions $\Rightarrow$ weird yes/no optimization constraint - call this bottom conditions.
- Have multiple buoys able to achieve these goals for each source $\Rightarrow$ another strange nondifferentiable optimization constraint - call this sensor coverage.

Given some buoys positions, PMEL produced software that measures these quantities. The CPU time for these computations is of the order of 30 seconds.

Presentation Outline

First problem formulation

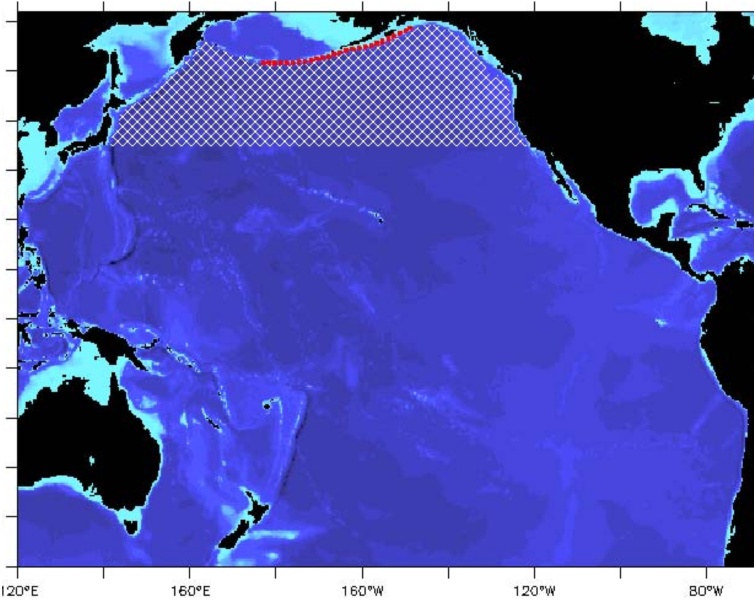
First we tried to optimize:

$\min$ (time to detection)

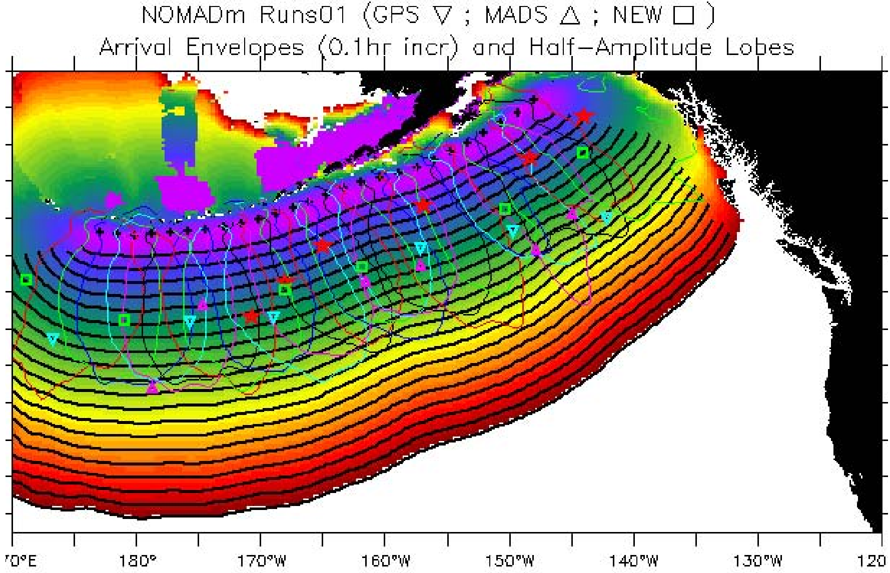
subject to buoy placements that satisfy:

- the closed constraints:
    - bottom conditions
    - not too close
- the open constraints:
    - sufficient detection amplitude
    - sensor coverage

First test problem - the domain



First test problem - NOMADm results



What did we learn from the first test problem ?

- Two NOMAD variants converged in an hour to reasonable solutions for this test problem
- The detection time was adequate (the objective function)
- Unfortunately, the solutions did not satisfy every constraints of the initial model. To satisfy the open sensor coverage constraint, we had to loosen the required tsunami detection amplitude constraints to lower levels

The conclusion of this first model is that we do not have enough buoys to achieve the specified tsunami detection amplitude and sensor coverage constraints for 7.5 earthquakes.

What did we learn from the first test problem ?

- Two NOMAD variants converged in an hour to reasonable solutions for this test problem
- The detection time was adequate (the objective function)
- Unfortunately, the solutions did not satisfy every constraints of the initial model. To satisfy the open sensor coverage constraint, we had to loosen the required tsunami detection amplitude constraints to lower levels

The conclusion of this first model is that we do not have enough buoys to achieve the specified tsunami detection amplitude and sensor coverage constraints for 7.5 earthquakes.
The objective was satisfactory. So tried a second test problem:

Second problem formulation

To nail down how much we miss the data quality requirement we solved:

$\max$ (tsunami detection amplitude) $\Leftarrow$ was a $\geq$ constraint
subject to buoy placements that satisfy:

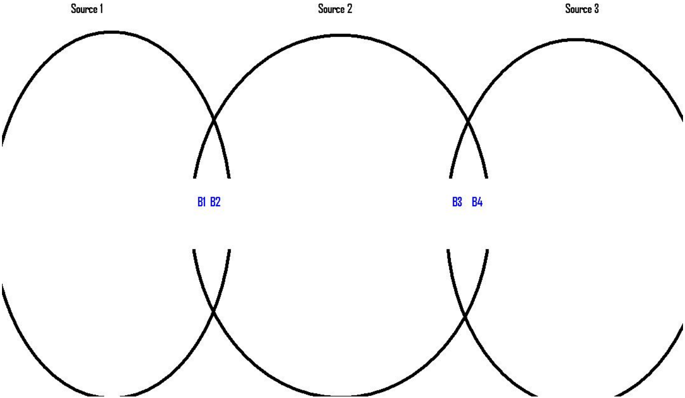
- the closed constraints:
  - bottom conditions
  - not too close
- the open constraints:
  - adequate time to detection $\Leftarrow$ was the objective
  - sensor coverage

What did we learn from the second test problem ?

- Again MADS converged in an hour.
- The buoys found the "sweet spots" in the overlaps of highest amplitude envelopes and paired up there

What did we learn from the second test problem ?

- Again MADS converged in an hour.
- The buoys found the “sweet spots” in the overlaps of highest amplitude envelopes and paired up there



- The extra buoys wandered off in the feasible region, clearly

Collaboration is an iterative process

Coming out with the formulation of the first test problems took several days.
Solving with NOMAD was rapid.

Collaboration is an iterative process

Coming out with the formulation of the first test problems took several days.
Solving with NOMAD was rapid.

The second test problem was generated faster than the first.
Solving with NOMAD was again rapid.

Collaboration is an iterative process

Coming out with the formulation of the first test problems took several days.
Solving with NOMAD was rapid.

The second test problem was generated faster than the first.
Solving with NOMAD was again rapid.

In summary, NOMAD is used as a tool by the decision makers.
The solutions provided by NOMAD allow the user to refine the model, and his interpretation of objectives and constraints.

Presentation Outline

The Mesh Adaptive Direct Search algorithm

MADS is a derivative-free, direct search class of methods that targets problems of the form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

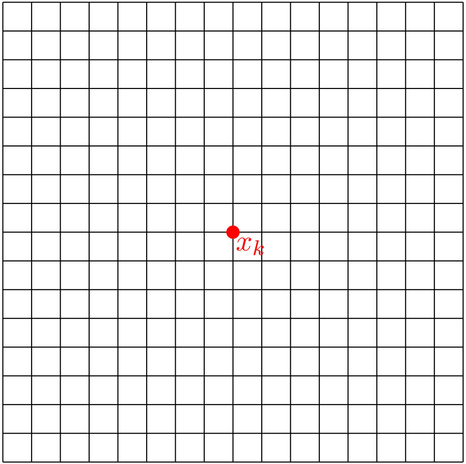
The Mesh Adaptive Direct Search algorithm

MADS is a derivative-free, direct search class of methods that targets problems of the form:

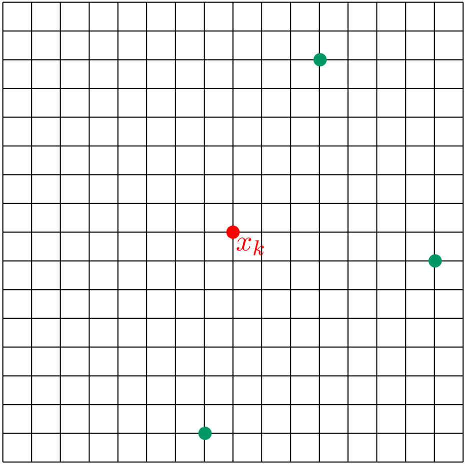
$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array}$$

The optimality conditions that MADS guarantees on $\hat{x}$ are 'proportional' to the smoothness of f and Ω.

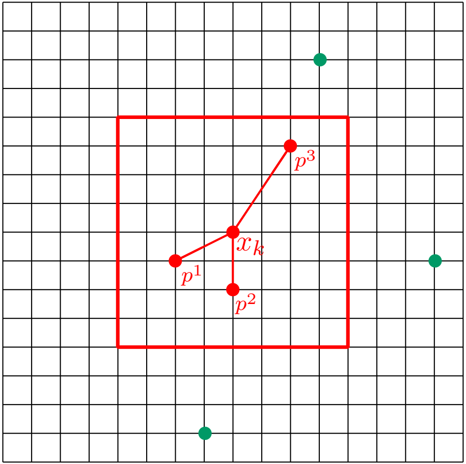
A MADS iteration

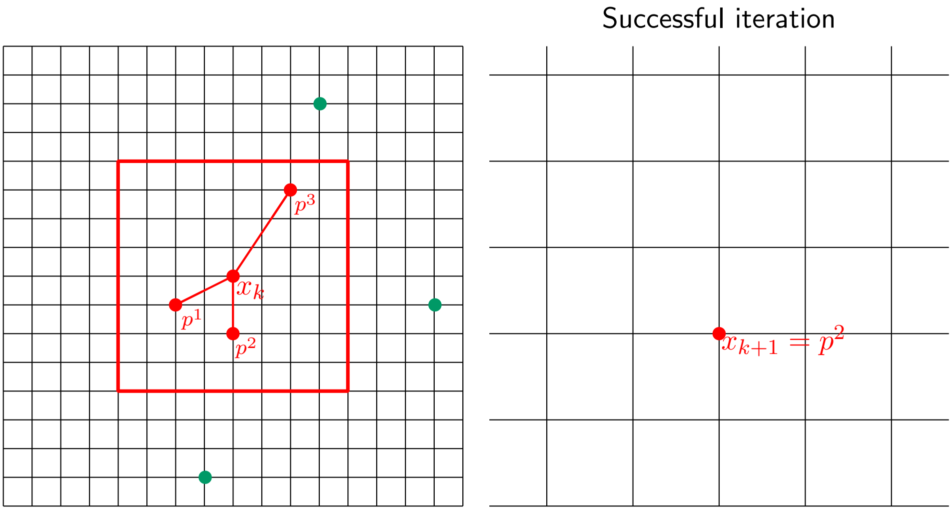


A MADS iteration

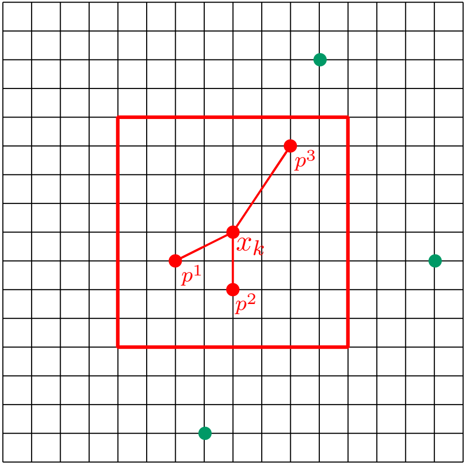


A MADS iteration

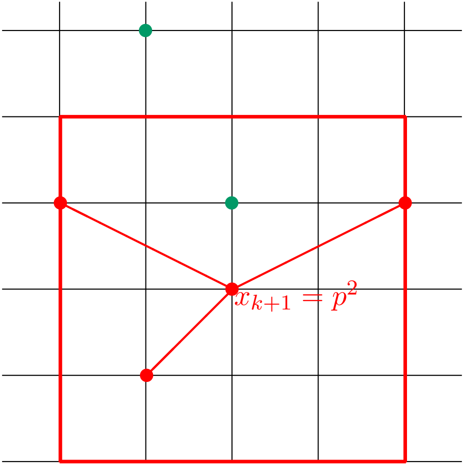



Successful iteration
p^3
x_k
p^1
p^2
x_{k+1} = p^2

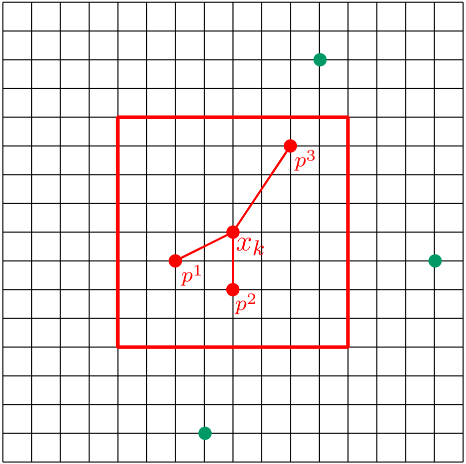
A MADS iteration



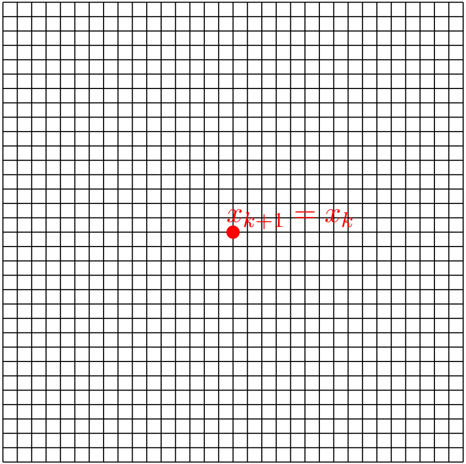
Successful iteration



A MADS iteration

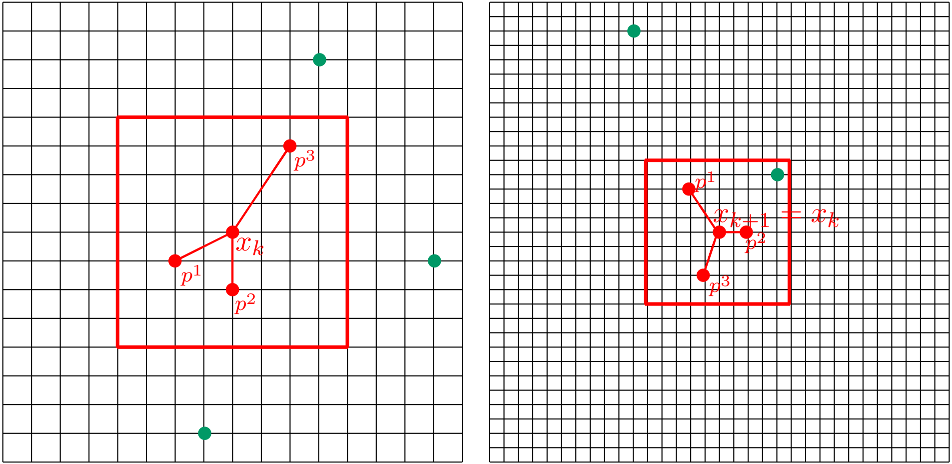


unsuccessful iteration



A MADS iteration

unsuccessful iteration



Barrier approach to closed constraints

To enforce X constraints, replace f by a barrier objective

$$f_X(x) := \begin{cases} f(x) & \text{if } x \in X, \\ +\infty & \text{otherwise.} \end{cases}$$

Then apply the unconstrained algorithm to f_X.

Barrier approach to closed constraints

To enforce X constraints, replace f by a barrier objective

$$f_X(x) := \begin{cases} f(x) & \text{if } x \in X, \\ +\infty & \text{otherwise.} \end{cases}$$

Then apply the unconstrained algorithm to f_X.
Remarks :

- The quality of the limit solution depends the local smoothness of f, not of f_X.
- This approach can handle strict inequalities.
- Expensive evaluations of f are saved when x is found to be infeasible.

Filter approach to open constraints

Define the nonnegative constraint violation function

$$h(x) \ := \ \sum_j \max(0, c_j(x))^2$$

Remarks :

- $h(x) = 0$ if and only if all open constraints are satisfied.
- Accept a new trial points if it is feasible and improves f or if it is infeasible but improves h.

Presentation Outline

Hierarchy of MADS convergence results

- Regardless one the smoothness of the function, there exists a convergent subsequence of mesh local optimizers $x_k \longrightarrow \hat{x}$ on meshes that get infinitely fine.

Hierarchy of MADS convergence results

- Regardless one the smoothness of the function, there exists a convergent subsequence of mesh local optimizers $x_k \to \hat{x}$ on meshes that get infinitely fine.
- If f is Lipschitz near any such limit $\hat{x}$ and if $T_\Omega^H(\hat{x}) \neq \emptyset$, then with probability 1, $\hat{x}$ is a Clarke stationary point of f over Ω:
$$f^\circ(\hat{x}; v) \geq 0, \forall v \in T_\Omega^{Cl}(\hat{x}).$$

Hierarchy of MADS convergence results

- Regardless one the smoothness of the function, there exists a convergent subsequence of mesh local optimizers $x_k \longrightarrow \hat{x}$ on meshes that get infinitely fine.
- If f is Lipschitz near any such limit $\hat{x}$ and if $T_\Omega^H(\hat{x}) \neq \emptyset$, then with probability 1, $\hat{x}$ is a Clarke stationary point of f over Ω: $f^\circ(\hat{x}; v) \geq 0, \forall v \in T_\Omega^{Cl}(\hat{x})$.
- Furthermore, if f is strictly differentiable at $\hat{x}$ and if Ω is regular at $\hat{x}$, then with probability 1, $\hat{x}$ is a contingent KKT stationary point of f over Ω: $\nabla f(\hat{x})^T v \geq 0, \forall v \in T_\Omega^{Co}(\hat{x})$,

Hierarchy of MADS convergence results

- Regardless one the smoothness of the function, there exists a convergent subsequence of mesh local optimizers $x_k \to \hat{x}$ on meshes that get infinitely fine.
- If f is Lipschitz near any such limit $\hat{x}$ and if $T_\Omega^H(\hat{x}) \neq \emptyset$, then with probability 1, $\hat{x}$ is a Clarke stationary point of f over Ω:
$$f^\circ(\hat{x}; v) \geq 0, \forall v \in T_\Omega^{Cl}(\hat{x}).$$
- Furthermore, if f is strictly differentiable at $\hat{x}$ and if Ω is regular at $\hat{x}$, then with probability 1, $\hat{x}$ is a contingent KKT stationary point of f over Ω: $\nabla f(\hat{x})^T v \geq 0, \forall v \in T_\Omega^{Co}(\hat{x}),$ where $T_\Omega^{Co}(\hat{x})$ is the contingent cone to Ω at x.
- Furthermore, if f is twice strictly differentiable at $\hat{x}$ and $\nabla^2 f(\hat{x})$ is non-singular, and if Ω locally convex near $\hat{x}$, then with probability 1, $\hat{x}$ is local minimizer of f over Ω:
$$\exists \epsilon > 0 \text{ such that } f(\hat{x}) \leq f(y), \forall y \in \Omega \cap B_\epsilon(\hat{x}).$$

Presentation Outline

1. Tsunamy warning buoys

2. Buoy placement optimization
   - Initiating the collaboration
   - The building blocks of an optimization model
   - Playing with model formulations

3. A direct search algorithm
   - The Mesh Adaptive Direct Search algorithm
   - Summary of convergence analysis

4. Conclusions and plans

Conclusions

- NOMADm solved several tweaks of the first two test problem easily and quickly.
- Collaboration is an iterative process.

```
0-     Learn each other's language
1-     Build an initial model
2-     REPEAT
3-         Solve the model
4-         Interpret the results
5-         Adapt, adjust and correct the model
6-     UNTIL a satisfactory solution is found.
```

Conclusions

- NOMADm solved several tweaks of the first two test problem easily and quickly.
- Collaboration is an iterative process.

```
0-     Learn each other's language
1-     Build an initial model
2-     REPEAT
3-         Solve the model
4-         Interpret the results
5-         Adapt, adjust and correct the model
6-     UNTIL a satisfactory solution is found.
```

- Step 0 is hard. But once it is done, things progress rapidly.
- Collaboration between both groups is essential in steps 0,1,4 and 5.
  PMEL is the judge for step 6.
  Our optimization team handles step 3 using NOMAD.

Plans

- Continue to work with NOAA/PMEL tsunami experts to refine the formulation to get answers they like.
- Tailor the underlying MADS algorithm to algorithms with this block structure - this should have a great payoff for a whole class of similar sensor location problems.

Plans

- Continue to work with NOAA/PMEL tsunami experts to refine the formulation to get answers they like.
- Tailor the underlying MADS algorithm to algorithms with this block structure - this should have a great payoff for a whole class of similar sensor location problems.
- Publicity: session WA9 at 10h30 has talks that discuss MADS.
- NOMAD is Gilles Couture's C++ industrial strength implementation, freely available at `www.gerad.ca/NOMAD`
- NOMADm is Mark Abramson's MATLAB implementation freely available at `www.afit.edu/en/enc/Faculty/MAbramson/nomadm.html`
- MADS is in the GADS MATHWORKS MATLAB toolbox.

Plans

- Continue to work with NOAA/PMEL tsunami experts to refine the formulation to get answers they like.
- Tailor the underlying MADS algorithm to algorithms with this block structure - this should have a great payoff for a whole class of similar sensor location problems.
- Publicity: session WA9 at 10h30 has talks that discuss MADS.
- NOMAD is Gilles Couture's C++ industrial strength implementation, freely available at `www.gerad.ca/NOMAD`
- NOMADm is Mark Abramson's MATLAB implementation freely available at `www.afit.edu/en/enc/Faculty/MAbramson/nomadm.html`
- MADS is in the GADS MATHWORKS MATLAB toolbox.

Thank you for your attention.