Pattern Search for Mixed Variable Optimization Problems

Mark A. Abramson (Air Force Institute of Technology)
Charles Audet (École Polytechnique de Montréal)
John Dennis, Jr. (Rice University)

The views expressed here are those of the authors and do not reflect the official policy or position of the USAF, Department of Defense, US government, or our other sponsors:

Boeing, ExxonMobil, CSRI, LASCi, NSF.
### Pattern Search for Mixed Variable Optimization Problems

**Title and Subtitle**

Pattern Search for Mixed Variable Optimization Problems

**Performing Organization**

Air Force Institute of Technology, Wright Patterson AFB, OH, 45433

**Abstract**

Approved for public release; distribution unlimited

**DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution unlimited

**Security Classification of:***

<table>
<thead>
<tr>
<th>a. Report</th>
<th>b. Abstract</th>
<th>c. This Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>unclassified</td>
<td>unclassified</td>
<td>unclassified</td>
</tr>
</tbody>
</table>

**Limitation of ABSTRACT**

Same as Report (SAR)

**Number of Pages**

96
Outline

- Mixed variable problem motivation and formulation
Outline

- Mixed variable problem motivation and formulation
- GPS for linearly constrained MVP problems
Outline

- Mixed variable problem motivation and formulation
- GPS for linearly constrained MVP problems
- Filter GPS for general constrained MVP problems
Outline

- Mixed variable problem motivation and formulation
- GPS for linearly constrained MVP problems
- Filter GPS for general constrained MVP problems
- Results for thermal insulation system design
Heat intercept insulation system

\[ T_{n+1} = T_H \]

\[ T_{i+1} = T_i \]

\[ T_{i-1} = T_0 = T_C \]
Heat intercept insulation system

\[ T_{n+1} = T_H \]
\[ T_{i+1} \]
\[ T_i \]
\[ T_{i-1} \]
\[ T_0 = T_C \]

\[ \text{min } \text{power}(n, I, x, T) \]
Heat intercept insulation system

\[
\begin{align*}
\bar{T}_{n+1} &= \bar{T}_H \\
\bar{T}_{i+1} &= \bar{T}_i \\
\bar{T}_{i-1} &= \bar{T}_i \\
\bar{T}_0 &= \bar{T}_C
\end{align*}
\]

\[
\min \: \text{power}(n, I, x, \bar{T})
\]

subject to

\[
\begin{align*}
n &\in \{1, 2, \ldots, n_{\max}\}, \quad I \in \mathcal{I}^{n+1} \\
\bar{T}_{i-1} &\leq \bar{T}_i \leq \bar{T}_{i+1}, \quad i = 1, 2, \ldots, n \\
\sum_{i=1}^{n+1} x_i &= L, \quad x_i \geq 0, \quad i = 1, 2, \ldots, n + 1
\end{align*}
\]
The general MVP problem

\[ \min_{x \in X} f(x) \]

subject to \( C(x) \leq 0 \),
The general MVP problem

\[ \min_{x \in X} f(x) \]

subject to \( C(x) \leq 0, \)

where

- \( x = (x^c, x^d) \in \mathbb{R}^{n^c} \times \mathbb{Z}^{n^d} \)
- \( X = X^c \times X^d, \) where
  \[ X^c(x^d) = \{ x^c \in \mathbb{R}^{n^c} : \ell(x^d) \leq A(x^d)x^c \leq u(x^d) \} \]
The general MVP problem

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{subject to} & \quad C(x) \leq 0,
\end{align*}
\]

where

- \(x = (x^c, x^d) \in \mathbb{R}^{n^c} \times \mathbb{Z}^{n^d}\)
- \(X = X^c \times X^d\), where
  \[
  X^c(x^d) = \{ x^c \in \mathbb{R}^{n^c} : \ell(x^d) \leq A(x^d)x^c \leq u(x^d) \}
  \]
- \(f\) and \(C = (c_1, c_2, \ldots, c_p)\) may be discontinuous, extended valued, costly
The general MVP problem

\[ \min_{x \in X} f(x) \]
subject to \( C(x) \leq 0, \)

where

- \( x = (x^c, x^d) \in \mathbb{R}^{n_c} \times \mathbb{Z}^{n_d} \)
- \( X = X^c \times X^d, \) where
  \[ X^c(x^d) = \{ x^c \in \mathbb{R}^{n_c} : \ell(x^d) \leq A(x^d)x^c \leq u(x^d) \} \]
- \( f \) and \( C = (c_1, c_2, \ldots, c_p) \) may be discontinuous, extended valued, costly
Choices other than pattern search

Some methods that come to mind are:

- **MINLP methods**: cannot handle categorical variables
Choices other than pattern search

Some methods that come to mind are:

- **MINLP methods**: cannot handle categorical variables
- **Search heuristics**: huge numbers of evaluations and very limited convergence theory
Choices other than pattern search

Some methods that come to mind are:

- **MINLP methods**: cannot handle categorical variables
- **Search heuristics**: huge numbers of evaluations and very limited convergence theory
  - Simulated annealing
Choices other than pattern search

Some methods that come to mind are:

- **MINLP methods**: cannot handle categorical variables
- **Search heuristics**: huge numbers of evaluations and very limited convergence theory
  - Simulated annealing
  - Tabu search
Choices other than pattern search

Some methods that come to mind are:

- **MINLP methods**: cannot handle categorical variables
- **Search heuristics**: huge numbers of evaluations and very limited convergence theory
  - Simulated annealing
  - Tabu search
  - Evolutionary algorithms
Choices other than pattern search

Some methods that come to mind are:

- **MINLP methods**: cannot handle categorical variables
- **Search heuristics**: huge numbers of evaluations and very limited convergence theory
  - Simulated annealing
  - Tabu search
  - Evolutionary algorithms
- **Other methods**: SQP/direct search with 1 categorical variable
INITIALIZATION of directions and step size

For $k = 1, 2, \ldots$

- SEARCH a finite set of mesh points

End
GENERALIZED PATTERN SEARCHES

INITIALIZATION of directions and step size

For \( k = 1, 2, \ldots \)

- SEARCH a finite set of mesh points
- POLL neighboring mesh points

End
INITIALIZATION of directions and step size

For $k = 1, 2, \ldots$

- SEARCH a finite set of mesh points
- POLL neighboring mesh points
- UPDATE parameters:

End
INITIALIZATION of directions and step size

For $k = 1, 2, \ldots$

- SEARCH a finite set of mesh points
- POLL neighboring mesh points
- UPDATE parameters:
  - Success: Accept new iterate
INITIALIZATION of directions and step size

For $k = 1, 2, \ldots$

- SEARCH a finite set of mesh points
- POLL neighboring mesh points
- UPDATE parameters:
  - Success: Accept new iterate
  - Failure: Refine mesh

End
Details of $k^{th}$ POLL step

Mesh:  \[ M_k = \{ p_k + \Delta_k D z : z \in \mathbb{Z}^{|D|}_+ \} , \]

Poll set:  \[ P_k = \{ p_k + \Delta_k d : d \in D_k \subseteq D \} , \]
Details of $k$th POLL step

Mesh: $M_k = \{p_k + \Delta_k D_z : z \in \mathbb{Z}_+^{|D|}\}$,
Poll set: $P_k = \{p_k + \Delta_k d : d \in D_k \subseteq D\}$,

where
- $p_k$ is the current poll center
- $\Delta_k > 0$ is the mesh size parameter
- $D_k, D$ are positive spanning sets
Details of $k$th POLL step

Mesh: $M_k = \{ p_k + \Delta_k D z : z \in \mathbb{Z}_+^{|D|} \}$,
Poll set: $P_k = \{ p_k + \Delta_k d : d \in D_k \subset D \}$,

where
- $p_k$ is the current poll center
- $\Delta_k > 0$ is the mesh size parameter
- $D_k, D$ are positive spanning sets

Examples: $D = [I, -I]$ $D = [I, -e]$

One of these directions should be a descent direction.

Derivative information can reduce poll set to a singleton.
Details of $k^{th}$ POLL step

**Mesh:** $M_k = \{p_k + \Delta_k Dz : z \in \mathbb{Z}^{|D|}_+ \}$,

**Poll set:** $P_k = \{p_k + \Delta_k d : d \in D_k \subseteq D\}$,

where
- $p_k$ is the current poll center
- $\Delta_k > 0$ is the mesh size parameter
- $D_k, D$ are positive spanning sets

**Examples:**
- $D = [I, -I]$  
- $D = [I, -e]$

One of these directions should be a descent direction.

Derivative information can reduce poll set to a singleton.
Definition of local optimality

\[ x = (x^c, x^d) \in X \text{ is a } \textit{local minimizer of } f \text{ with respect to neighbors } \mathcal{N}(x) \subset X \text{ if } \exists \epsilon > 0 \text{ such that } f(x) \leq f(v) \]

\[ \forall v \in X \cap \bigcup_{y \in \mathcal{N}(x)} \left( B(y^c, \epsilon) \times y^d \right). \]
Definition of local optimality

$x = (x^c, x^d) \in X$ is a local minimizer of $f$ with respect to neighbors $\mathcal{N}(x) \subset X$ if $\exists \epsilon > 0$ such that $f(x) \leq f(v)$ for all $v \in X \cap \bigcup_{y \in \mathcal{N}(x)} (B(y^c, \epsilon) \times y^d)$. 

Pattern Search for Mixed Variable Optimization Problems – p.8/35
Definition of local optimality

\[ x = (x^c, x^d) \in X \text{ is a local minimizer of } f \text{ with respect to neighbors } \mathcal{N}(x) \subset X \text{ if } \exists \epsilon > 0 \text{ such that } f(x) \leq f(v) \]

\[ \forall v \in X \cap \bigcup_{y \in \mathcal{N}(x)} (B(y^c, \epsilon) \times y^d). \]
Definition of local optimality

\[ x = (x^c, x^d) \in X \text{ is a local minimizer of } f \text{ with respect to neighbors } N(x) \subset X \text{ if } \exists \epsilon > 0 \text{ such that } f(x) \leq f(v) \]

\[ \forall v \in X \cap \bigcup_{y \in N(x)} (B(y^c, \epsilon) \times y^d). \]
Definition of local optimality

\[ x = (x^c, x^d) \in X \] is a local minimizer of \( f \) with respect to neighbors \( \mathcal{N}(x) \subset X \) if \( \exists \epsilon > 0 \) such that \( f(x) \leq f(v) \)

\[ \forall v \in X \cap \bigcup_{y \in \mathcal{N}(x)} (B(y^c, \epsilon) \times y^d) . \]
Heatshield discrete neighbors

- Replace any single insulator with a different type
Heatshield discrete neighbors

- Replace any single insulator with a different type.
- Remove any intercept with its left insulator, and increase the thickness of its right insulator to absorb the deficit.

Pattern Search for Mixed Variable Optimization Problems – p.9/35
Heatshield discrete neighbors

- Replace any single insulator with a different type
- Remove any intercept with its left insulator, and increase the thickness of its right insulator to absorb the deficit
- Add an intercept at any position:
Heatshield discrete neighbors

- Replace any single insulator with a different type
- Remove any intercept with its left insulator, and increase the thickness of its right insulator to absorb the deficit
- Add an intercept at any position:
  - The existing insulator is divided (rounded to the mesh)
Heatshield discrete neighbors

- Replace any single insulator with a different type
- Remove any intercept with its left insulator, and increase the thickness of its right insulator to absorb the deficit
- Add an intercept at any position:
  - The existing insulator is divided (rounded to the mesh)
  - The cooling temperature is set to the average of the two intercepts adjacent to it (rounded to the mesh)
Construction of the poll set

\[ P_k = \{ u, v, w \} \]
Construction of the poll set

\[ P_k = \{u, v, w\} \]
\[ \mathcal{N}(x_k) = \{x_k, y^0_1, y^0_2\} \]

\[ y^0_1 \in \mathcal{N}(x_k) \text{ satisfies} \]
\[ f(x_k) < f(y^0_1) < f(x_k) + \xi \]
Construction of the poll set

\[ P_k = \{u, v, w\} \]
\[ \mathcal{N}(x_k) = \{x_k, y_0^1, y_0^2\} \]
\[ x_k = \{y_1^1\} \cup \{a, b, c\} \]

\[ y_1^0 \in \mathcal{N}(x_k) \text{ satisfies} \]
\[ f(x_k) < f(y_1^0) < f(x_k) + \xi \]

Poll Set: \( P_k \cup \mathcal{N}(x_k) \cup x_k \)
Filter GPS for nonlinear constraints

\[ f(x) = \| C(x) \|_2^2 \]

Poll center is either best feasible point or least infeasible point. For each trial point \( x \), \( h(x) \) and \( f(x) \) are plotted on the bi-loss map. If \( x \) is unchanged, it is added to the filter; otherwise, the mesh is refined.
Filter GPS for nonlinear constraints

Poll center is either best feasible point or least infeasible point.

\[ h(x) = \| C(x)_+ \|^2 \]
Filter GPS for nonlinear constraints

Poll center is either best feasible point or least infeasible point.

For each trial point \( x \), \( h(x) \) and \( f(x) \) are plotted on the bi-loss map.

\[
h(x) = \|C(x)_+\|^2
\]
Filter GPS for nonlinear constraints

$\begin{align*}
  f(x) &= h(x) = \|C(x)_+\|^2 \\
  F_k &= (h_k^I, f_k^I) \\
  \overline{F_k} \\
\end{align*}$

Poll center is either best feasible point or least infeasible point.

For each trial point $x$, $h(x)$ and $f(x)$ are plotted on the bi-loss map.

If $x$ is unfiltered, it is added to the filter; otherwise, the mesh is refined.
Construction of the poll set

\[ P_k = \{u, v, w\} \]
\[ \mathcal{N}(p_k) = \{p_k, y_1^0, y_2^0\} \]
\[ \mathcal{X}_k = \{y_1^1\} \cup \{a, b, c\} \]

\( y_1^0 \in \mathcal{N}(p_k) \) satisfies the extended poll criteria

Poll Set: \( P_k \cup \mathcal{N}(p_k) \cup \mathcal{X}_k \)
Local filter for extended polling

$f_k^F + \xi_k^F - f_k^F$

* $(h_k^L, f_k^I)$

$\mathcal{F}_k$

$h_k^L + \xi_k^h$

$h_{\text{max}}$

Main Filter
Local filter for extended polling

\[ f_k^F + \xi_k^F \]

\[ \overline{F}_k \]

Main Filter

Local Filter

Pattern Search for Mixed Variable Optimization Problems – p.16/35
INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do
- Update poll center $p_k$ and extended poll triggers $\xi^f_k, \xi^h_k$. 
INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi_k^f, \xi_k^h$
- Compute incumbent values $f_k^F, f_k^I, h_k^I$
INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi^f_k, \xi^h_k$.
- Compute incumbent values $f^F_k, f^I_k, h^I_k$.
- While trial points are filtered do:
INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi^F_k, \xi^h_k$
- Compute incumbent values $f^F_k, f^I_k, h^I_k$
- While trial points are filtered do:
  - SEARCH: Any finite strategy on the mesh
INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi_k^f, \xi_k^h$
- Compute incumbent values $f_k^F, f_k^I, h_k^I$
- While trial points are filtered do:
  - SEARCH: Any finite strategy on the mesh
  - POLL: Evaluate points in $P_k \cup N(p_k)$
Filter GPS algorithm for MVP

INITIALIZE: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi^f_k, \xi^h_k$
- Compute incumbent values $f^F_k, f^I_k, h^I_k$
- While trial points are filtered do:
  - SEARCH: Any finite strategy on the mesh
  - POLL: Evaluate points in $P_k \cup N(p_k)$
  - EXTENDED POLL: Evaluate points in $X_k(\xi^f_k, \xi^h_k)$
Filter GPS algorithm for MVP

INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter

For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi^f_k, \xi^h_k$
- Compute incumbent values $f^F_k, f^I_k, h^I_k$
- While trial points are filtered do:
  - SEARCH: Any finite strategy on the mesh
  - POLL: Evaluate points in $P_k \cup \mathcal{N}(p_k)$
  - EXTENDED POLL: Evaluate points in $\mathcal{X}_k(\xi^f_k, \xi^h_k)$
- Update:
Filter GPS algorithm for MVP

INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi^f_k, \xi^h_k$
- Compute incumbent values $f^F_k, f^I_k, h^I_k$
- While trial points are filtered do:
  - SEARCH: Any finite strategy on the mesh
  - POLL: Evaluate points in $P_k \cup \mathcal{N}(p_k)$
  - EXTENDED POLL: Evaluate points in $\mathcal{X}_k(\xi^f_k, \xi^h_k)$
- Update:
  - If (found), set $\Delta_{k+1} \geq \Delta_k$ and update filter
INITIALIZATION: Set $\Delta_0, \xi > 0$, and populate filter
For $k = 1, 2, \ldots$, do

- Update poll center $p_k$ and extended poll triggers $\xi^f_k, \xi^h_k$
- Compute incumbent values $f^F_k, f^I_k, h^I_k$
- While trial points are filtered do:
  - SEARCH: Any finite strategy on the mesh
  - POLL: Evaluate points in $P_k \cup N(p_k)$
  - EXTENDED POLL: Evaluate points in $X_k(\xi^f_k, \xi^h_k)$
- Update:
  - If (found), set $\Delta_{k+1} \geq \Delta_k$ and update filter
  - If (not found), set $\Delta_{k+1} < \Delta_k$
Convergence theory assumptions

- All iterates lie in a compact set
Convergence theory assumptions

- All iterates lie in a compact set
- The linear constraint matrix $A$ is rational
Convergence theory assumptions

- All iterates lie in a compact set
- The linear constraint matrix $A$ is rational
- The mesh directions conform to the geometry of $X_c$
Convergence theory assumptions

- All iterates lie in a compact set
- The linear constraint matrix $A$ is rational
- The mesh directions conform to the geometry of $X^c$
- All discrete neighbors lie on the current mesh
Convergence theory assumptions

- All iterates lie in a compact set
- The linear constraint matrix $A$ is rational
- The mesh directions conform to the geometry of $X^c$
- All discrete neighbors lie on the current mesh
- The set-valued neighborhood function $N : X \rightarrow 2^X$ satisfies a notion of continuity.
There exists a subsequence $K$ such that $\lim_{k \in K} \Delta_k = 0$, with limit points:

1. $\hat{p} = \lim_{k \in K} p_k$, where $p_k \in \{p_k^F, p_k^I\}$
2. $\hat{y} = \lim_{k \in K} y_k$, where $y_k \in \mathcal{N}(p_k)$ and $\hat{y} \in \mathcal{N}(\hat{p})$.
3. $\hat{z} = \lim_{k \in K} z_k$, where $z_k$ are extended poll endpoints.

Pattern Search for Mixed Variable Optimization Problems – p.19/35
\[ \exists \text{ subsequence } K \text{ such that } \lim_{k \in K} \Delta_k = 0, \text{ with limit points:} \]

1. \[ \hat{p} = \lim_{k \in K} p_k, \quad \text{where } p_k \in \{p^F_k, p^I_k\} \]
2. \[ \hat{y} = \lim_{k \in K} y_k, \quad \text{where } y_k \in \mathcal{N}(p_k) \text{ and } \hat{y} \in \mathcal{N}(\hat{p}). \]
3. \[ \hat{z} = \lim_{k \in K} z_k, \quad \text{where } z_k \text{ are extended poll endpoints.} \]
Limit points of the algorithm

\[ \exists \text{ subsequence } K \text{ such that } \lim_{k \in K} \Delta_k = 0, \text{ with limit points:} \]

1. \( \hat{p} = \lim_{k \in K} p_k, \) where \( p_k \in \{ p_k^F, p_k^I \} \)

2. \( \hat{y} = \lim_{k \in K} y_k, \) where \( y_k \in \mathcal{N}(p_k) \text{ and } \hat{y} \in \mathcal{N}(\hat{p}). \)

3. \( \hat{z} = \lim_{k \in K} z_k, \) where \( z_k \) are extended poll endpoints.
There exists a subsequence $K$ such that $\lim_{k \in K} \Delta_k = 0$, with limit points:

1. $\hat{p} = \lim_{k \in K} p_k$, where $p_k \in \{p^F_k, p^I_k\}$
2. $\hat{y} = \lim_{k \in K} y_k$, where $y_k \in \mathcal{N}(p_k)$ and $\hat{y} \in \mathcal{N}(\hat{p})$.
3. $\hat{z} = \lim_{k \in K} z_k$, where $z_k$ are extended poll endpoints.
∃ subsequence $K$ such that $\lim_{k \in K} \Delta_k = 0$, with limit points:

1. $\hat{p} = \lim_{k \in K} p_k$, where $p_k \in \{p^{F}_k, p^{I}_k\}$

2. $\hat{y} = \lim_{k \in K} y_k$, where $y_k \in \mathcal{N}(p_k)$ and $\hat{y} \in \mathcal{N}(\hat{p})$

3. $\hat{z} = \lim_{k \in K} z_k$, where $z_k$ are extended poll endpoints
Filter convergence results

Let $D(\hat{p})$ be the set of polling directions used i.o.

- $h$ continuous* at $\hat{p}$ and $\hat{y}$ $\Rightarrow$ $h(\hat{p}) \leq h(\hat{y})$
- $f$ continuous* at $\hat{p}$ and $\hat{y}$ and $p_k = p_k^F$ i. o.
  $\Rightarrow$ $f(\hat{p}) \leq f(\hat{y})$
- $h$ Lipschitz* near $\hat{p}$ $\Rightarrow$ $h^\circ(\hat{p}; (d, 0)) \geq 0 \ \forall d \in D(\hat{p})$
- $f$ Lipschitz* near $\hat{p}$ and $p_k = p_k^F$ i. o.
  $\Rightarrow f^\circ(\hat{p}; (d, 0)) \geq 0 \ \forall d \in D(\hat{p})$
- $h$ strictly differentiable* at $\hat{p}$ and $\Rightarrow \nabla h(\hat{p}) = 0$

Similar results hold for certain $\hat{z}$
Filter convergence results

\( f \) strictly differentiable* at \( \hat{p} \) and \( p_k = p_k^F \) i. o.
\[ \Rightarrow -\nabla^c f(\hat{p}) \in C_d^\circ. \] Similar results hold for certain \( \hat{z} \)

\[ C_d^\circ \]

\[ C_d \]

\[ -\nabla f \]

* with respect to the continuous variables
Filter convergence results

\[ f \text{ strictly differentiable}^* \text{ at } \hat{p} \text{ and } p_k = p_k^F \text{ i. o.} \]
\[ \Rightarrow -\nabla^c f(\hat{p}) \in C^\circ_d. \text{ Similar results hold for certain } \hat{z} \]

* with respect to the continuous variables
Heat intercept insulation system

\[ T_{n+1} = T_H \]

\[ T_{i+1} = T_{i-1} \]

\[ T_0 = T_C \]

\[ \min \text{ power}(n, I, x, T) \]
Heat intercept insulation system

\[
\begin{align*}
T_{n+1} &= T_H \\
T_{i+1} &= T_i \\
T_{i-1} &= T_i \\
T_0 &= T_C \\
\end{align*}
\]

\[
\min \text{ power}(n, I, x, T) \\
\text{subject to} \quad n \in \{1, 2, \ldots, n_{\text{max}}\}, \quad I \in \mathcal{I}^{n+1} \\
T_{i-1} \leq T_i \leq T_{i+1}, \quad i = 1, 2, \ldots, n \\
\sum_{i=1}^{n+1} x_i = L, \quad x_i \geq 0, \quad i = 1, 2, \ldots, n + 1
\]
Previous heat shield studies

- **Hilal & Boom**: 1-3 intercepts, single insulator type, constant cross-sectional areas

- **Hilal & Eyssa**: 1-3 intercepts, single insulator type, variable cross-sectional areas

- **Kokkolaras, Audet & Dennis**: Variable number of intercepts, multiple insulator types, constant cross-sectional areas

- **Current work**: Variable number of intercepts, multiple insulator types, variable cross-sectional areas, load-bearing nonlinear constraints
Previous heat shield studies

- **Hilal & Boom**: 1-3 intercepts, single insulator type, constant cross-sectional areas
- **Hilal & Eyssa**: 1-3 intercepts, single insulator type, variable cross-sectional areas
Previous heat shield studies

- **Hilal & Boom**: 1-3 intercepts, single insulator type, constant cross-sectional areas
- **Hilal & Eyssa**: 1-3 intercepts, single insulator type, variable cross-sectional areas
- **Kokkolaras, Audet & Dennis**: Variable number of intercepts, multiple insulator types, constant cross-sectional areas
Previous heat shield studies

- **Hilal & Boom**: 1-3 intercepts, single insulator type, constant cross-sectional areas
- **Hilal & Eyssa**: 1-3 intercepts, single insulator type, variable cross-sectional areas
- **Kokkolaras, Audet & Dennis**: Variable number of intercepts, multiple insulator types, constant cross-sectional areas
- **Current work**: Variable number of intercepts, multiple insulator types, variable cross-sectional areas, load-bearing nonlinear constraints
### Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k(T; I_i)$</td>
<td>Thermal conductivity function for insulator $i$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Power applied to intercept $i$</td>
</tr>
<tr>
<td>$C_i$</td>
<td>Thermodynamic cycle coefficient at intercept $i$</td>
</tr>
<tr>
<td>$q_i$</td>
<td>Heat flow from intercept $i$ to $i - 1$</td>
</tr>
<tr>
<td>$A_i$</td>
<td>Cross-sectional area of insulator $i$</td>
</tr>
<tr>
<td>$\sigma(T; I_i)$</td>
<td>Maximum allowable stress function</td>
</tr>
<tr>
<td>$e(T; I_i)$</td>
<td>Unit thermal expansion function</td>
</tr>
<tr>
<td>$\rho(I_i)$</td>
<td>Density of the insulator $i$ material</td>
</tr>
<tr>
<td>$F$</td>
<td>Load (force) to be placed on the system</td>
</tr>
<tr>
<td>$m_{\text{max}}$</td>
<td>Maximum allowable mass of the insulators</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Maximum allowable % thermal contraction</td>
</tr>
</tbody>
</table>
Minimize power:
\[
\sum_{i=1}^{n} P_i = \sum_{i=1}^{n} C_i \left( \frac{T_H}{T_i} - 1 \right) (q_i - q_{i-1})
\]

By Fourier’s law:
\[
q_i = \frac{A_i}{x_i} \int_{\overline{T}_{i-1}}^{\overline{T}_i} k(T; I_i) dT, \quad i = 1, 2, \ldots, n + 1
\]
Heat shield objective and nonlinear constraints

- **Minimize power:**
  \[ \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} C_i \left( \frac{T_H^i}{T_i^i} - 1 \right) (q_i - q_{i-1}) \]

  By Fourier’s law:
  \[ q_i = \frac{A_i}{x_i} \int_{T_{i-1}}^{T_i} k(T; I_i) dT, \quad i = 1, 2, \ldots, n + 1 \]

- **Stress:**
  \[ \frac{F}{A_i} \leq \min \{ \sigma(T; I_i) : \overline{T}_{i-1} \leq T \leq \overline{T}_i \} \quad \text{(binding)} \]
Heat shield objective and nonlinear constraints

- **Minimize power:**
  \[
  \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} C_i \left( \frac{T_{H}}{T_i} - 1 \right) (q_i - q_{i-1})
  \]

  By Fourier’s law:
  \[
  q_i = \frac{A_i}{x_i} \int_{T_{i-1}}^{T_i} k(T; I_i) dT, \quad i = 1, 2, \ldots, n + 1
  \]

- **Stress:**
  \[
  \frac{F}{A_i} \leq \min \{ \sigma(T; I_i) : T_{i-1} \leq T \leq T_i \} \quad \text{(binding)}
  \]

- **Mass:**
  \[
  \sum_{i=1}^{n+1} \rho(I_i) A_i x_i \leq m_{\text{max}}
  \]
Heat shield objective and nonlinear constraints

- **Minimize power:**
  \[
  \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} C_i \left( \frac{T_i^H}{T_i} - 1 \right) (q_i - q_{i-1})
  \]

  By Fourier’s law:
  \[
  q_i = \frac{A_i}{x_i} \int_{T_{i-1}}^{T_i} k(T; I_i) dT, \quad i = 1, 2, \ldots, n + 1
  \]

- **Stress:**
  \[
  \frac{F}{A_i} \leq \min \{ \sigma(T; I_i) : T_{i-1} \leq T \leq T_i \} \quad \text{(binding)}
  \]

- **Mass:**
  \[
  \sum_{i=1}^{n+1} \rho(I_i) A_i x_i \leq m_{\text{max}}
  \]

- **Contraction:**
  \[
  \sum_{i=1}^{n+1} \left( \frac{\int_{T_{i-1}}^{T_i} e(T; I_i) k(T; I_i) dT}{\int_{T_{i-1}}^{T_i} k(T; I_i) dT} \right) \left( \frac{x_i}{L} \right) \leq \frac{\delta}{100}
  \]
Heat shield implementation

- **Materials:**
  - Nylon 6063-T5 Aluminum
  - Teflon Fiberglass Epoxy (normal cloth)
  - 304 Stainless Steel Fiberglass Epoxy (plane cloth)
  - 1020 Low Carbon Steel
Heat shield implementation

- **Materials:**
  - Nylon
  - Teflon
  - 304 Stainless Steel
  - 1020 Low Carbon Steel
  - 6063-T5 Aluminum
  - Fiberglass Epoxy (normal cloth)
  - Fiberglass Epoxy (plane cloth)

- **Material Data from Lookup Tables or Graphs:**
  - Thermal Conductivity
  - Unit Thermal Contraction
  - Maximum Allowable Stress (Tensile Yield Strength)

Pattern Search for Mixed Variable Optimization Problems – p.31/35
Heat shield implementation

Materials:
- Nylon
- 6063-T5 Aluminum
- Teflon
- Fiberglass Epoxy (normal cloth)
- 304 Stainless Steel
- Fiberglass Epoxy (plane cloth)
- 1020 Low Carbon Steel

Material Data from Lookup Tables or Graphs:
- Thermal Conductivity
- Unit Thermal Contraction
- Maximum Allowable Stress (Tensile Yield Strength)

Interpolation/Integration: Cubic splines, Simpson’s rule
Heat shield implementation

- **Materials:**
  - Nylon
  - 6063-T5 Aluminum
  - Teflon
  - Fiberglass Epoxy (normal cloth)
  - 304 Stainless Steel
  - Fiberglass Epoxy (plane cloth)
  - 1020 Low Carbon Steel

- **Material Data from Lookup Tables or Graphs:**
  - Thermal Conductivity
  - Unit Thermal Contraction
  - Maximum Allowable Stress (Tensile Yield Strength)

- **Interpolation/Integration:** Cubic splines, Simpson’s rule

- **Search/Poll:** No Search, Poll around $p_k^F$
Heat shield computational results

<table>
<thead>
<tr>
<th>Source</th>
<th>(\frac{PL}{A} \ [\frac{W}{cm}])</th>
<th>Insulators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hilal &amp; Boom</td>
<td>68.6</td>
<td>(E_pE_pE_p)</td>
</tr>
<tr>
<td>Kokkolaras et al.</td>
<td>25.3</td>
<td>(NNNNNNNNEEET)</td>
</tr>
<tr>
<td>Kokkolaras rerun</td>
<td>25.59</td>
<td>(NNNNNNTEETT)</td>
</tr>
<tr>
<td>Hilal &amp; Eyssa</td>
<td>53.2</td>
<td>(E_pE_pE_p)</td>
</tr>
<tr>
<td>with stress constraint</td>
<td>24.55</td>
<td>(EEEEEEEEEEEE)</td>
</tr>
<tr>
<td>with all constraints</td>
<td>23.77</td>
<td>(EEEEEEEEEEEE)</td>
</tr>
</tbody>
</table>
# Heat shield computational results

<table>
<thead>
<tr>
<th>Source</th>
<th>$\frac{PL}{A}$ [ W/cm ]</th>
<th>Insulators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hilal &amp; Boom</td>
<td>68.6</td>
<td>$E_pE_pE_p$</td>
</tr>
<tr>
<td>Kokkolaras et al.</td>
<td>25.3</td>
<td>$NNNNNNNNEEET$</td>
</tr>
<tr>
<td>Kokkolaras rerun</td>
<td>25.59</td>
<td>$NNNNNNTEETT$</td>
</tr>
<tr>
<td>Hilal &amp; Eyssa</td>
<td>53.2</td>
<td>$E_pE_pE_p$</td>
</tr>
<tr>
<td>with stress constraint</td>
<td>24.55</td>
<td>$EEEEEEEEEEEEEEE$</td>
</tr>
<tr>
<td>with all constraints</td>
<td>23.77</td>
<td>$EEEEEEEEEEEEEEE$</td>
</tr>
</tbody>
</table>

**Parameters:**

- $T_H = 300$ K, $T_C = 4.2$ K, $L = 100$ cm, $n_{max} = 10$
- $F = 250$ kN, $m_{max} = 10$ kg, $\delta = 5\%$
### Heat shield computational results

<table>
<thead>
<tr>
<th>Source</th>
<th>$\frac{PL}{A} \left[ \frac{W}{cm} \right]$</th>
<th>Insulators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hilal &amp; Boom</td>
<td>68.6</td>
<td>$E_pE_pE_p$</td>
</tr>
<tr>
<td>Kokkolaras et al.</td>
<td>25.3</td>
<td>$NNNNNNNNNEET$</td>
</tr>
<tr>
<td>Kokkolaras rerun</td>
<td>25.59</td>
<td>$NNNNNNNTETTT$</td>
</tr>
<tr>
<td>Hilal &amp; Eyssa</td>
<td>53.2</td>
<td>$E_pE_pE_p$</td>
</tr>
<tr>
<td>with stress constraint</td>
<td>24.55</td>
<td>$EEEEEEEEEEEEEE$</td>
</tr>
<tr>
<td>with all constraints</td>
<td>23.77</td>
<td>$EEEEEEEEEEEEEE$</td>
</tr>
</tbody>
</table>

#### Parameters:
- $\bar{T}_H = 300 \text{ K}$, $\bar{T}_C = 4.2 \text{ K}$, $L = 100 \text{ cm}$, $n_{\text{max}} = 10$
- $F = 250 \text{ kN}$, $m_{\text{max}} = 10 \text{ kg}$, $\delta = 5\%$
- Termination: $\Delta_k \leq 0.15625$
Profile of heat shield run

![Graph of Number of function evaluations vs. Power Required](image)

Pattern Search for Mixed Variable Optimization Problems – p.33/35
Heat shield filter progress

Filter after 150 evaluations

Filter after 200 evaluations

Filter after 500 evaluations

Zoom of filter on left

Zoom of filter on left

Zoom of filter on left
Conclusions

- We have found an effective method for mixed variable problems... BUT...
Conclusions

- We have found an effective method for mixed variable problems... BUT...
  - It requires serious thinking by the user as to what constitutes an acceptable solution for each problem.
Conclusions

- We have found an effective method for mixed variable problems... BUT...
  - It requires serious thinking by the user as to what constitutes an acceptable solution for each problem
  - It is not easy to incorporate surrogates with categorical variables
Conclusions

- We have found an effective method for mixed variable problems... BUT...
  - It requires serious thinking by the user as to what constitutes an acceptable solution for each problem
  - It is not easy to incorporate surrogates with categorical variables
  - It is unlikely to be effective for many categorical variables
Conclusions

- We have found an effective method for mixed variable problems... BUT...
  - It requires serious thinking by the user as to what constitutes an acceptable solution for each problem
  - It is not easy to incorporate surrogates with categorical variables
  - It is unlikely to be effective for many categorical variables
- We have analyzed the method, BUT..., better results depend on the filter method for continuous variables
Conclusions

- We have found an effective method for mixed variable problems... BUT...
  - It requires serious thinking by the user as to what constitutes an acceptable solution for each problem
  - It is not easy to incorporate surrogates with categorical variables
  - It is unlikely to be effective for many categorical variables

- We have analyzed the method, BUT.., better results depend on the filter method for continuous variables