Generalization of Separability for Optimization Problems

23rd European Workshop on Automatic Differentiation

Jens Deussen

Online, August 12, 2020
Motivation

Separability

• Consider unconstrained optimization problem

\[ y^* = \min_{x \in D \subseteq \mathbb{R}^n} f(x) \]

• For separable objective function \( f \)

\[ f(x) = \sum_{i=0}^{n-1} f_i(x_i) \]

the optimization problem decomposes into

\[ y^* = \sum_{i=0}^{n-1} \min_{x_i \in D_i \subseteq \mathbb{R}} f_i(x_i) \]

Aim and Outlook

• Generalize the definition of separability
  – to make it less strict and thus applicable to a larger set of optimization problems

• Introduction of interval-valued derivatives
  – to obtain monotonicity information of separators
  – to identify separators
### Generalization of Separability

#### Notation and Assumptions

- **Continuously differentiable functions** \( f : \mathbb{R}^n \rightarrow \mathbb{R} \)

- **Evaluation formula** \( y = f(x) \) given as code list with elemental functions \( \varphi \)

\[
\begin{align*}
  v_i &= x_i \quad i = 0, \ldots, n - 1 \\
  v_j &= \varphi_j(v_{i \prec j}) \quad j = n, \ldots, n + p \\
  y &= v_{n+p}
\end{align*}
\]

- **x:** independent variables, **y:** dependent variable, **v:** intermediate variables

- **Representation of the code list as a directed acyclic graph (DAG)**

\[
G = (V, E) ,
\]

\[
V = \{ v_j \mid j = 0, \ldots, n + p \} ,
\]

\[
E = \{ (v_i, v_j) \mid i \prec j \} .
\]
Generalization of Separability

Definition (Structural Separability)

Function $f : \mathbb{R}^n \to \mathbb{R}$ is structurally separable if its DAG has an articulation vertex $s$.

$$ f(x) = g\left(s\left((x_i)_{i \in X_1}\right), (x_i)_{i \in X_2}\right) $$

with index sets $X_1 \neq \emptyset$, $X_2 \neq \emptyset$, $X_1 \cup X_2 = \{0, \ldots, n - 1\}$ and $X_1 \cap X_2 = \emptyset$.

Attributes of Structurally Separable Functions

- Articulation vertex will be called separator $s \in \mathbb{R}$
- Definition of structural separability covers conventional separability
- Important derivative information of structural separable functions

$$ \frac{df}{dx_i}(x) = \frac{df}{ds}(x) \cdot \frac{ds}{dx_i}((x_j)_{j \in X_1}) \quad \forall i \in X_1 \quad \text{and} \quad \frac{ds}{dx_i}((x_j)_{j \in X_1}) = 0 \quad \forall i \in X_2 $$
Generalization of Separability

Theorem (Domain Decomposition of Structural Separable Functions)

Considering the box-constrained optimization problem

$$\min_{x \in D \subseteq \mathbb{R}^n} f(x) = g\left(s\left((x_i)_{i \in X_1}; (x_i)_{i \in X_2}\right)\right),$$

with structurally separable and continuously differentiable objective function $f$ and separator $s$. If the objective function is monotonic w.r.t. the separator on the domain

$$\frac{df}{ds}(x) \geq 0 \quad \forall x \in D \lor \frac{df}{ds}(x) \leq 0 \quad \forall x \in D,$$

then the optimization problem can be decomposed into

$$\min_{(x_i \in D_i)_{i \in X_2}} g\left(s^*, (x_i)_{i \in X_2}\right),$$

s.t. $s^* = \begin{cases} 
\min_{(x_i \in D_i)_{i \in X_1}} s\left((x_i)_{i \in X_1}\right) & \text{if } \frac{df}{ds}(x) \geq 0 \quad \forall x \in D, \\
\min_{(x_i \in D_i)_{i \in X_1}} -s\left((x_i)_{i \in X_1}\right) & \text{if } \frac{df}{ds}(x) \leq 0 \quad \forall x \in D.
\end{cases}$
Interval-valued Adjoints

Natural Interval Extension (NIE) of the Adjoint Model

- Interval representation
  \[ [\mathbf{x}] = [\mathbf{x}, \mathbf{x}] = \{ \mathbf{x} \in \mathbb{R}^n \mid x_i \leq x_i \leq \bar{x}_i \} \]

- Replacing all elemental functions \( \varphi \) by exact interval versions \( \varphi^* \) yields NIE

- NIE computes enclosure
  \[ [y] = f([\mathbf{x}]) \supseteq \{ f(\mathbf{x}) \in \mathbb{R} \mid \mathbf{x} \in [\mathbf{x}] \} =: [y^*] \]

- NIE of adjoint model computes enclosures of the adjoints of all independent and intermediate variables
  \[ [x^{(1)},i] = \frac{df}{dx_i}(\mathbf{x}) \cdot [y^{(1)}] \supseteq \frac{df}{dx_i}(\mathbf{x}) \cdot y^{(1)} \quad \forall \mathbf{x} \in [\mathbf{x}], \ y^{(1)} \in [y^{(1)}] \]

- Seeding the adjoint of an intermediate variable \( u \) instead yields
  \[ [x^{(1)},i] = \frac{du}{dx_i}(\mathbf{x}) \cdot [u^{(1)}] \supseteq \frac{du}{dx_i}(\mathbf{x}) \cdot u^{(1)} \quad \forall \mathbf{x} \in [\mathbf{x}], \ u^{(1)} \in [u^{(1)}] \]
Interval-valued Adjoints for Separability

Monotonicity Check for Separators

- Check for monotonicity on domain with interval-valued adjoints
- One interval-valued adjoint evaluation is required for all separators

Detection of a Separator by Checking Monotonic Intermediates

- Seed $u_{(1)} = \frac{df}{du}([x])$ and evaluate adjoint model to obtain

$$x_{(1),i} = \frac{du}{dx_i}([x]) \cdot \frac{df}{du}([x]) .$$

- $u$ is a structural separator only if

$$\frac{du}{dx_i}([x]) \cdot \frac{df}{du}([x]) = \frac{df}{dx_i}([x]) \forall i \in X_1 \land \frac{du}{dx_i}([x]) \cdot \frac{df}{du}([x]) = 0 \forall i \in X_2$$

- One additional interval-valued adjoint evaluation per separator candidate
Proof of Concept Example

**Function**

- Box-constrained global optimization problem

\[
\min_{\mathbf{x} \in [-2,3]^n \subseteq \mathbb{R}^n} f(\mathbf{x}) = y_n
\]

\[
y_{i+1} = \exp(x_i^2 + y_i - 1)
\]

\[
y_0 = 1
\]

- Objective is monotonic w.r.t. separators \(y_i, i = \{1, \ldots, n - 1\}\)

\[
\frac{df}{dy_i}(\mathbf{x}) = \prod_{j=i}^{n-1} \frac{\partial y_{j+1}}{\partial y_j} = \prod_{j=i}^{n-1} y_{j+1} \geq 1
\]

**Remark**

- This function is monotonic everywhere, but the approach also works if it would be monotone only on the current subdomain.
## Proof of Concept Example

### Branch and Bound Prototype

- Subdivision into $2^n$ subdomains
- Value check and elimination of subdomain $x$ if $f([x]) > \bar{y}^*$
- First-order optimality check and elimination of subdomain $x$ if $0 \notin \frac{df}{dx_i}([x])$
- Improvement of $\bar{y}^*$ by function evaluation at midpoint of subdomain
- Monotonicity check for separators

### Interval-valued Adjoint Implementation

- Interval type from the Boost library
- First-order adjoint type from dco/c++

---

1. [https://www.nag.co.uk/content/adjoint-algorithmic-differentiation](https://www.nag.co.uk/content/adjoint-algorithmic-differentiation)
Proof of Concept Example

Results

- Separation enables decomposition into problems of size 1
- Computational overhead for monotonicity check is negligible if separators are known a priori
- Computation for $n = 1000$ is still possible with separation
## Summary

- Generalized the concept of separability
- If monotonicity condition is fulfilled the size of the optimization problem reduces
- In the context of a branch and bound algorithm checking first-order optimality condition by adjoints, the overhead of the monotonicity check is negligible

## Outlook

- Measurement of the overhead in performance per branch and bound node
- Prototype already supports detection of separators, but for solving sub-problems efficiently, an early return of the separator value is required
- Implementation of McCormick relaxations as substitute for interval computation
Thank you!