IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

ENSEEIHT-IRIT, Team APO, Toulouse, France
collaboration with LAPLACE-GREM3: B. Nogarède, Y. Lefèvre, J. Fontchastagner

EWMINLP 2010
1 – Direct Problem of Design
1 – Direct Problem of Design

- Direct Solve of the Maxwell’s Equations
  - By Finite Element Methods
1 – Direct Problem of Design

Direct Solve of the Maxwell’s Equations By Finite Element Methods

CHARACTERISTICAL VALUES FOR A GIVEN STRUCTURE
1 – Direct Problem of Design

Direct Solve of the Maxwell’s Equations By Finite Element Methods

CHARACTERISTIC VALUES FOR A GIVEN STRUCTURE

2 – Inverse Problem of Dimensioning

Analytical Model of the given structure

Functions:
- min mass
- ... 
- min volume

Some assumptions on the Maxwell’s Equations => Analytical Models (simple)
Direct and Inverse Problem of Design

1 – Direct Problem of Design

Direct Solve of the Maxwell’s Equations By Finite Element Methods

CHARACTERISTIC VALUES FOR A GIVEN STRUCTURE

2 – Inverse Problem of Dimensioning

Analytical Model of the given structure

Functions:
- \( \text{min mass} \)
- \( \text{...} \)
- \( \text{min volume} \)

Some assumptions on the Maxwell’s Equations => Analytical Models (simple)

3 – Inverse Problem of Design

Type of structure, dimensions and constitutions

Objectives:
- \( \text{min masse} \)
- \( \text{...} \)
- \( \text{min volume} \)

Model associating many different elementary structures:
- General Model
Mathematical Formulation

Dimensioning Inverse Problem:

\[
\begin{align*}
\min_{x \in X \subseteq \mathbb{R}^n} & \quad f(x) \\
\text{s.t.} & \quad g_i(x) \leq 0 \quad \forall i \in \{1, \ldots, n_g\} \\
& \quad h_j(x) = 0 \quad \forall j \in \{1, \ldots, n_h\}
\end{align*}
\]

More General Inverse Problem of Design:

\[
\begin{align*}
\min_{x \in X \subseteq \mathbb{R}^{nr}, z \in Z \subseteq \mathbb{N}^{ne}, \sigma \in \prod_{i=1}^{nc} K_i, b \in \{0,1\}^{nb}} & \quad f(x, z, \sigma, b) \\
\text{s.t.} & \quad g_i(x, z, \sigma, b) \leq 0 \quad \forall i \in \{1, \ldots, n_g\} \\
& \quad h_j(x, z, \sigma, b) = 0 \quad \forall j \in \{1, \ldots, n_h\}
\end{align*}
\]
Mathematical Formulation

- **Dimensioning Inverse Problem:**

\[
\begin{split}
\min_{x \in X \subseteq \mathbb{R}^n} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0 \quad \forall i \in \{1, \ldots, n_g\} \\
& \quad h_j(x) = 0 \quad \forall j \in \{1, \ldots, n_h\}
\end{split}
\]

- **More General Inverse Problem of Design:**

\[
\begin{split}
\min_{x \in X \subseteq \mathbb{R}^{nr}, z \in Z \subseteq \mathbb{N}^{ne}, \sigma \in \prod_{i=1}^{nc} \mathbb{K}, b \in \{0, 1\}^{nb}} & \quad f(x, z, \sigma, b) \\
\text{subject to} & \quad g_i(x, z, \sigma, b) \leq 0 \quad \forall i \in \{1, \ldots, n_g\} \\
& \quad h_j(x, z, \sigma, b) = 0 \quad \forall j \in \{1, \ldots, n_h\}
\end{split}
\]
Introduction to Deterministic Global Optimization

IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Design of Actuators
Extension to Black-Box Constraint Realizations & Conclusion
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Design of Actuators
Extension to Black-Box Constraint Realizations & Conclusion

Introduction to Deterministic Global Optimization

\[ f^* \]
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Design of Actuators
Extension to Black-Box Constraint
Realizations & Conclusion
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms

Design of Actuators

Extension to Black-Box Constraint

Realizations & Conclusion

Introduction to Deterministic Global Optimization

![Graph showing optimization process](image)

- $x_0 \rightarrow x_k \rightarrow X$
- Local Minimum
Interval Branch and Bound Algorithms

Interval Analysis
Interval methods for bounds
Branch and Bound Algorithms
Propagation Techniques
Algorithms for Mixed Problems
IBBA

Design of Electromechanical Actuators
Rotating Machines with Magnetic Effects
Combinatorial Models for Electrical Machines
Numerical Examples of Electrical Machines

Design with a Black-Box Constraint
Numerical Validations
IBBA+NUMT
Numerical Examples

Some Realizations and Conclusion
Outline

**Interval Branch and Bound Algorithms**

- Interval Analysis
- Interval methods for bounds
- Branch and Bound Algorithms
- Propagation Techniques
- Algorithms for Mixed Problems
- IBBA

**Design of Electromechanical Actuators**

- Rotating Machines with Magnetic Effects
- Combinatorial Models for Electrical Machines
- Numerical Examples of Electrical Machines

**Design with a Black-Box Constraint**

- Numerical Validations
- IBBA+NUMT
- Numerical Examples

Some Realizations and Conclusion
Outline

- Interval Branch and Bound Algorithms
  - Interval Analysis
  - Interval methods for bounds
  - Branch and Bound Algorithms
  - Propagation Techniques
  - Algorithms for Mixed Problems
  - IBBA

- Design of Electromechanical Actuators
  - Rotating Machines with Magnetic Effects
  - Combinatorial Models for Electrical Machines
  - Numerical Examples of Electrical Machines

- Design with a Black-Box Constraint
  - Numerical Validations
  - IBBA+NUMT
  - Numerical Examples

Some Realizations and Conclusion
Outline

Interval Branch and Bound Algorithms
   Interval Analysis
   Interval methods for bounds
   Branch and Bound Algorithms
   Propagation Techniques
   Algorithms for Mixed Problems
   IBBA

Design of Electromechanical Actuators
   Rotating Machines with Magnetic Effects
   Combinatorial Models for Electrical Machines
   Numerical Examples of Electrical Machines

Design with a Black-Box Constraint
   Numerical Validations
   IBBA+NUMT
   Numerical Examples

Some Realizations and Conclusion
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Interval Analysis
Methods for bounds
Branch and Bound Algorithms
Propagation Techniques
Mixed Problems

IBBA
Design of Actuators
Extension to Black-Box Constraint
Realizations & Conclusion

Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Interval Analysis
Methods for bounds
Branch and Bound Algorithms
Propagation Techniques
Mixed Problems
IBBA

Design of Actuators
Extension to Black-Box Constraint
Realizations & Conclusion
Interval Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Interval Analysis
Methods for bounds
Branch and Bound Algorithms
Propagation Techniques
Mixed Problems
IBBA

Design of Actuators
Extension to Black-Box Constraint Realizations & Conclusion

Interval Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case
Interval Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case
Interval Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case
Interval Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case

IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Interval Analysis
Methods for bounds
Branch and Bound Algorithms
Propagation Techniques
Mixed Problems
IBBA

Design of Actuators
Extension to Black-Box Constraint Realizations & Conclusion
Interval Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case
Interval Branch and Bound Algorithm for Continuous Optimization Problems: Unconstrained Case
Interval Analysis

Let $X = [a, b]$ and $Y = [c, d]$ 2 intervals. Moore (1966) defines the interval arithmetic as follows:

\[
\begin{align*}
[a, b] + [c, d] &= [a + c, b + d] \\
[a, b] - [c, d] &= [a - d, b - c] \\
[a, b] \times [c, d] &= [\min\{ac, ad, bc, bd\}, \\
&\quad \ max\{ac, ad, bc, bd\}] \\
[a, b] \div [c, d] &= [a, b] \times \left[\frac{1}{d}, \frac{1}{c}\right] \text{ if } 0 \notin [c, d].
\end{align*}
\]

Remark

Subtraction and division are not the inverse operations of addition and respectively multiplication.

Difficulties:

$\div 0 \implies$ extended interval arithmetic, (E. Hansen).

Numerical errors $\implies$ rounded interval analysis, (Moore).
Interval Analysis

Let $X = [a, b]$ and $Y = [c, d]$ 2 intervals.
Moore (1966) defines the interval arithmetic as follows:

\[
\begin{align*}
[a, b] + [c, d] &= [a + c, b + d] \\
[a, b] - [c, d] &= [a - d, b - c] \\
[a, b] \times [c, d] &= \left[\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}\right] \\
[a, b] \div [c, d] &= [a, b] \times \left[\frac{1}{d}, \frac{1}{c}\right] \text{ if } 0 \notin [c, d].
\end{align*}
\]

**Remark**

Subtraction and division are not the inverse operations of addition and respectively multiplication.

**Difficulties:**

$\div 0 \implies$ extended interval arithmetic, (E. Hansen).
Numerical errors $\implies$ rounded interval analysis, (Moore).
Some Properties of Interval Analysis and Inclusion Function

**Property**

For all $x \in X$ and $y \in Y$, one has: $x \star y \in X \star Y$, where $\star$ is $+,-,\times,\div$.

**Property**

Let $A, B, C$ 3 intervals, therefore $A \times (B + C) \subseteq A \times B + A \times C$.

**Property**

Let $Y_1, Y_2, Z_1, Z_2$ 4 intervals, if $Y_1 \subseteq Z_1$ and if $Y_2 \subseteq Z_2$ then $Y_1 \star Y_2 \subseteq Z_1 \star Z_2$ where $\star$ is $+,-,\times,\div$.

**Definition**

An **inclusion function** $F(X)$ of $f$ over a box $X$ is such that

$$f(X) := [\min_{x \in X} f(x), \max_{x \in X} f(x)] \subseteq F(X) = [F^L(X), F^U(X)]$$
Natural Extension: an Inclusion Function

Theorem

The natural extension into interval of an expression of \( f \) over a box \( X \) is an inclusion function.

Example

Let \( f(x) = x^2 - x + 1 \) and \( x \in X = [0, 1] \)

Inclusion functions:

\[
\begin{align*}
F_1(X) &= X^2 - X + 1 = [0, 1]^2 - [0, 1] + [1, 1] = [0, 2], \\
F_2(X) &= X(X - 1) + 1 = [0, 1](0, 1) - 1) + [1, 1] = [0, 1], \\
F_3(X) &= (X - \frac{1}{2})^2 + \frac{3}{4} = \left[ -\frac{1}{2}, \frac{1}{2} \right]^2 + \frac{3}{4} = \left[ \frac{3}{4}, 1 \right].
\end{align*}
\]
Natural Extension: an Inclusion Function

**Theorem**

The natural extension into interval of an expression of $f$ over a box $X$ is an inclusion function.

**Example**

Let $f(x) = x^2 - x + 1$ and $x \in X = [0, 1]

Inclusion functions:

- $F_1(X) = X^2 - X + 1 = [0, 1]^2 - [0, 1] + [1, 1] = [0, 2],$
- $F_2(X) = X(X - 1) + 1 = [0, 1][0, 1] - 1 + [1, 1] = [0, 1] \times [-1, 0] + [1, 1] = [0, 1],$
- $F_3(X) = \left( X - \frac{1}{2} \right)^2 + \frac{3}{4} = \left[ -\frac{1}{2}, \frac{1}{2} \right]^2 + \frac{3}{4} = \left[ \frac{3}{4}, 1 \right],$
Natural Extension: an Inclusion Function

Theorem

The natural extension into interval of an expression of f over a box X is an inclusion function.

Example

Let \( f(x) = x^2 - x + 1 \) and \( x \in X = [0, 1] \)

Inclusion functions:

- \( F_1(X) = X^2 - X + 1 = [0, 1]^2 - [0, 1] + [1, 1] = [0, 2] \),
- \( F_2(X) = X(X - 1) + 1 = [0, 1]([0, 1] - 1) + [1, 1] = [0, 1] \times [-1, 0] + [1, 1] = [0, 1] \),
- \( F_3(X) = \left(X - \frac{1}{2}\right)^2 + \frac{3}{4} = \left[-\frac{1}{2}, \frac{1}{2}\right]^2 + \frac{3}{4} = \left[\frac{3}{4}, 1\right] \),
Natural Extension: an Inclusion Function

**Theorem**
The *natural extension* into interval of an expression of $f$ over a box $X$ is an inclusion function.

**Example**
Let $f(x) = x^2 - x + 1$ and $x \in X = [0, 1]$

**Inclusion functions:**

- $F_1(X) = X^2 - X + 1 = [0, 1]^2 - [0, 1] + [1, 1] = [0, 2]$
- $F_2(X) = X(X - 1) + 1 = [0, 1]( [0, 1] - 1) + [1, 1] = [0, 1] \times [-1, 0] + [1, 1] = [0, 1]$
- $F_3(X) = \left( X - \frac{1}{2} \right)^2 + \frac{3}{4} = \left[ -\frac{1}{2}, \frac{1}{2} \right]^2 + \frac{3}{4} = \left[ \frac{3}{4}, 1 \right]$, 

Natural Extension: an Inclusion Function

**Theorem**

The natural extension into interval of an expression of \( f \) over a box \( X \) is an inclusion function.

**Example**

Let \( f(x) = x^2 - x + 1 \) and \( x \in X = [0, 1] \)

Inclusion functions:

\[
\begin{align*}
F_1(X) &= X^2 - X + 1 = [0, 1]^2 - [0, 1] + [1, 1] = [0, 2], \\
F_2(X) &= X(X - 1) + 1 = [0, 1]([0, 1] - 1) + [1, 1] = [0, 1], \\
F_3(X) &= \left(X - \frac{1}{2}\right)^2 + \frac{3}{4} = \left[-\frac{1}{2}, \frac{1}{2}\right]^2 + \frac{3}{4} = \left[\frac{3}{4}, 1\right],
\end{align*}
\]
Inclusion Functions based on Taylor’s Expansions

Let $f$ be a differentiable function with one variable and $x$, $y$ and $\xi$, 3 variables of $X$ an interval of $\mathbb{R}$.

$$f(x) = f(y) + (x - y)f'(y) + \frac{(x - y)^2}{2} f''(y) + \ldots + \frac{(x - y)^n}{n!} f^{(n)}(\xi)$$

Let denote $F^{(n)}(X)$ an enclosure of $f^{(n)}(\xi)$ over $X$ (computed with an interval automatic differentiation tool).

Hence,

$$f(x) \in f(y) + (x - y)f'(y) + \frac{(x - y)^2}{2} f''(y) + \ldots + \frac{(x - y)^n}{n!} F^{(n)}(X)$$

2 inclusion functions:

- $T_1(y, X) = f(y) + (X - y)F'(X)$
- $T_2(y, X) = f(y) + (X - y)f'(y) + \frac{(X - y)^2}{2} F''(X)$
Inclusion Functions based on Taylor’s Expansions

- Generalized Interval Arithmetic (E. Hansen - 1982)
- Baumann centered Form (Baumann - 1988),
- Linear Boundary Value Form (Neumaier - 1992, P. Hansen - 1992)
- Admissible Simplex Form (Lagouanelle - 1997)
- Affine Arithmetic (Andrade, Stolfy, De Figueiredo - 1993)
- ... 

+ Accelerating techniques for the unconstrained case.
Mathematical Formulation

\[
\begin{align*}
\min_{x \in X \subseteq \mathbb{R}^n} & \quad f(x) \\
g_i(x) & \leq 0 \quad \forall i \in \{1, \ldots, n_g\} \\
h_j(x) & = 0 \quad \forall j \in \{1, \ldots, n_h\}
\end{align*}
\]
Principle of a Branch and Bound Algorithm for a problem with constraints

- **Choice and Subdivision of the box** $X$, (in 2 parts by step): list of possible solutions,
- **Reduction of the sub-boxes**, by using a constraint propagation technique,
- **Computation of bounds** of the functions $F$, $G_j$, $H_j$ on the sub-boxes, - inclusion functions -
- **Elimination of the sub-boxes** which cannot contain the global optimum: $F^L(X) > \tilde{f}$ or $G^L_i(X) > 0$ or $0 \not\in H(X)$, where $\tilde{f}$ denotes the current solution,
- **STOP** when accurate enclosures of the optimum are obtained.
Propagation Techniques

\( g(x) \in [a, b] \) is a constraint \( \Longrightarrow \) implicit (or explicit) relations between the variables of the problem.

Idea: use some deduction steps for reducing the box \( X \).

Linear case: if \( g(x) = \sum_{i=1}^{n} a_i x_i \) then:

\[
X_k := \left( [a, b] - \sum_{i=1, i \neq k}^{n} a_i X_i \right) \cap X_k, \text{ si } a_k \neq 0. \quad (1)
\]

where \( k \) is in \( \{1, \ldots, n\} \) and \( X_i \) is the \( i^{\text{th}} \) component of \( X \).

Non-linear case: Idea (E. Hansen): one linearizes using \( T_1 \) (or \( T_2 \)). Then one solve a linear system with interval coefficients.

Other Idea: construction of the calculus tree and propagation.
Propagation Techniques

$g(x) \in [a, b]$ is a constraint $\implies$ implicit (or explicit) relations between the variables of the problem.

Idea: use some deduction steps for reducing the box $X$.

Linear case: if $g(x) = \sum_{i=1}^{n} a_i x_i$ then:

$$X_k := \left( [a, b] - \sum_{i=1, i \neq k}^{n} a_i X_i \right) \cap X_k, \text{ if } a_k \neq 0. \quad (1)$$

where $k$ is in $\{1, \cdots, n\}$ and $X_i$ is the $i^{th}$ component of $X$.

Non-linear case: Idea (E. Hansen): one linearizes using $T_1$ (or $T_2$). Then one solve a linear system with interval coefficients. Other Idea: construction of the calculus tree and propagation.
Propagation Techniques

\[ g(x) \in [a, b] \] is a constraint \(\iff\) implicit (or explicit) relations between the variables of the problem.

**Idea:** use some deduction steps for reducing the box \(X\).

**Linear case:** if \( g(x) = \sum_{i=1}^{n} a_i x_i \) then:

\[
X_k := \left( [a, b] - \sum_{i=1, i \neq k}^{n} a_i X_i \right) \cap X_k, \text{ si } a_k \neq 0.
\] (1)

where \( k \) is in \( \{1, \cdots, n\} \) and \( X_i \) is the \( i^{\text{th}} \) component of \( X \).

**Non-linear case:** Idea (E. Hansen): one linearizes using \( T_1 \) (or \( T_2 \)). Then one solve a linear system with interval coefficients.

Other Idea: construction of the calculus tree and propagation.
Propagation Techniques

\[ g(x) \in [a, b] \text{ is a constraint } \implies \text{ implicit (or explicit) relations between the variables of the problem.} \]

**Idea:** use some deduction steps for reducing the box \( X \).

**Linear case:** if \( g(x) = \sum_{i=1}^{n} a_i x_i \) then:

\[
X_k := \left( [a, b] - \sum_{i=1, i \neq k}^{n} a_i X_i \right) \frac{a_k}{a_k} \cap X_k, \text{ si } a_k \neq 0. \tag{1}
\]

where \( k \) is in \( \{1, \cdots, n\} \) and \( X_i \) is the \( i^{th} \) component of \( X \).

**Non-linear case:** Idea (E. Hansen): one linearizes using \( T_1 \) (or \( T_2 \)). Then one solve a linear system with interval coefficients.

Other Idea: construction of the calculus tree and propagation.
Propagation Techniques

\[ g(x) \in [a, b] \] is a constraint \( \implies \) implicit (or explicit) relations between the variables of the problem.

**Idea:** use some deduction steps for reducing the box \( X \).

**Linear case:** if \( g(x) = \sum_{i=1}^{n} a_i x_i \) then:

\[
X_k := \left( [a, b] - \sum_{i=1, i \neq k}^{n} a_i X_i \right) \cap X_k, \quad \text{si } a_k \neq 0. \tag{1}
\]

where \( k \) is in \( \{1, \cdots, n\} \) and \( X_i \) is the \( i^{th} \) component of \( X \).

**Non-linear case:** Idea (E. Hansen): one linearizes using \( T_1 \) (or \( T_2 \)). Then one solve a linear system with interval coefficients.

**Other Idea:** construction of the calculus tree and propagation.
Example of Propagation Technique based on the Calculus Tree

Let \( g(x) = 2x_3x_2 + x_1 \) and

\[
g(x) = 3
\]

where \( x_i \in [1, 3] \) for all \( i \in \{1, 2, 3\} \).
Example of Propagation Technique based on the Calculus Tree

Let \( g(x) = 2x_3x_2 + x_1 \) and

\[
g(x) = 3
\]

where \( x_i \in [1, 3] \) for all \( i \in \{1, 2, 3\} \).

The propagation is:
Example of Propagation Technique based on the Calculus Tree

Let \( g(x) = 2x_3x_2 + x_1 \) and

\[
g(x) = 3
\]

where \( x_i \in [1, 3] \) for all \( i \in \{1, 2, 3\} \).

The propagation is:
Mathematical Formulation

\[
\begin{align*}
\min_{\substack{x \in X \subseteq \mathbb{R}^{nr}, z \in Z \subseteq \mathbb{N}^{ne}, \\ \sigma \in \prod_{i=1}^{nc} K_i, b \in \{0,1\}^{nb}}} & \quad f(x, z, \sigma, b) \\
\text{s.t.} & \quad g_i(x, z, \sigma, b) \leq 0 \quad \forall i \in \{1, \ldots, n_g\} \\
& \quad h_j(x, z, \sigma, b) = 0 \quad \forall j \in \{1, \ldots, n_h\}
\end{align*}
\]
Continuous variables: real variables (dimensions of an electrical machines such as the diameter).

Discrete variables: integer (number of pair of poles of a machine), boolean (machine with or without slot), categorical variable (which kind of magnet is used).

For integer and boolean variables $\implies$ relaxation for computing bounds $+$ particular bisection technique and propagation.

For categorical variables $\implies$ we introduce 4 particular algorithms with propagation and retro-propagation $+$ properties about the bisection techniques.
Algorithms for Mixed Problems: Integer Variables

**Computation of bounds:**
For integer and boolean variables \(\implies\) *relaxation* for computing bounds.
All sets \(Y = \{i, \cdots, j\}\) are relaxed by \(\overline{Y} = [i, j] \implies\) use of interval arithmetic.

**Bisection technique:**
First a weight is added to bisect first those kind of variables.
Examples of bisection of an integer variable:
\(Y = \{1, \cdots, 10\} \implies Y_1 = \{1, \cdots, 5\}\) and \(Y_2 = \{6, \cdots, 10\}\)
\(Y = \{0, \cdots, 4\} \implies Y_1 = \{0, 1, 2\}\) and \(Y_2 = \{3, 4\}\)

**Acceleration techniques:** A lot of accelerating techniques can be adapted to the mixed-integer case.
For example the propagation technique:
In the propagation tree if a leave corresponds to an integer variable: the bounds can be converted as follows:
\(\overline{Y} = [lb, ub] \implies Y := \{\lceil lb \rceil, \cdots, \lfloor ub \rfloor\}\)
Algorithms for Mixed Problems: Integer Variables

**Computation of bounds:**
For integer and boolean variables \( \implies \text{relaxation for computing bounds.} \)
All sets \( Y = \{i, \cdots, j\} \) are relaxed by \( \overline{Y} = [i, j] \longrightarrow \text{use of interval arithmetic.} \)

**Bisection technique:**
First **a weight** is added to bisect first those kind of variables.
Examples of bisection of an integer variable:
\[
Y = \{1, \cdots, 10\} \longrightarrow Y_1 = \{1, \cdots, 5\} \text{ and } Y_2 = \{6, \cdots, 10\} \\
Y = \{0, \cdots, 4\} \longrightarrow Y_1 = \{0, 1, 2\} \text{ and } Y_2 = \{3, 4\}
\]

**Acceleration techniques:** A lot of accelerating techniques can be adapted to the mixed-integer case.
For example the propagation technique:
In the propagation tree if a leave corresponds to an integer variable: the bounds can be converted as follows:
\[
\overline{Y} = [lb, ub] \longrightarrow Y := \{\lfloor lb \rfloor, \cdots, \lfloor ub \rfloor\}
\]
Computation of bounds:
For integer and boolean variables → relaxation for computing bounds.
All sets $Y = \{i, \cdots, j\}$ are relaxed by $\overline{Y} = [i, j]$ → use of interval arithmetic.

Bisection technique:
First a weight is added to bisect first those kind of variables.
Examples of bisection of an integer variable:
$Y = \{1, \cdots, 10\} → Y_1 = \{1, \cdots, 5\}$ and $Y_2 = \{6, \cdots, 10\}$
$Y = \{0, \cdots, 4\} → Y_1 = \{0, 1, 2\}$ and $Y_2 = \{3, 4\}$

Acceleration techniques: A lot of accelerating techniques can be adapted to the mixed-integer case.
For example the propagation technique:
In the propagation tree if a leave corresponds to an integer variable: the bounds can be converted as follows:
$\overline{Y} = [lb, ub] → Y := \{\lceil lb \rceil, \cdots, \lfloor ub \rfloor\}$
Algorithms for Mixed Problems: Categorical Variables

(+B. Jeannet)

\[ k \in \prod_{i=1}^{m} K_i \]

where \( K_i \) is an enumerated set (without order) and \( k_i \) is a categorical variable.
To manipulate these variables \( \rightarrow \) a real function:

\[ c : K_i \rightarrow \mathbb{R} \]

univariate real functions are sufficient.

**Bisection technique:**
As for integer variables.

**Computation of bounds:**
relaxation does not work, (because the categorical sets are not ordered).
Algorithms for Mixed Problems: Categorical Variables

→ 4 methods:

- **Method M₁**: \( Cc := [\min_{k_i \in K_i} c(k_i), \max_{k_i \in K_i} c(k_i)] \). if \( K_i = \{1\} \), \( C := [c(1), c(1)] \), ..., if \( K_i = \{j\} \), \( C := [c(j), c(j)] \) else \( C := Cc \)

- **Method M₂**: \( C := [\min_{k_i \in K_i} c(k), \max_{k_i \in K_i} c(k)] \).

- **Method M₃**: Sort the enumerate set corresponding to the value of \( c(k) \); difficulty: another function \( d \) use \( k \).

- **Method M₄**: Introduction of a new real variable \( y \in Cc \) and a constraint: \( \exists k \in K_i, y = c(k) \). \( y \) will replace \( k \) and bisections are done with \( y \). Computation of bounds are easy (IA), and some propagation techniques are used to deal with \( y = c(k) \).
Algorithms for Mixed Problems: Categorical Variables

--- 4 methods:

- **Method $M_1$:** $Cc := [\min_{k_i \in K_i} c(k_i), \max_{k_i \in K_i} c(k_i)]$. If $K_i = \{1\}$, $C := [c(1), c(1)]$, ..., if $K_i = \{j\}$, $C := [c(j), c(j)]$ else $C := Cc$

- **Method $M_2$:** $C := [\min_{k_i \in K_i} c(k), \max_{k_i \in K_i} c(k)]$.

- **Method $M_3$:** Sort the enumerate set corresponding to the value of $c(k)$; difficulty: another function $d$ use $k$.

- **Method $M_4$:** Introduction of a new real variable $y \in Cc$ and a constraint: $\exists k \in K_i, y = c(k)$. $y$ will replace $k$ and bisections are done with $y$. Computation of bounds are easy (IA), and some propagation techniques are used to deal with $y = c(k)$. 
Algorithms for Mixed Problems: Categorical Variables

→ 4 methods:

- Method $M_1$: $Cc := [\min_{k_i \in K_i} c(k_i), \max_{k_i \in K_i} c(k_i)]$. if $K_i = \{1\}$, $C := [c(1), c(1)]$, ..., if $K_i = \{j\}$, $C := [c(j), c(j)]$ else $C := Cc$

- Method $M_2$: $C := [\min_{k_i \in K_i} c(k), \max_{k_i \in K_i} c(k)]$.

- Method $M_3$: Sort the enumerate set corresponding to the value of $c(k)$; difficulty: another function $d$ use $k$.

- Method $M_4$: Introduction of a new real variable $y \in Cc$ and a constraint: $\exists k \in K_i, y = c(k)$. $y$ will replace $k$ and bisections are done with $y$. Computation of bounds are easy (IA), and some propagation techniques are used to deal with $y = c(k)$. 
Algorithms for Mixed Problems: Categorical Variables

→ 4 methods:

- Method $M_1$: $Cc := \left[ \min_{k_i \in K_i} c(k_i), \max_{k_i \in K_i} c(k_i) \right]$. if $K_i = \{1\}$, $C := [c(1), c(1)]$, ..., if $K_i = \{j\}$, $C := [c(j), c(j)]$ else $C := Cc$

- Method $M_2$: $C := \left[ \min_{k_i \in K_i} c(k), \max_{k_i \in K_i} c(k) \right]$.

- Method $M_3$: Sort the enumerate set corresponding to the value of $c(k)$; difficulty: another function $d$ use $k$.

- Method $M_4$: Introduction of a new real variable $y \in Cc$ and a constraint: $\exists k \in K_i, y = c(k)$. $y$ will replace $k$ and bisections are done with $y$. Computation of bounds are easy (IA), and some propagation techniques are used to deal with $y = c(k)$. 
Proposition

It is better to bisect by \( Y \) than by \( K \)
i.e., \( Y \rightarrow Y_1, Y_2 \) and \( K \rightarrow K_1, K_2 \) therefore,

\[
C(K_1) \subseteq Y_1 \implies Y_2 \subseteq C(K_2)
\]

and \( C(K_2) \subseteq Y_2 \implies Y_1 \subseteq C(K_1) \).

Proposition

When values of \( c_j(k) \) are uniformly distributed \( \implies \)

\[
p(Y_j \subseteq C_j(K)) = 1 - \frac{|K|}{(|K| - 1)2^{\frac{|K|}{2}}}.
\]

examples: \( |K| = 4, p = 0.66, |K| = 6, p = 0.85, \) and \( |K| = 8, p = 0.92. \)
Algorithms for Mixed Problems: Properties of $M_4$

Properties of Method $M_4$ using propagation techniques.

**Proposition**

*It is better to bisect by $Y$ than by $K*

i.e., $Y \longrightarrow Y_1, Y_2$ and $K \longrightarrow K_1, K_2$ therefore,

$$C(K_1) \subseteq Y_1 \implies Y_2 \subseteq C(K_2)$$

and $C(K_2) \subseteq Y_2 \implies Y_1 \subseteq C(K_1)$.

**Proposition**

*When values of $c_j(k)$ are uniformly distributed*

$$p(Y_j \subseteq C_j(K)) = 1 - \frac{|K|}{(|K| - 1)^2}$$

examples: $|K| = 4, p = 0.66$, $|K| = 6, p = 0.85$, and $|K| = 8, p = 0.92$. 
Algorithms for Mixed Problems: Numerical examples

\[ f_1(x_1, x_2, c(k_1)) = 20c_1(k_1)x_2^2 + 2c_2(k_1)x_1x_2, \]
\[ f_2(x_1, x_2, c(k_1)) = 20\frac{x_1^2}{(1 - c_1(k_1))^2} + 2c_2(k_1)x_1x_2, \]

where \( k_1 \in K_1 \) with \( |K_1| = 6 \), and \( x_1 \in [-15, 25] \), \( x_2 \in [3, 10] \). Following \( k_1 \), the two univariate functions \( c_1 \) and \( c_2 \) take the following values:

<table>
<thead>
<tr>
<th>( k_1 )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_1 )</td>
<td>0.5</td>
<td>0.3</td>
<td>0.8</td>
<td>0.1</td>
<td>0.9</td>
<td>0.12</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>−0.5</td>
<td>0.6</td>
<td>0.1</td>
<td>1.5</td>
<td>−1</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Global minimum for $f_1$ is 112.5 corresponding to $k_1^* = 4$, $x_1^* = -7.5$ and $x_2^* = 10$.

<table>
<thead>
<tr>
<th></th>
<th>$M_0$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_{4v1}$</th>
<th>$M_{4v2}$</th>
<th>$M_{4v3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nb its</td>
<td>45799</td>
<td>9914</td>
<td>8378</td>
<td>4210</td>
<td>3271</td>
<td>3148</td>
</tr>
<tr>
<td>CPU (s)</td>
<td>42.24</td>
<td>3.2</td>
<td>3.31</td>
<td>0.71</td>
<td>0.45</td>
<td>0.27</td>
</tr>
</tbody>
</table>

The result for $f_2$ is 9.1 for the minimum value and $(4, -0.6, 10)$ for its corresponding solution.

<table>
<thead>
<tr>
<th></th>
<th>$M_0$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_{4v1}$</th>
<th>$M_{4v2}$</th>
<th>$M_{4v3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nb its</td>
<td>77928</td>
<td>35019</td>
<td>29107</td>
<td>15230</td>
<td>10794</td>
<td>6466</td>
</tr>
<tr>
<td>CPU (s)</td>
<td>169</td>
<td>43.01</td>
<td>35.35</td>
<td>2.57</td>
<td>1.46</td>
<td>0.86</td>
</tr>
</tbody>
</table>
Global minimum for $f_1$ is 112.5 corresponding to $k_1^* = 4$, $x_1^* = -7.5$ and $x_2^* = 10$.

<table>
<thead>
<tr>
<th>Nb its</th>
<th>$M_0$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_{4v1}$</th>
<th>$M_{4v2}$</th>
<th>$M_{4v3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU (s)</td>
<td>42.24</td>
<td>3.2</td>
<td>3.31</td>
<td>0.71</td>
<td>0.45</td>
<td>0.27</td>
</tr>
</tbody>
</table>

The result for $f_2$ is 9.1 for the minimum value and $(4, -0.6, 10)$ for its corresponding solution.

<table>
<thead>
<tr>
<th>Nb its</th>
<th>$M_0$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_{4v1}$</th>
<th>$M_{4v2}$</th>
<th>$M_{4v3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU (s)</td>
<td>169</td>
<td>43.01</td>
<td>35.35</td>
<td>2.57</td>
<td>1.46</td>
<td>0.86</td>
</tr>
</tbody>
</table>
Interval Branch and Bound Algorithms: IBBA

Summarize of IBBA:

- **Heuristics:**
  - Larger or deeper first, lowest lower bound first...
  - Bisection or Multisections
  - Bisection by the middle or by the Baumann center...
  - ...

- **Computations of Bounds:** $NE, T_1, T_B, ASF +$Affine Arithmetic...

- **Reductions Techniques:** Constraint Propagation Algorithms, ...

Extensions:

- **Heuristic:** Limitation of the Memory (Polynomial Algorithm but Not global method - +Ninin)

- **Computation of Bounds:** Automatic Linear Reformulation Techniques based on Affine Arithmetic (Solve a Linear Program by iteration +Ninin) (Csendes, Kearfott [GlobSol], Ratchek, Rokne, E. Hansen...)
Outline

IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Design of Actuators
MAPSE
Combinatorial Models
Numerical Examples
Extension to Black-Box Constraint
Realizations & Conclusion

Interval Branch and Bound Algorithms
Interval Analysis
Interval methods for bounds
Branch and Bound Algorithms
Propagation Techniques
Algorithms for Mixed Problems
IBBA

Design of Electromechanical Actuators
Rotating Machines with Magnetic Effects
Combinatorial Models for Electrical Machines
Numerical Examples of Electrical Machines

Design with a Black-Box Constraint
Numerical Validations
IBBA+NUMT
Numerical Examples

Some Realizations and Conclusion
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Massine

IBBA Algorithms

MAPSE

Combinatorial Models

Numerical Examples

Extension to Black-Box Constraint Realizations & Conclusion

Direct and Inverse Problem of Design

1 – Direct Problem of Design

2 – Inverse Problem of Dimensioning

3 – Inverse Problem of Design

- Direct Solve of the Maxwell’s Equations By Finite Element Methods

- Functions on the Maxwell’s Equations => Analytical Models

- Model associating many different elementary structures

Objectives:

- Min volume

- Min mass

General Model

CHARACTERISTIC VALUES FOR A GIVEN STRUCTURE
Rotating Machines with Magnetic Effects

• Criteria:

\[ V_{ap} = \pi \frac{D}{\lambda} (D + E - e - l_a)(2C + E + e + l_a) \]

\[ V_m = \pi \beta l_a \frac{D}{\lambda} (D - 2e - l_a) \]

\[ p_j = \pi \rho_c u \frac{D}{\lambda} (D + E) E_{ch} \]

Constraints:

\[ C_{em} = \frac{\pi}{2} \frac{1 - K_f}{\lambda} \sqrt{k, \beta E_{ch}} \frac{E}{D^2} (D + E) B_e \]

\[ E_{ch} = A J_{cu} = k, E J^2_{cu}, K_f \approx 1.5 p \beta \frac{e + E}{D}, B_e = \frac{2l_a p}{D \log \left( \frac{D + 2E}{D - 2(l_a + e)} \right)} \]

\[ C = \frac{\pi B_e}{4 p B_{fer}} D, p = \frac{\pi D}{A_p} e_{\min} - e \leq 0, K_f - K_{f_{\text{max}}} \leq 0 \]
Example for the Dimensioning of an Electrical Motor

Electrical Slotless Rotating Machines with Permanent Magnet:

- IBBA standard (defined by Ratschek and Rokne 1988) \(\rightarrow 1h35\),
- IBBA + propagation due to E. Hansen \(\rightarrow 41.5s\),
- IBBA + propagation with the calculus tree \(\rightarrow 0.5s\).
Example for the Dimensioning of an Electrical Motor

Electrical Slotless Rotating Machines with Permanent Magnet:

- IBBA standard (defined by Ratschek and Rokne 1988) \(\rightarrow 1\text{h35},\)
- IBBA + propagation due to E. Hansen \(\rightarrow 41.5\text{s},\)
- IBBA + propagation with the calculus tree \(\rightarrow 0.5\text{s}.\)
Example for the Dimensioning of an Electrical Motor

Electrical Slotless Rotating Machines with Permanent Magnet:

▶ IBBA standard (defined by Ratschek and Rokne 1988) \( \rightarrow 1h35 \),
▶ IBBA + propagation due to E. Hansen \( \rightarrow 41.5s \),
▶ IBBA + propagation with the calculus tree \( \rightarrow 0.5s \).
Combination of Different Rotating Electrical Machines

Figure: 4 structures possible machines × 2 modes (rectangular or sinusoidal waveform).
Discrete Variables for Modeling Electrical Machines

1. $b_r = 1$ for machines with an internal rotoric configuration and $b_r = 0$ for an external one,

2. $b_e = 1$ for machines with slots or $b_e = 0$ slotless machines,

3. $b_f = 1$ represents rectangular waveform or $b_f = 0$ for a sinusoidal one.

3 boolean variables to represent 8 possible structures + 2 categorical variables.
Combinatorial Models for Electrical Machines

\[ \Gamma_{em} = k_\Gamma D [D + (1 - b_e)(2b_r - 1)E] L B_e K_S, \]
\[ K_S = k_r E j \left( b_e \frac{a}{a + d} + (1 - b_e) \right), \]
\[ k_\Gamma = \frac{\pi}{2} \left[ b_f [1 - K_f] \sqrt{\beta} + (1 - b_f) \frac{\sqrt{2}}{2} \sin(\beta \frac{\pi}{2}) \right], \]
\[ K_f = 1.5p \beta \left[ \frac{E + g}{D} \right] (1 - b_e) . b_f, \]
\[ B_e = \frac{2 J (\sigma_m) l_a}{(2b_r - 1) D \ln \left[ \frac{D + 2E(2b_r - 1)(1 - b_e)}{D - 2(2b_r - 1)[l_a + g]} \right]} \frac{1}{k_c}, \]
\[ k_c = \frac{1}{1 - b_e \left[ \frac{N_e a^2}{5\pi D . g + \pi D . a} \right]}, \]

Generally, the torque \( \Gamma_{em} \) is fixed \( \Rightarrow \) a strong equality.
Examples of 4 optimal machines with magnetical effects
Outline

IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Design of Actuators
Extension to Black-Box Constraint
Numerical Validations
IBBA+NUMT
Numerical Examples
Realizations & Conclusion

Interval Branch and Bound Algorithms
Interval Analysis
Interval methods for bounds
Branch and Bound Algorithms
Propagation Techniques
Algorithms for Mixed Problems
IBBA

Design of Electromechanical Actuators
Rotating Machines with Magnetic Effects
Combinatorial Models for Electrical Machines
Numerical Examples of Electrical Machines

Design with a Black-Box Constraint
Numerical Validations
IBBA+NUMT
Numerical Examples

Some Realizations and Conclusion
1 – Direct Problem of Design

Direct Solve of the Maxwell’s Equations By Finite Element Methods

CHARACTERISTICAL VALUES FOR A GIVEN STRUCTURE

2 – Inverse Problem of Dimensioning

Analytical Model of the given structure

Functions:
- min mass
- ... 
- min volume

Some assumptions on the Maxwell’s Equations => Analytical Models (simple)

3 – Inverse Problem of design

Type of structure, dimensions and constitutions

Objectives:
- min masse
- ... 
- min volume

Model associating many different elementary structures : General Model
Direct and Inverse Problem of Design

1 – Direct Problem of Design

Direct Solve of the Maxwell’s Equations By Finite Element Methods

CHARACTERISTIC VALUES FOR A GIVEN STRUCTURE

2 – Inverse Problem of Dimensioning

Analytical Model of the given structure

Functions:
\[ \text{min mass} \]
\[ \ldots \]
\[ \text{min volume} \]

Some assumptions on the Maxwell’s Equations => Analytical Models (simple)

3 – Inverse Problem of design

Type of structure, dimensions and constitutions

Objectives:
\[ \text{min masse} \]
\[ \ldots \]
\[ \text{min volume} \]

Model associating many different elementary structures:

General Model
Extension to BB constraint: Analytical versus Numerical Models

Schedule of conditions ➞ a fixed torque.

Optimal solutions satisfy the equality constraint about 1%.

Computation of the torque using finite elements methods (by a numerical model); software EFCAD (from the LEEI) or ANSYS.

EFCAD and ANSYS are too general ➞ NUMT
Extension to BB constraint: Numerical Validations

**Figure:** Draw 2 optimal solutions (min mass and min multicriteria).
Extension to BB constraint: Numerical Validations

Figure: Torque of 3 solutions and design of teeth of the slot.

Using Triangle and EFCAD.

Name of the Software: NUMT.
Extension to BB constraint: Discussions

Analytical Value ≠ Numerical Value

Generally about 10% !

- If the gap is less than 3% our solution is validated.
- Else $\Rightarrow$ modifications of the solution until the numerical value is correct.

Optimize with a black-box constraint:

$$\min_{x \in \mathbb{R}^{nr}, z \in \mathbb{N}^{ne}, \sigma \in \prod_{i=1}^{nc} K_i, b \in B^{nb}} f(x, z, \sigma, b)$$

$$g_i(x, z, \sigma, b) \leq 0 \ \forall i \in \{1, \ldots, n_g\}$$

$$h_j(x, z, \sigma, b) = 0 \ \forall j \in \{1, \ldots, n_h - 1\}$$

$$NUMT(x, z, \sigma, b) = \Gamma_{em}$$

$$NUMT(x, z, \sigma, b) = \Gamma_{em} \rightarrow NUMT(x, z, \sigma, b) \geq \Gamma_{em}$$
Analytical Value ≠ Numerical Value

Generally about 10% !

- If the gap is less than 3% our solution is validated.
- Else → modifications of the solution until the numerical value is correct.

Optimize with a black-box constraint:

\[
\begin{align*}
\min_{x \in \mathbb{R}^{n_r}, z \in \mathbb{N}^{n_e},} & \quad f(x, z, \sigma, b) \\
\text{s.t.} \quad & g_i(x, z, \sigma, b) \leq 0 \quad \forall i \in \{1, \ldots, n_g\} \\
& h_j(x, z, \sigma, b) = 0 \quad \forall j \in \{1, \ldots, n_h - 1\} \\
& \text{NUMT}(x, z, \sigma, b) = \Gamma_{em}
\end{align*}
\]

\[
\text{NUMT}(x, z, \sigma, b) = \Gamma_{em} \implies \text{NUMT}(x, z, \sigma, b) \geq \Gamma_{em}
\]
Extension to BB constraint: Discussions

Analytical Value ≠ Numerical Value

Generally about 10% !

- If the gap is less than 3% our solution is validated.
- Else $\Rightarrow$ modifications of the solution until the numerical value is correct.

Optimize with a black-box constraint:

$$\min_{x \in \mathbb{R}^{nr}, z \in \mathbb{N}^{ne}, \sigma \in \prod_{i=1}^{nc} K_i, b \in B^{nb}} f(x, z, \sigma, b)$$

$$g_i(x, z, \sigma, b) \leq 0 \ \forall i \in \{1, \ldots, n_g\}$$

$$h_j(x, z, \sigma, b) = 0 \ \forall j \in \{1, \ldots, n_h - 1\}$$

$$\text{NUMT}(x, z, \sigma, b) = \Gamma_{em}$$

$$\text{NUMT}(x, z, \sigma, b) = \Gamma_{em} \rightarrow \text{NUMT}(x, z, \sigma, b) \geq \Gamma_{em}$$
Extension to BB constraint: IBBA+NUMT

Idea: Define a zone where the numerical solution is searched.
Analytical model + tolerance about 10% $\implies$ zone.
In this zone, all the solutions satisfies that
$NUMT(x^*, z^*, \sigma^*, b^*) = \Gamma_{em}$ about 2%.

Minimize $f(x, z, \sigma, b)$
subject to
$g_i(x, z, \sigma, b) \leq 0 \quad \forall i \in \{1, \ldots, n_g\}$
$h_j(x, z, \sigma, b) = 0 \quad \forall j \in \{1, \ldots, n_h - 1\}$
$(1 - pc) \times \Gamma_{em} \leq \Gamma(x, z, \sigma, b) \leq (1 + pc) \times \Gamma_{em}$
$NUMT(x, z, \sigma, b) = \Gamma_{em}$

where $pc$ is a percentage.
Extension to BB constraint: IBBA+NUMT

Idea: Define a zone where the numerical solution is searched. Analytical model + tolerance about 10% $\implies$ zone. In this zone, all the solutions satisfies that $\text{NUMT}(x^*, z^*, \sigma^*, b^*) = \Gamma_{em}$ about 2%.

\[
\begin{aligned}
\min_{x \in \mathbb{R}^{nr}, z \in \mathbb{N}^{ne}, \\ \sigma \in \prod_{i=1}^{nc} K_i, b \in B^{nb}} & f(x, z, \sigma, b) \\
\text{subject to:} & \\
& g_i(x, z, \sigma, b) \leq 0 \ \forall i \in \{1, \ldots, n_g\} \\
& h_j(x, z, \sigma, b) = 0 \ \forall j \in \{1, \ldots, n_h - 1\} \\
& (1 - pc) \times \Gamma_{em} \leq \Gamma(x, z, \sigma, b) \leq (1 + pc) \times \Gamma_{em} \\
& \text{NUMT}(x, z, \sigma, b) = \Gamma_{em} \end{aligned}
\]

where $pc$ is a percentage.
IBBA: an Exact Global Optimization Software for the Design of Electromechanical Actuators

Frédéric Messine

IBBA Algorithms
Design of Actuators
Extension to Black-Box Constraint
Numerical Validations
IBBA+NUMT
Numerical Examples
Realizations & Conclusion

Extension to BB constraint: Algorithm IBBA+NUMT

1. Set $X :=$ the initial hypercube.
2. Set $\tilde{f} := +\infty$ and set $\mathcal{L} := (+\infty, X)$.
3. Extract from $\mathcal{L}$ the lowest lower bound.
4. **Bisect the considered box** chosen by its midpoint: $V_1, V_2$.
5. For $j := 1$ to 2 do
   5.1 **Compute** $v_j := \text{lb}(f, V_j)$.
   5.2 **Compute all the lower and upper bounds** of all the analytical constraints on $V_j$ + some deduction steps.
   5.3 if $\tilde{f} \geq v_j$ and no analytical constraint is unsatisfied then
      ▶ insert $(v_j, V_j)$ in $\mathcal{L}$.
      ▶ set $m$ the midpoint of $V_j$
      ▶ if $m$ satisfies all the analytical constraints and then if the numerical constraint $\text{NUMT}(x, z, \sigma, b) = \Gamma$ is also satisfied then
        $\tilde{f} := \min(\tilde{f}, f(m))$.
      ▶ if $\tilde{f}$ is changed then remove from $\mathcal{L}$ all $(z, Z)$ where $z > \tilde{f}$ and set $\tilde{y} := m$.
6. If $\tilde{f} - \min\limits_{(z, Z) \in \mathcal{L}} z < \epsilon$ (where $z = \text{lb}(f, Z)$) then STOP.
   Else GoTo Step 4.
## Extension to BB constraint: Numerical Example 1 with IBBA+NUMT

<table>
<thead>
<tr>
<th>Name</th>
<th>Bounds</th>
<th>Unit</th>
<th>IBBA</th>
<th>IBBA+NUMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>[0.01, 0.3] m</td>
<td>0.1331</td>
<td>0.1310</td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>[0.01, 0.3] m</td>
<td>0.0474</td>
<td>0.0497</td>
<td></td>
</tr>
<tr>
<td>$l_a$</td>
<td>[0.003, 0.01] m</td>
<td>0.0047</td>
<td>0.0047</td>
<td></td>
</tr>
<tr>
<td>$E$</td>
<td>[0.005, 0.03] m</td>
<td>0.0074</td>
<td>0.0074</td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>[0.003, 0.02] m</td>
<td>0.0049</td>
<td>0.0049</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>[0.7, 0.9]</td>
<td>0.89</td>
<td>0.89</td>
<td></td>
</tr>
<tr>
<td>$k_d$</td>
<td>[0.4, 0.6]</td>
<td>0.5043</td>
<td>0.5043</td>
<td></td>
</tr>
<tr>
<td>$p$</td>
<td>[3, 10]</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>$m$</td>
<td>{1, 2}</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$b_r$</td>
<td>{0, 1}</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Min $V_g$</th>
<th>IBBA</th>
<th>IBBA+NUMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>$m^3$</td>
<td>8.881 $10^{-4}$</td>
<td>9.072 $10^{-4}$</td>
</tr>
<tr>
<td>Mass</td>
<td>kg</td>
<td>3.21</td>
<td>3.31</td>
</tr>
<tr>
<td>Multi</td>
<td></td>
<td>2.09</td>
<td>2.15</td>
</tr>
<tr>
<td>Analytical Torque</td>
<td>N·m</td>
<td>9.81</td>
<td>10.00</td>
</tr>
<tr>
<td>Numerical Torque</td>
<td>N·m</td>
<td>9.35</td>
<td>9.96</td>
</tr>
<tr>
<td>CPU - Time</td>
<td>min</td>
<td>0min35s</td>
<td>7min15s</td>
</tr>
<tr>
<td>Numerical Computations</td>
<td></td>
<td>-</td>
<td>437</td>
</tr>
</tbody>
</table>
### Extension to BB constraint: Numerical Example 2 with IBBA+NUMT

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Bounds</th>
<th>Unit</th>
<th>Min $M_a$ IBBA</th>
<th>Min $M_a$ IBBA+NUMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>[0.01, 0.3]</td>
<td>m</td>
<td>0.1400</td>
<td>0.1400</td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>[0.01, 0.3]</td>
<td>m</td>
<td>0.0496</td>
<td>0.0519</td>
<td></td>
</tr>
<tr>
<td>$I_a$</td>
<td>[0.003, 0.01]</td>
<td>m</td>
<td>0.0039</td>
<td>0.0039</td>
<td></td>
</tr>
<tr>
<td>$E$</td>
<td>[0.005, 0.03]</td>
<td>m</td>
<td>0.0074</td>
<td>0.0075</td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>[0.003, 0.02]</td>
<td>m</td>
<td>0.0039</td>
<td>0.0039</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>[0.7, 0.9]</td>
<td></td>
<td>0.74</td>
<td>0.71</td>
<td></td>
</tr>
<tr>
<td>$k_d$</td>
<td>[0.4, 0.6]</td>
<td></td>
<td>0.4978</td>
<td>0.5022</td>
<td></td>
</tr>
<tr>
<td>$p$</td>
<td>[3, 10]</td>
<td></td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>$m$</td>
<td>{1, 2}</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$b_r$</td>
<td>{0, 1}</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

| Volume | $m^3$ | $9.716 \times 10^{-4}$ | $10.157 \times 10^{-4}$ |
| Mass   | $kg$  | 2.94            | 3.07            |
| Multi  |       | 2.10            | 2.19            |
| Analytical Torque | N·m | 9.82            | 10.21            |
| Numerical Torque   | N·m  | 9.26            | 9.86             |
| CPU - Time   | min  | 1min14s         | 8min17s         |
| Numerical Computations |       | -               | 560              |
## Extension to BB constraint: Numerical Example 3 with IBBA+NUMT

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Bounds</th>
<th>Unit</th>
<th>IBBA</th>
<th>IBBA+NUMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>$D$</td>
<td>[0.01, 0.3]</td>
<td>m</td>
<td>0.1400</td>
<td>0.1310</td>
</tr>
<tr>
<td>L</td>
<td>$L$</td>
<td>[0.01, 0.3]</td>
<td>m</td>
<td>0.0451</td>
<td>0.0497</td>
</tr>
<tr>
<td>$l_a$</td>
<td>$l_a$</td>
<td>[0.003, 0.01]</td>
<td>m</td>
<td>0.0039</td>
<td>0.0047</td>
</tr>
<tr>
<td>E</td>
<td>$E$</td>
<td>[0.005, 0.03]</td>
<td>m</td>
<td>0.0074</td>
<td>0.0073</td>
</tr>
<tr>
<td>C</td>
<td>$C$</td>
<td>[0.003, 0.02]</td>
<td>m</td>
<td>0.0050</td>
<td>0.0049</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$\beta$</td>
<td>[0.7, 0.9]</td>
<td></td>
<td>0.89</td>
<td>0.89</td>
</tr>
<tr>
<td>$k_d$</td>
<td>$k_d$</td>
<td>[0.4, 0.6]</td>
<td></td>
<td>0.5043</td>
<td>0.4957</td>
</tr>
<tr>
<td>$p$</td>
<td>$p$</td>
<td>$[3, 10]$</td>
<td></td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>$m$</td>
<td>$m$</td>
<td>${1, 2}$</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$b_r$</td>
<td>$b_r$</td>
<td>${0, 1}$</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Volume</td>
<td></td>
<td></td>
<td>$m^3$</td>
<td>9.067 $10^{-4}$</td>
<td>9.072 $10^{-4}$</td>
</tr>
<tr>
<td>Mass</td>
<td></td>
<td></td>
<td>$kg$</td>
<td>3.10</td>
<td>3.30</td>
</tr>
<tr>
<td>Multi</td>
<td></td>
<td></td>
<td></td>
<td>2.07</td>
<td>2.14</td>
</tr>
<tr>
<td>Analytical Torque</td>
<td></td>
<td></td>
<td>$N\cdot m$</td>
<td>9.86</td>
<td>9.93</td>
</tr>
<tr>
<td>Numerical Torque</td>
<td></td>
<td></td>
<td>$N\cdot m$</td>
<td>9.06</td>
<td>9.96</td>
</tr>
<tr>
<td>CPU - Time</td>
<td></td>
<td></td>
<td>min</td>
<td>1min03s</td>
<td>7min37s</td>
</tr>
<tr>
<td>Numerical Computations</td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>526</td>
</tr>
</tbody>
</table>
## Extension to BB constraint: Impact of the zone

<table>
<thead>
<tr>
<th>zone: (pc)</th>
<th>Mass</th>
<th>(\Gamma_{em})</th>
<th>NUMT</th>
<th>Time</th>
<th>Its</th>
<th>Its of NUMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBBA</td>
<td>2.92</td>
<td>9.81</td>
<td>9.12</td>
<td>0'51</td>
<td>152,126</td>
<td>–</td>
</tr>
<tr>
<td>2%</td>
<td>3.44</td>
<td>10.18</td>
<td>9.92</td>
<td>7'59</td>
<td>223,769</td>
<td>585</td>
</tr>
<tr>
<td>5%</td>
<td>3.35</td>
<td>10.20</td>
<td>9.83</td>
<td>17'08</td>
<td>213,094</td>
<td>1,404</td>
</tr>
<tr>
<td>10%</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>34'44</td>
<td>216,623</td>
<td>2,898</td>
</tr>
<tr>
<td>20%</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>79'29</td>
<td>223,118</td>
<td>7,162</td>
</tr>
<tr>
<td>30%</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>159'25</td>
<td>228,324</td>
<td>14,751</td>
</tr>
<tr>
<td>40%</td>
<td>=</td>
<td>=</td>
<td>=</td>
<td>281'09</td>
<td>231,513</td>
<td>25,004</td>
</tr>
</tbody>
</table>
Conclusion

- IBBA is an efficient tool. Specially dedicated to solve some designs of electromechanical actuators.
- Extension to consider constraints of Black-Box type.
- IBBA is still in progress: improve the computation of bounds by linear relaxations based on affine arithmetic (J. Ninin) and propagation techniques.
Some Realizations

**Figure:** Motor / Transformer.

**Figure:** Design of piezoelectric bimorphs.