

Introduction of auxiliary variables into an interval function for improving its monotonicity-based evaluation

Ignacio Araya, Bertrand Neveu, Gilles Trombettoni

INRIA, Université de Nice-Sophia, CERTIS, France

SWIM, Juin 2009, Lausanne, Switzerland

Outline

- 1 Introduction
- 2 Occurrence Grouping
- 3 An Efficient OG Algorithm
- 4 Experiments

Interval Analysis

- An **interval** $[x] = [a; b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$ represents the domain of a variable x .
- A **box** $[B]$ is a vector of intervals defining the search space.
- Basic interval operators for $[x] = [a; b]$ and

$$[y] = [c; d].$$

$$[x] \mathbf{op} [y] = \{x \mathbf{op} y \mid x \in [x], y \in [y]\}$$

$$[x] + [y] = [a + c; b + d]$$

$$[x] - [y] = [a - d; b - c]$$

$$[x] \times [y] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$[x]/[y] = [a; b] \times [1/d, 1/c], 0 \notin [y].$$

Natural Interval Extension

- An **interval extension** $[f]([x], [y], \dots)$ encloses all the values of $f(x, y, \dots)$ for $\{x, y, \dots\} \in \{[x], [y], \dots\}$, that is:

$$[f]([\mathbf{B}]) \supseteq \{f(\mathbf{X}) / \mathbf{X} \in [\mathbf{B}]\}$$

- The **natural interval extension** of a function f is obtained replacing each variable for its domain and applying interval arithmetics. For example:

$$f(x, y) = x^2 - x + y \text{ and } [x] = [1; 5], [y] = [-3; 4].$$

The natural extension of f is:

$$[f]([x], [y]) = [1; 5]^2 - [1; 5] + [-3; 4] = [-7; 28]$$

- If a variable occurs more than once in a function, the natural extension **overestimates** the true range (**dependency problem**). The true range of f is $[-3; 24]$.

Interval Extension using monotonicity

- The monotonicity of a function f in $[\mathbf{B}]$ can be used to compute sharper interval extensions of f eliminating the dependency problem in the **monotonic variables** (x s.t. $\frac{\partial f}{\partial x} \geq 0$ or $\frac{\partial f}{\partial x} \leq 0$, $\forall \mathbf{X} \in [\mathbf{B}]$).
- Consider the function $f(\mathbf{X}) = f(x_1, \dots, x_k)$
 - x_1, \dots, x_i are **monotonic increasing**.
 - x_{i+1}, \dots, x_j are **monotonic decreasing**.
 - The **evaluation by monotonicity** of f in the box $[\mathbf{B}] = \{[x_1], \dots, [x_k]\}$ is given by:

$$f_{min} = \underline{f}(\underline{x}_1, \dots, \underline{x}_i, \overline{x}_{i+1}, \dots, \overline{x}_j, [x_{j+1}], \dots, [x_k])$$

$$f_{max} = \overline{f}(\overline{x}_1, \dots, \overline{x}_i, \underline{x}_{i+1}, \dots, \underline{x}_j, [x_{j+1}], \dots, [x_k])$$

$$[f]_M([\mathbf{B}]) = [f_{min}, f_{max}]$$

Evaluation by monotonicity

- Consider the function $f(\mathbf{x}, \mathbf{y}) := \mathbf{x}^2 - \mathbf{x} + \mathbf{y}$ with domains $[\mathbf{x}] = [1; 5]$ and $[\mathbf{y}] = [-3; 4]$.
- The **natural extension** $[f]$ of f is:

$$[f]([1; 5], [-3; 4]) = [1; 5]^2 - [1; 5] + [-3; 4] = [-7; 28]$$

- The **evaluation by monotonicity** $[f]_M$ of f :

$$f_{min} = \underline{[f]}(1, -3) = -3$$

$$f_{max} = \overline{[f]}(5, 4) = 24$$

$$[f]_M([1; 5], [-3; 4]) = [-3; 24]$$

Evaluation by monotonicity

- The evaluation by monotonicity is **sharper than** (or equivalent to) the natural extension if at least one variable is monotonic and has multiple occurrences.

Contributions

- Improving the evaluation by monotonicity $[f]_M([\mathbf{B}])$ when a variable x is **not monotonic** in $[\mathbf{B}]$ using a so-called **Occurrence Grouping (OG)**:
 - Function f is transformed in f^* .
 - f^* replace all the occurrences of x in f by auxiliary variables: x_a, x_b or x_c .
 - x_a and x_b **are monotonic variables** in f^*
→ $[f^*]_M([\mathbf{B}^*]) \subseteq [f]_M([\mathbf{B}])$.
- Finding a OG that minimizes a Taylor-based estimation of $W([f^*]_M)$.

Outline

- 1 Introduction
- 2 Occurrence Grouping**
- 3 An Efficient OG Algorithm
- 4 Experiments

Occurrence Grouping (Example 1/2)

- Consider the function $f(\mathbf{x}) = \mathbf{x}^2 + \mathbf{x}^3 + 7\mathbf{x} + \mathbf{x}^4 - 2\mathbf{x}$ with $[\mathbf{x}] = [-1; 1] \Rightarrow x$ is not monotonic in $[\mathbf{B}]$.
- The **OG** method consists in transforming $f(x)$ in a new function $f^*(x_a, x_b, x_c)$, replacing the occurrences of x by x_a, x_b or x_c , s.t. $\frac{\partial f}{\partial x_a} \geq 0, \frac{\partial f}{\partial x_b} \leq 0, \forall \{x_a, x_b, x_c\} \in [\mathbf{B}^*]$ ($[g](x_a) \geq 0, [g](x_b) \leq 0$) and $[x_a] = [x_b] = [x_c] = [x]$.
- A possible OG for f :

$$f^* = \mathbf{x}_b^2 + \mathbf{x}_c^3 + 7\mathbf{x}_a + \mathbf{x}_a^4 - 2\mathbf{x}_b$$

where $[g](\mathbf{x}_a) = [3; 11] \geq \mathbf{0}$ and $[g](\mathbf{x}_b) = [-4; 0] \leq \mathbf{0}$

Occurrence Grouping (Example 2/2)

$$f = x^2 + x^3 + 7x + x^4 - 2x$$
$$f^* = x_b^2 + x_c^3 + 7x_a + x_a^4 - 2x_b$$

- The natural extension $f[\mathbf{B}]$ is $[-10; 12]$.
- The evaluation by monotonicity $f^*[\mathbf{B}^*]$ is $[-8; 12]$.

Occurrence Grouping

Proposition

Let f^* be the result of applying an OG procedure to a function f for one non-monotonic variable. Then:
the evaluation by monotonicity of f^* is sharper than the evaluation by monotonicity of f .

$$[f^*]_M \subseteq [f]_M \subseteq [f]$$

The Optimization Problem

- The objective of OG is to sharpen, as much as possible, the evaluation by monotonicity of a function f in a box $[\mathbf{B}]$.
- \rightarrow We want to **minimize the width(W)** of the interval: $[f^*]_M$.
- Using Taylor series we can represent an **overestimation** of $[f^*]_M$ as an interval $[W_{f^*}] \supseteq W$:

Overestimation of the width of $[f^*]_M$

$$[W_{f^*}] = \sum_{i=1}^k \left[w([x_i]) \times \left([g](x_{i_a}) - [g](x_{i_b}) + \sum_{j=1}^{\#x_{i_c}} |[g](x_{i_c}^{(j)})| \right) \right]$$

$[g](x_{i_z})$ is the interval gradient of x_{i_z} in $[\mathbf{B}]$.

Observations

- $[W_{f^*}]$ is an interval and overestimates the width W of $[f^*]_M$.
- A good **heuristic approach** to minimize W is minimizing $\overline{W_{f^*}}$ and $\underline{W_{f^*}}$ at the same time.
- The optimal grouping is **independent for each variable** (every term in the sum is related to only one variable).
- The minimization of $\overline{W_{f^*}}$ and $\underline{W_{f^*}}$ can be reduced to the minimization of \overline{G} and \underline{G} for each variable $x \in \mathbf{X}$, where:

$$\overline{G} = \left(\overline{[g]}(x_a) - \underline{[g]}(x_b) + \sum_{j=1}^{\#x_c} |[g](x_c^{(j)})|_{max} \right)$$

$$\underline{G} = \left(\underline{[g]}(x_a) - \overline{g}([x_b]) + \sum_{j=1}^{\#x_c} |[g](x_c^{(j)})|_{min} \right)$$

The Optimization Problem

Minimize: $\overline{G}, \underline{G}$

$$\overline{G} = \left(\overline{[g]}(x_a) - \underline{[g]}(x_b) + \sum_{j=1}^{\#x} r_{c_j} \cdot |[g](x_j)|_{max} \right)$$

$$\underline{G} = \left(\underline{[g]}(x_a) - \overline{[g]}(x_b) + \sum_{j=1}^{\#x} |[g](x_j)|_{min} \right)$$

subject to:

$$[g](x_a) = \sum_{j=1}^{\#x} r_{a_j} \cdot [g](x_j) \geq 0, \quad [g](x_b) = \sum_{j=1}^{\#x} r_{b_j} \cdot [g](x_j) \leq 0$$

$$r_{a_j} + r_{b_j} + r_{c_j} = 1$$

$$r_{a_j}, r_{b_j}, r_{c_j} = \{0, 1\}$$

The Optimization Problem

- The problem is **discrete and likely NP-hard**: each occurrence of x is replaced by x_a, x_b **or** x_c .
- If we replace each variable x in \mathbf{X} by $r_a \cdot x_a + r_b \cdot x_b + r_c \cdot x_c$ in f^* ($r_a, r_b, r_c \geq 0$ and $r_a + r_b + r_c = 1$):
 - Natural extensions are equivalents: $[f] = [f^*]$.
 - The new problem is **continuous and tractable**: each occurrence of x is split into 3 parts that are replaced by x_a, x_b **and** x_c .
 - The relaxed problem has lower minimums.

The Optimization Problem

Minimize: $\overline{G}, \underline{G}$

$$\overline{G} = \left(\overline{[g]}(x_a) - \underline{[g]}(x_b) + \sum_{j=1}^{\#x} r_{c_j} \cdot |[g](x_j)|_{max} \right)$$

$$\underline{G} = \left(\underline{[g]}(x_a) - \overline{[g]}(x_b) + \sum_{j=1}^{\#x} |[g](x_j)|_{min} \right)$$

subject to:

$$[g](x_a) = \sum_{j=1}^{\#x} r_{a_j} \cdot [g](x_j) \geq 0, \quad [g](x_b) = \sum_{j=1}^{\#x} r_{b_j} \cdot [g](x_j) \leq 0$$

$$r_{a_j} + r_{b_j} + r_{c_j} = 1$$

$$r_{a_j}, r_{b_j}, r_{c_j} \geq 0$$

The Optimization Problem

Proposition

For any variable x , **there exists a grouping** $\{x_a, x_b, x_c\}$ such that: $\overline{G}(x_a, x_b, x_c) \leq \overline{G}(x_{a_1}, x_{b_1}, x_{c_1})$ and $\underline{G}(x_a, x_b, x_c) \leq \underline{G}(x_{a_2}, x_{b_2}, x_{c_2})$ for all groupings $\{x_{a_1}, x_{b_1}, x_{c_1}\}$ and $\{x_{a_2}, x_{b_2}, x_{c_2}\}$. Where,

$$\overline{G}(x_a, x_b, x_c) = \left(\overline{[g]}(x_a) - \underline{[g]}(x_b) + \sum_{j=1}^{\#x_c} |[g](x_c^{(j)})|_{max} \right)$$

$$\underline{G}(x_a, x_b, x_c) = \left(\underline{[g]}(x_a) - \overline{[g]}(x_b) + \sum_{j=1}^{\#x_c} |[g](x_c^{(j)})|_{min} \right)$$

subject to the previous constraints.

Outline

- 1 Introduction
- 2 Occurrence Grouping
- 3 An Efficient OG Algorithm**
- 4 Experiments

Description of the algorithm

- Our algorithm finds a grouping that minimizes: $\overline{G}(x_a, x_b, x_c)$ and $\underline{G}(x_a, x_b, x_c)$ **at the same time**.
- Let \mathbf{x}_m and \mathbf{x}_n be the sets of monotonic and non monotonic occurrences of a variable x resp. There exists 3 possible cases:
 - $[g](x) > 0$ (or $[g](x) < 0$): the variable is monotone \Rightarrow all the occurrences of x are replaced by x_a (or x_b).
 - $[g](\mathbf{x}_m) \ni 0$: the variable is non monotone, each occurrence in \mathbf{x}_m is replaced by $r.x_a + (1 - r).x_b$.
 - $[g](\mathbf{x}_m) > 0$ (or $[g](\mathbf{x}_m) < 0$) and $[g](x) \ni 0$: the variable is non monotone, all the occurrences in \mathbf{x}_m and a part of the occurrences in \mathbf{x}_n are replaced by x_a (or x_b).

Description of the algorithm

- Our algorithm finds a grouping that minimizes: $\overline{G}(x_a, x_b, x_c)$ and $\underline{G}(x_a, x_b, x_c)$ **at the same time**.
- Let \mathbf{x}_m and \mathbf{x}_n be the sets of monotonic and non monotonic occurrences of a variable x resp. There exists 3 possible cases:
 - $[g](x) > 0$ (or $[g](x) < 0$): the variable is monotone \Rightarrow all the occurrences of x are replaced by x_a (or x_b).
 - $[g](\mathbf{x}_m) \ni 0$: the variable is non monotone, each occurrence in \mathbf{x}_m is replaced by $r.x_a + (1 - r).x_b$.
 - $[g](\mathbf{x}_m) > 0$ (or $[g](\mathbf{x}_m) < 0$) and $[g](x) \ni 0$: the variable is non monotone, all the occurrences in \mathbf{x}_m and a part of the occurrences in \mathbf{x}_n are replaced by x_a (or x_b).

Description of the algorithm

- Our algorithm finds a grouping that minimizes: $\overline{G}(x_a, x_b, x_c)$ and $\underline{G}(x_a, x_b, x_c)$ **at the same time**.
- Let \mathbf{x}_m and \mathbf{x}_n be the sets of monotonic and non monotonic occurrences of a variable x resp. There exists 3 possible cases:
 - $[g](x) > 0$ (or $[g](x) < 0$): the variable is monotone \Rightarrow all the occurrences of x are replaced by x_a (or x_b).
 - $[g](\mathbf{x}_m) \ni 0$: the variable is non monotone, each occurrence in \mathbf{x}_m is replaced by $r.x_a + (1 - r).x_b$.
 - $[g](\mathbf{x}_m) > 0$ (or $[g](\mathbf{x}_m) < 0$) and $[g](x) \ni 0$: the variable is non monotone, all the occurrences in \mathbf{x}_m and a part of the occurrences in \mathbf{x}_n are replaced by x_a (or x_b).

Description of the algorithm

- Our algorithm finds a grouping that minimizes: $\overline{G}(x_a, x_b, x_c)$ and $\underline{G}(x_a, x_b, x_c)$ **at the same time**.
- Let \mathbf{x}_m and \mathbf{x}_n be the sets of monotonic and non monotonic occurrences of a variable x resp. There exists 3 possible cases:
 - $[g](x) > 0$ (or $[g](x) < 0$): the variable is monotone \Rightarrow all the occurrences of x are replaced by x_a (or x_b).
 - $[g](\mathbf{x}_m) \ni 0$: the variable is non monotone, each occurrence in \mathbf{x}_m is replaced by $r.x_a + (1 - r).x_b$.
 - $[g](\mathbf{x}_m) > 0$ (or $[g](\mathbf{x}_m) < 0$) **and** $[g](x) \ni 0$: the variable is non monotone, all the occurrences in \mathbf{x}_m and a part of the occurrences in \mathbf{x}_n are replaced by x_a (or x_b).

Case: $[g](\mathbf{x}_m) \ni 0$

$$1: [\underline{g}_{m^+}, \overline{g}_{m^+}] \leftarrow [g](\mathbf{x}_{m^+})$$

$$2: [\underline{g}_{m^-}, \overline{g}_{m^-}] \leftarrow [g](\mathbf{x}_{m^-})$$

3:

$$4: r_1 \leftarrow \frac{\underline{g}_{m^+} \cdot \overline{g}_{m^-} + \overline{g}_{m^-} \cdot \underline{g}_{m^-}}{\underline{g}_{m^+} \cdot \overline{g}_{m^-} - \underline{g}_{m^-} \cdot \overline{g}_{m^+}}$$

$$5: r_2 \leftarrow \frac{\underline{g}_{m^+} \cdot \overline{g}_{m^+} + \overline{g}_{m^-} \cdot \underline{g}_{m^+}}{\underline{g}_{m^+} \cdot \overline{g}_{m^-} - \underline{g}_{m^-} \cdot \overline{g}_{m^+}}$$

6:

7: **for all** $x \in \mathbf{x}_{m^+}$ **do** $x \leftarrow (1 - r_1) \cdot x_a + r_1 \cdot x_b$ **end for**

8: **for all** $x \in \mathbf{x}_{m^-}$ **do** $x \leftarrow r_2 \cdot x_a + (1 - r_2) \cdot x_b$ **end for**

9: **for all** $x \in \mathbf{x}_n$ **do** $x \leftarrow x_c$ **end for**

Case: $[g](\mathbf{x}_m) > 0$ and $[g](x) \ni 0$

for all $x \in \mathbf{x}_m$ **do** $x \leftarrow x_a$ **end for**
 $\mathbf{x}_n \leftarrow \mathbf{AscOrder}(\mathbf{x}_n, f_{order} = -\overline{[g]}(x)/\underline{[g]}(x))$

for all $x \in \mathbf{x}_n$ **do**
 if $\underline{[g]}(x_a) + \underline{[g]}(x) \geq 0$ **then**
 $x \leftarrow x_a$
 else
 $r \leftarrow -\underline{[g]}(x_a)/\underline{[g]}(x)$
 $x \leftarrow r.x_a + (1 - r).x_c$
 end if
end for

Occurrence grouping v/s natural extension

Example:

- Consider $f(x) = x^2 + x^3 + 7x + x^4 - 2x$, $[x] = [-1; 1]$.
- The grouping algorithm convert f into:
$$f^* = x_a^2 + x_a^3 + 7x_a + \left(\frac{3}{4}x_a + \frac{1}{4}x_c\right)^4 - 2x_a$$
- The evaluation by monotonicity of f^* :
 $[f^*]_M = [-4.94; 8]$ that is 40% sharper than
 $[f] = [f]_M = [-10; 12]$

Outline

- 1 Introduction
- 2 Occurrence Grouping
- 3 An Efficient OG Algorithm
- 4 Experiments**

Experimentation

- Implementation in Ibex, C++.
- Interval solver using 3BCID + Interval Newton as contractors (3BCID).
- OG has been incorporated in two methods that use the evaluation by monotonicity.
 - Monotonicity test applied before the contractor (MonoTest).
 - MoHC, a monotonicity-based contractor, inside 3BCID (MoHC).

Benchs	3BCID	MonoTest		MoHC	
		\neg OG	OG	\neg OG	OG
brent-10	19.19 3923	18.92 3923	18.89 3923	19.03 3923	19.41 3923
caprasse	2.63 1309	2.67 1305	2.73 1309	3.59 1299	3.61 1145
hayes	40.95 17763	41.9 17763	41.63 17763	31.59 9627	31.23 9277
i5	56.7 10621	57.1 10621	55.85 10621	56.07 10543	53.72 10111
katsura	76.06 4251	81.33 4251	77.8 4251	81.52 4151	80.17 4097
kin1	1.9 87	1.95 87	1.96 87	1.95 87	1.96 87
eco9	13.77 6193	13.72 6193	13.85 6193	13.76 6183	14.17 6145
redeco8	6.28 2441	6.47 2441	6.28 2441	6.3 2437	6.47 2423
trigexp2-11	86.31 15187	90.69 15187	90.9 15187	90.15 15187	93.51 15187

Benchs	3BCID	MonoTest		MoHC	
		\neg OG	OG	\neg OG	OG
butcher*	351.1	354.28	347.37	235.1	10.27
	228887	228885	228823	112513	4871
fourbar*	257.71	133.37	22.95	96.6	18.87
	154171	97303	21105	55527	5053
geneig*	159.42	142.22	116	115.23	94.69
	45677	43283	36797	22627	12581
pramanik	95.7	59.58	35.9	51.28	23.18
	124661	98971	69259	59639	18487
trigo1-10	145.83	149.19	150.9	127.61	71.32
	2565	2565	2565	1831	673
virasoro*	22.1	22.5	20.1	23.84	16.4
	2789	2789	2641	2757	689
yam-8	11.62	11.82	11.66	11.51	2.35
	3017	3017	3017	2815	393

Conclusions

- We have proposed **Occurrence Grouping**, a new method to obtain sharper evaluations using monotonicity.
- OG adds monotonicities of non-monotonic variables
 $\Rightarrow [f^*]_M \subseteq [f]_M$.
- We solve an overestimation of an interesting problem:
minimization of the width of the evaluation by monotonicity of f^* .
- The results show that OG can produce considerable improvements.

Introduction of auxiliary variables into an interval function for improving its monotonicity-based evaluation

Ignacio Araya, Bertrand Neveu, Gilles Trombettoni

INRIA, Université de Nice-Sophia, CERTIS, France

SWIM, Juin 2009, Lausanne, Switzerland