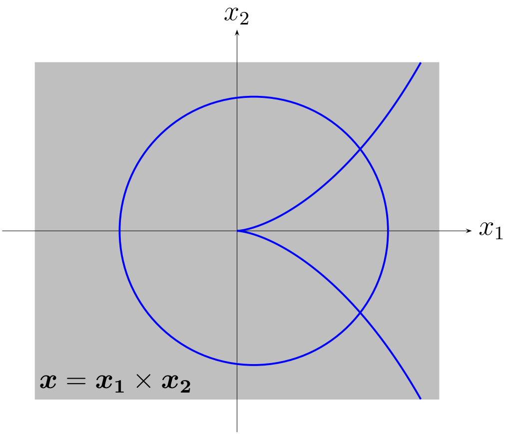
Adaptive Bisection of Numerical CSPs

Laurent Granvilliers

Univ. Nantes, Lab. Computer Science, France

Bisection Algorithm

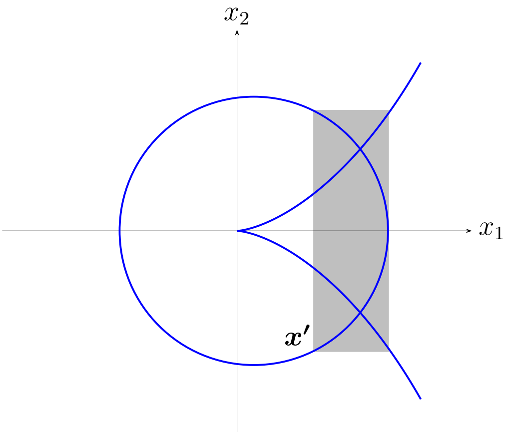
Goal : Solving numerical CSPs using interval computations.



$$\begin{cases} x_1(x_1^2 + x_2^2) = 6x_2^2 \\ (x_1 - 0.25)^2 + x_2^2 = 4 \\ (x_1, x_2) \in \boldsymbol{x_1} \times \boldsymbol{x_2} \end{cases}$$

Bisection Algorithm

Goal : Solving numerical CSPs using interval computations.

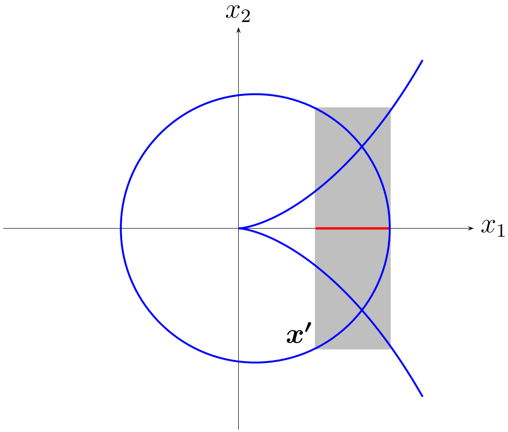


$$\begin{cases} x_1(x_1^2 + x_2^2) = 6x_2^2 \\ (x_1 - 0.25)^2 + x_2^2 = 4 \\ (x_1, x_2) \in \boldsymbol{x_1} \times \boldsymbol{x_2} \end{cases}$$

- Contract $\boldsymbol{x} \rightarrow \boldsymbol{x'}$

Bisection Algorithm

Goal : Solving numerical CSPs using interval computations.

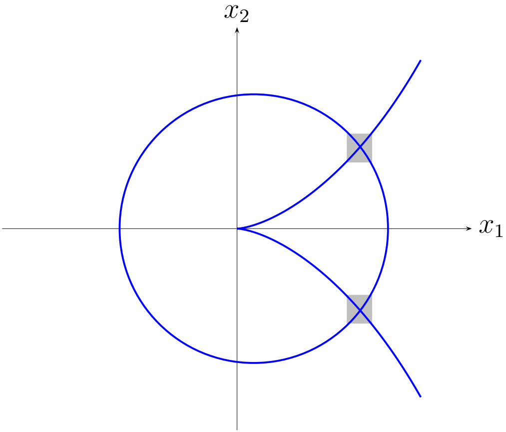


$$\begin{cases} x_1(x_1^2 + x_2^2) = 6x_2^2 \\ (x_1 - 0.25)^2 + x_2^2 = 4 \\ (x_1, x_2) \in \boldsymbol{x_1} \times \boldsymbol{x_2} \end{cases}$$

- Contract $\boldsymbol{x} \to \boldsymbol{x'}$
- Select x_2
- Bisect $\boldsymbol{x'}$ along x_2

Bisection Algorithm

Goal : Solving numerical CSPs using interval computations.



$$\begin{cases} x_1(x_1^2 + x_2^2) = 6x_2^2 \\ (x_1 - 0.25)^2 + x_2^2 = 4 \\ (x_1, x_2) \in \boldsymbol{x_1} \times \boldsymbol{x_2} \end{cases}$$

- Contract $\boldsymbol{x} \rightarrow \boldsymbol{x'}$
- Select x_2
- Bisect $\boldsymbol{x'}$ along x_2
- Iterate and stop on ϵ-boxes

MaxDom Strategy

Let $f(x_1, x_2) = x_1(x_1^2 + x_2^2) - 6x_2^2$ be the function defining the cissoid.
Let $\boldsymbol{x} = [2,4] \times [0,1]$ and evaluate the natural extension of f :

$$\boldsymbol{f}(\boldsymbol{x}) = [2,4]\left([2,4]^2 + [0,1]^2\right) - 6\,[0,1]^2 = [2,68]$$

MaxDom Strategy

Let $f(x_1, x_2) = x_1(x_1^2 + x_2^2) - 6x_2^2$ be the function defining the cissoid.
Let $\boldsymbol{x} = [2,4] \times [0,1]$ and evaluate the natural extension of f :

$$\boldsymbol{f}(\boldsymbol{x}) = [2,4]\left([2,4]^2 + [0,1]^2\right) - 6\,[0,1]^2 = [2,68]$$

Bisect $\boldsymbol{x_1}$

$$\boldsymbol{f}([2,3]\,, \boldsymbol{x_2}) = [2,30]$$
$$\boldsymbol{f}([3,4]\,, \boldsymbol{x_2}) = [21,68]$$

Bisect $\boldsymbol{x_2}$

$$\boldsymbol{f}(\boldsymbol{x_1}, [0,0.5]) = [6.5,65]$$
$$\boldsymbol{f}(\boldsymbol{x_1}, [0.5,1]) = [2.5,66.5]$$

MaxDom Strategy

Let $f(x_1, x_2) = x_1(x_1^2 + x_2^2) - 6x_2^2$ be the function defining the cissoid.
Let $\boldsymbol{x} = [2,4] \times [0,1]$ and evaluate the natural extension of f :

$$\boldsymbol{f}(\boldsymbol{x}) = [2,4]\left([2,4]^2 + [0,1]^2\right) - 6\,[0,1]^2 = [2,68]$$

Bisect $\boldsymbol{x_1}$

$$\boldsymbol{f}([2,3], \boldsymbol{x_2}) = [2,30]$$
$$\boldsymbol{f}([3,4], \boldsymbol{x_2}) = [21,68]$$

Bisect $\boldsymbol{x_2}$

$$\boldsymbol{f}(\boldsymbol{x_1}, [0,0.5]) = [6.5,65]$$
$$\boldsymbol{f}(\boldsymbol{x_1}, [0.5,1]) = [2.5,66.5]$$

MaxDom : bisecting the largest domain reduces overestimation in interval computations $\implies$ more contraction using local consistency

MaxSmear Strategy

Let $\boldsymbol{x} = [0.5, 1] \times [0.5, 1]$ and evaluate the derivatives of f :

$$\boldsymbol{\nabla f}(\boldsymbol{x}) = \left([1, 4]\,, [-11.5, -4]\right)$$

MaxSmear Strategy

Let $\boldsymbol{x} = [0.5, 1] \times [0.5, 1]$ and evaluate the derivatives of f :

$$\boldsymbol{\nabla f}(\boldsymbol{x}) = ([1, 4], [-11.5, -4])$$

Let c be the midpoint of $\boldsymbol{x}$ and evaluate the mean value extension of f :

$$\boldsymbol{f}(\boldsymbol{x}, c) = f(c) + \boldsymbol{\nabla f}(\boldsymbol{x})(\boldsymbol{x} - c) = [-6.5, 1.4]$$

MaxSmear Strategy

Let $\boldsymbol{x} = [0.5, 1] \times [0.5, 1]$ and evaluate the derivatives of f :

$$\boldsymbol{\nabla f}(\boldsymbol{x}) = ([1, 4], [-11.5, -4])$$

Let c be the midpoint of $\boldsymbol{x}$ and evaluate the mean value extension of f :

$$\boldsymbol{f}(\boldsymbol{x}, c) = f(c) + \boldsymbol{\nabla f}(\boldsymbol{x})(\boldsymbol{x} - c) = [-6.5, 1.4]$$

Bisect $\boldsymbol{x_1}$

$$\boldsymbol{f}([0.5, 0.75], \boldsymbol{x_2}) = [-6.0, 0.5]$$
$$\boldsymbol{f}([0.75, 1.], \boldsymbol{x_2}) = [-5.6, 1.1]$$

Bisect $\boldsymbol{x_2}$

$$\boldsymbol{f}(\boldsymbol{x_1}, [0.75, 1]) = [-6.0, -1.2]$$
$$\boldsymbol{f}(\boldsymbol{x_1}, [0.5, 0.75]) = [-3.6, 0.4]$$

MaxSmear Strategy

Let $\boldsymbol{x} = [0.5, 1] \times [0.5, 1]$ and evaluate the derivatives of f :

$$\boldsymbol{\nabla f}(\boldsymbol{x}) = ([1, 4], [-11.5, -4])$$

Let c be the midpoint of $\boldsymbol{x}$ and evaluate the mean value extension of f :

$$\boldsymbol{f}(\boldsymbol{x}, c) = f(c) + \boldsymbol{\nabla f}(\boldsymbol{x})(\boldsymbol{x} - c) = [-6.5, 1.4]$$

Bisect $\boldsymbol{x_1}$

$$\boldsymbol{f}([0.5, 0.75], \boldsymbol{x_2}) = [-6.0, 0.5]$$
$$\boldsymbol{f}([0.75, 1.], \boldsymbol{x_2}) = [-5.6, 1.1]$$

Bisect $\boldsymbol{x_2}$

$$\boldsymbol{f}(\boldsymbol{x_1}, [0.75, 1]) = [-6.0, -1.2]$$
$$\boldsymbol{f}(\boldsymbol{x_1}, [0.5, 0.75]) = [-3.6, 0.4]$$

MaxSmear : selecting the variable having the maximum smear value produces tighter linear relaxations $\Longrightarrow$ Newton operator stronger

RoundRobin Strategy

- Total order $x_{i_1} < x_{i_2} < \cdots < x_{i_n}$
  - $x_i < x_j$ if x_i occurs more than x_j
  - $x_i < x_j$ if x_i occurs in more constraints than x_j
  - ...

RoundRobin Strategy

- Total order $x_{i_1} < x_{i_2} < \cdots < x_{i_n}$
  - $x_i < x_j$ if x_i occurs more than x_j
  - $x_i < x_j$ if x_i occurs in more constraints than x_j
  - ...
- RoundRobin : select $x_{i_1}, x_{i_2}, \ldots, x_{i_n}, x_{i_1}, x_{i_2}, \ldots x_{i_n}, x_{i_1}, \ldots$
  - every domain is regularly bisected
  - fair strategy

New Adaptive Strategy

- Motivation
  - robust smear strategy
  - efficient fair strategy

New Adaptive Strategy

- Motivation
  - robust smear strategy
  - efficient fair strategy
- Adaptive bisection strategy
  - diversification ($\approx$ RoundRobin) in the early steps of the algorithm
  - intensification ($\approx$ MaxSmear) in the vicinity of solutions

New Adaptive Strategy

- Motivation
  - robust smear strategy
  - efficient fair strategy
- Adaptive bisection strategy
  - diversification ($\approx$ RoundRobin) in the early steps of the algorithm
  - intensification ($\approx$ MaxSmear) in the vicinity of solutions
- GRASP (Feo & Resende 1995)
  - adaptive search procedure for derivative-free optimization
  - greedy + randomized

New Algorithm

- Initialization : for every variable $(i = 1, \ldots, n)$
  - $s_i \geq 0$: smear value of x_i in $\boldsymbol{x}$
  - $n_i \geq 0$: number of times x_i has been selected
  - $[s_{\min}, s_{\max}]$: range of smear values

New Algorithm

- Initialization : for every variable $(i = 1, \ldots, n)$
  - $s_i \geq 0$: smear value of x_i in $\boldsymbol{x}$
  - $n_i \geq 0$: number of times x_i has been selected
  - $[s_{\min}, s_{\max}]$: range of smear values

- Subset of the best variables : let $S \in [s_{\min}, s_{\max}]$ and mark every variable x_j s.t.

$$s_j \geq S \wedge \text{width}(\boldsymbol{x_j}) > \epsilon$$

New Algorithm

- Initialization : for every variable $(i = 1, \dots, n)$
  - $s_i \geq 0$: smear value of x_i in $\boldsymbol{x}$
  - $n_i \geq 0$: number of times x_i has been selected
  - $[s_{\min}, s_{\max}]$: range of smear values

- Subset of the best variables : let $S \in [s_{\min}, s_{\max}]$ and mark every variable x_j s.t.

$$s_j \geq S \wedge \mathrm{width}(\boldsymbol{x_j}) > \epsilon$$

- Fair choice : select a marked variable x_k s.t.

$$n_k = \min\{n_i : 1 \leq i \leq n \wedge x_i \text{ marked}\}$$

Adaptive Behaviour

- Threshold on smear values

$$S = s_{\min} + \alpha(s_{\max} - s_{\min}),\ 0 \leq \alpha \leq 1$$

  - $\alpha = 1 \implies$ greedy
  - $\alpha = 0 \implies$ fair

Adaptive Behaviour

- Threshold on smear values

$$S = s_{\min} + \alpha(s_{\max} - s_{\min}),\ 0 \le \alpha \le 1$$

  - $\alpha = 1 \implies$ greedy
  - $\alpha = 0 \implies$ fair

- Adaptive behaviour

$$\alpha = \frac{1}{1 + \beta\sigma},\ \beta > 0$$

  - σ : standard deviation of the smear values
  - $\sigma \to 0 \implies \alpha \to 1$
  - $\sigma \to \infty \implies \alpha \to 0$

Experimental Results

Problem	n	MaxDom	RoundRobin	MaxSmear	Adaptive
Celestial	3	3 361	**2 599**	4 258	3 074
Combu.	10	559	**463**	1 303	509
Neuro	6	1 288 779	**6 367**	7 223	7 215
Trigo1	10	1 258	**1 244**	2 086	2 071
Broyden3	20	24	723	**23**	23
Geineg	6	8 546	8 116	**1 707**	2 400
Kapur	5	2 791	1 651	**231**	323
Nbody	8	1 976	2 049	**1 532**	1 765
Brown	10	12 443	5 850	6 827	**4 996**
Eco	6	1 108	1 530	1 063	**1 023**
Nauheim	8	1 204	722	816	**708**
Transistor	9	112 795	121 765	83 711	**42 085**

Experimental Results

Problem	n	MaxDom	RoundRobin	MaxSmear	Adaptive
Celestial	3	3 361	**2 599**	4 258	3 074
Combu.	10	559	**463**	1 303	509
Neuro	6	1 288 779	**6 367**	7 223	7 215
Trigo1	10	1 258	**1 244**	2 086	2 071
Broyden3	20	24	723	**23**	23
Geineg	6	8 546	8 116	**1 707**	2 400
Kapur	5	2 791	1 651	**231**	323
Nbody	8	1 976	2 049	**1 532**	1 765
Brown	10	12 443	5 850	6 827	**4 996**
Eco	6	1 108	1 530	1 063	**1 023**
Nauheim	8	1 204	722	816	**708**
Transistor	9	112 795	121 765	83 711	**42 085**

Experimental Results

Problem	n	MaxDom	RoundRobin	MaxSmear	Adaptive
Celestial	3	3 361	**2 599**	4 258	3 074
Combu.	10	559	**463**	1 303	509
Neuro	6	1 288 779	**6 367**	7 223	7 215
Trigo1	10	1 258	**1 244**	2 086	2 071
Broyden3	20	24	723	**23**	23
Geineg	6	8 546	8 116	**1 707**	2 400
Kapur	5	2 791	1 651	**231**	323
Nbody	8	1 976	2 049	**1 532**	1 765
Brown	10	12 443	5 850	6 827	**4 996**
Eco	6	1 108	1 530	1 063	**1 023**
Nauheim	8	1 204	722	816	**708**
Transistor	9	112 795	121 765	83 711	**42 085**

Conclusion

- The adaptive strategy is more robust.

Conclusion

- The adaptive strategy is more robust.

- Other adaptive schemes could be designed.
  - aggregation of s_i and n_i : $(\max_j n_j - n_i + 1) \times s_i$
  - learning techniques