Search Strategies for an Anytime Usage of the Branch and Prune Algorithm

R. Chenouard[1], A. Goldsztejn[2] and C. Jermann[1]

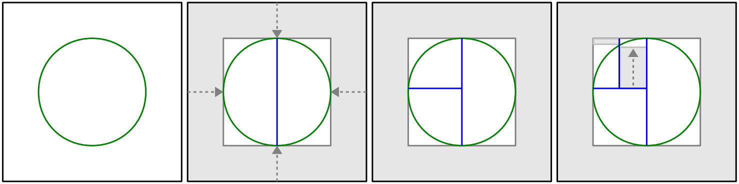
SWIM 2009

[1] Lina UMR CNRS 6241, University of Nantes, France
[2] CNRS, Lina UMR CNRS 6241, France

The Branch and Prune Algorithm

Alternates search (branch) and filtering (prune).

$\Rightarrow$ set of ε-boxes covering the solution set.



- Boxes to be processed : $\mathcal{L}$
- Computed ε-boxes : $\mathcal{S}$ (can be post-processed)

Search Issue

How to choose the next box to process in $\mathcal{L}$?

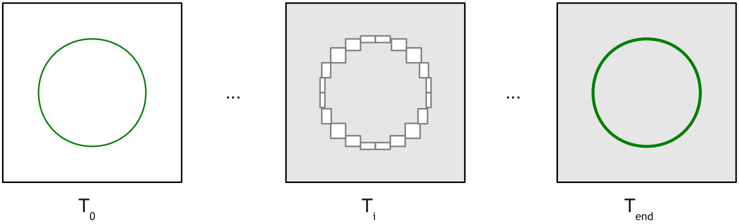
Breadth-First Search

Principle of BFS

Selects boxes in $\mathcal{L}$ following a FIFO strategy.

Main behavior

Explores uniformely the whole search space.



$\Rightarrow$ At time T_i no ε-boxes are computed.

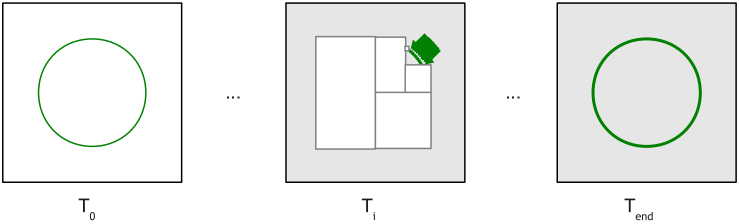
Depth-First Search

Principle of DFS

Selects boxes in $\mathcal{L}$ following a LIFO strategy.

Main behavior

Descends rapidly to ε-boxes then finds them by neighborhood.



$\Rightarrow$ At time T_i many similiar ε-boxes are computed.

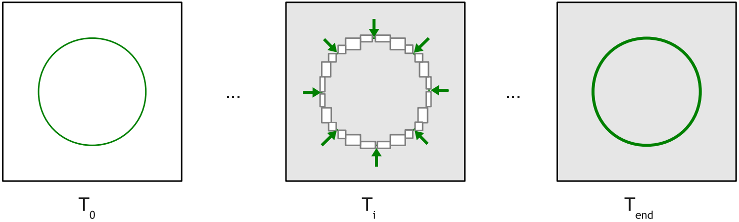
Ideal Search Strategy for an Anytime Usage

Principle

Selects boxes in $\mathcal{L}$ mixing BFS and DFS principles.

Main behavior

Explores uniformely the whole search space and find well-ditributed ε-boxes at early stage of the search.



Most Distant First Strategy

Principle of MDFS

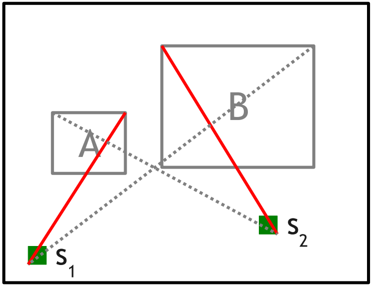
Select in $\mathcal{L}$ the most distant box to ε-boxes in $\mathcal{S}$.

Distance evaluation

$$\underset{[\mathbf{x}_1],\ldots,[\mathbf{x}_m]}{\mathrm{MinDist}}(\mathbf{x}) = \min\{d(\mathbf{x},[\mathbf{x}_1]),\ldots,d(\mathbf{x},[\mathbf{x}_m])\}, \qquad (1)$$

where d is the maximal distance between points of two boxes.

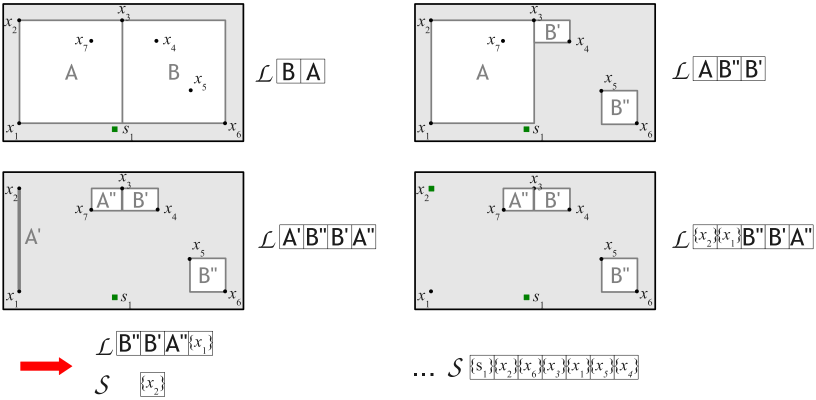
MinDist Evaluation Example



$$\underset{[s_1],[s_2]}{\mathrm{MinDist}}(A) = \min\{d(A,[s_1]), d(A,[s_2])\} = d(A,[s_1]),$$

$$\underset{[s_1],[s_2]}{\mathrm{MinDist}}(B) = \min\{d(B,[s_1]), d(B,[s_2])\} = d(B,[s_2]).$$

MDFS : a simple example



MDFS Theoretical Property

The next computed ε-box $\mathbf{x}_{m+1}$ satisfies :

$$\left| d^* - \underset{[\mathbf{x}_1],\ldots,[\mathbf{x}_m]}{\mathrm{MinDist}}(\mathbf{x}_{m+1}) \right| \leq \varepsilon,$$

where d^* is the global maximum of MinDist.

Therefore :

$$\lim_{\varepsilon \to 0} \underset{[\mathbf{x}_1],\ldots,[\mathbf{x}_m]}{\mathrm{MinDist}}(\mathbf{x}_{m+1}) = d^*$$

$\Rightarrow$ Slow discovery of ε-boxes due to breadth exploration.

Depth and Most Distant First Strategy

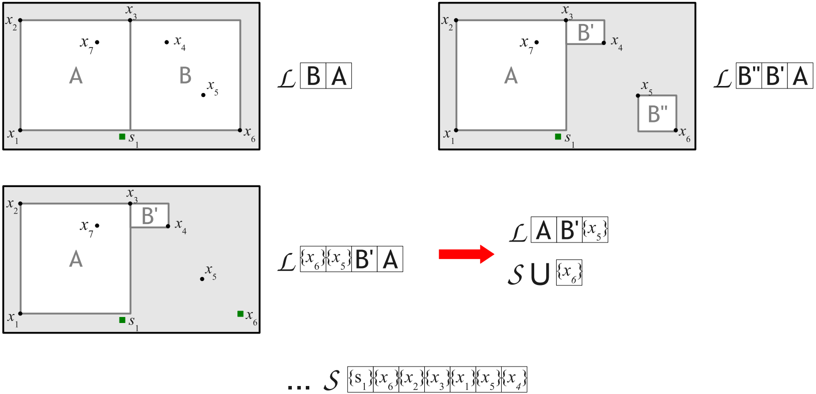
Principle of DMDFS

Mixing MDFS and DFS to quicken the discovery of ε-boxes :

- Select boxes in $\mathcal{L}$ following a LIFO* strategy.
- Sort $\mathcal{L}$ wrt. (1) after each insertion in $\mathcal{S}$.

*Heuristic based on MinDist for depth search.

DMDFS : a simple example



DMDFS and MDFS Theoretical comparison

	Optimum convergence	Quick discovery
MDFS	+	-
DMDFS	?	+

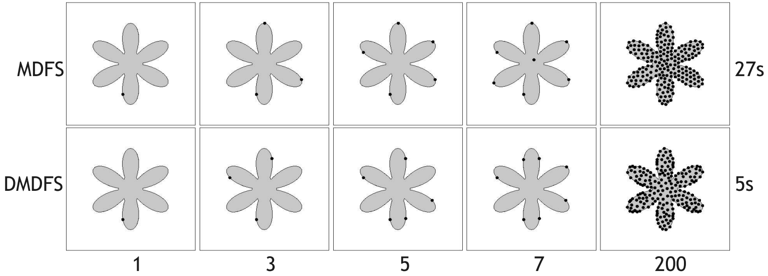
Evaluation of the spreading of ε-boxes using DMDFS :

- Homogeneous coverage of solution space.
- Reach every solution spaces.

Homogeneous coverage of DMDFS

Consider $(x,y) \in [-2,2]^2$, s.t. :

$$11x^8 - x^6 - 183x^4y^2 + 44x^6y^2 + 117x^2y^4 + 66x^4y^4 - 21y^6 + 44x^2y^6 + 11y^8 \leq 0. \quad (2)$$

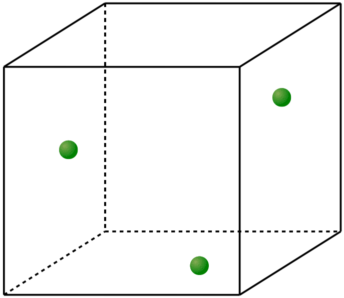


A scalable problem

n-dimensional balls randomly placed in $[-100, 100]^n$:

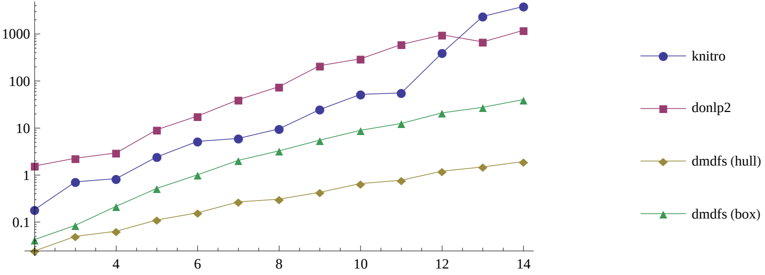
$$\begin{aligned} \sum_{i=1}^{n} (x_i - c_{1i})^2 - 1 &= y_1 \\ &\vdots \\ \sum_{i=1}^{n} (x_i - c_{ni})^2 - 1 &= y_n \\ \prod_{i=1}^{n} y_i &\leq 0. \end{aligned}$$

Example with n=3 :



DMDFS vs other approaches

Time to find 1 solution in each *n*-dimensional ball :



Conclusion and Future Work

New search strategy to use efficiently the BPA as an anytime algorithm :

- Good theoretical optimal property of MDFS.
- Good practical performances of DMDFS.

Future work

- Extension to discrete or mixed problems.
- Applying DMDFS to real problems in mechanical design.
- Comparison to other approaches ?