

Sondes locales intensives lors de l'exécution de MADS dans un environnement parallèle

Guillaume LAMEYNARDIE

Directeur : Charles AUDET, Co-directeur : Sébastien LE DIGABEL

Supervisé à l'IREQ par : Stéphane ALARIE



Plan

Introduction

- L'optimisation de boîte noire

- L'algorithme MADS

- Exploitation de ressources parallèles

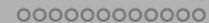
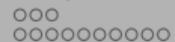
Motivations

Occupation des ressources parallèles

- A l'étape de recherche globale

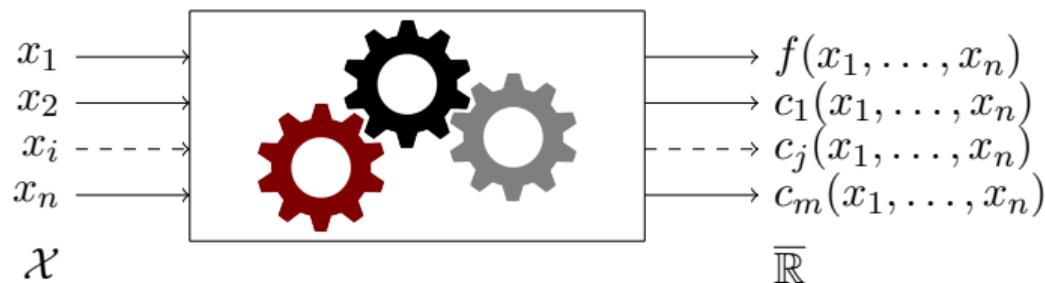
- A l'étape de sonde locale

Résultats numériques

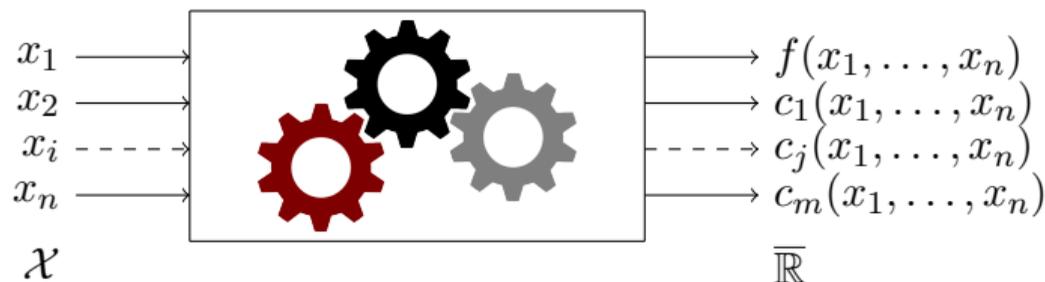


Introduction

L'optimisation de boîte noire

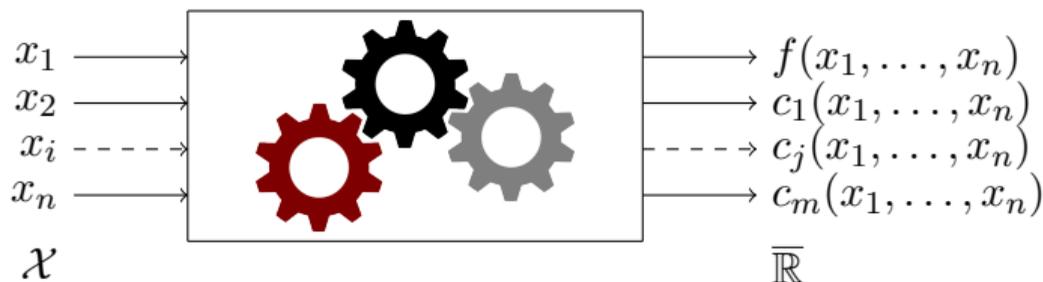


L'optimisation de boîte noire



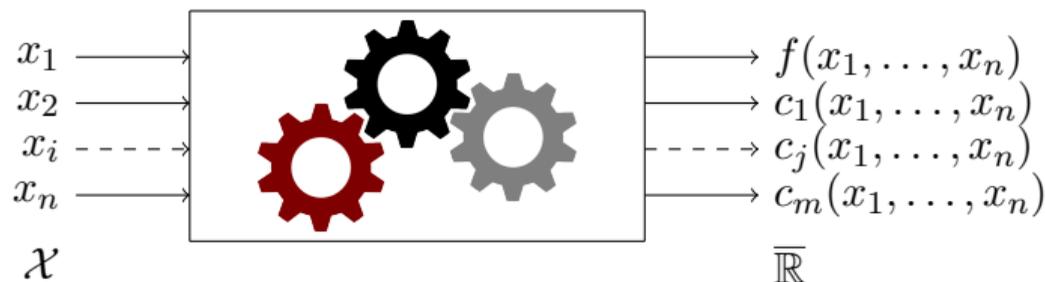
- ▶ $x \in \mathcal{X} \subset \mathbb{R}^n$.
- ▶ f et $c_j : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\} = \bar{\mathbb{R}}$.
- ▶ On cherche $x^* \in \mathcal{X}$ qui retourne la plus petite valeur de f tel que $c_j(x^*) \leq 0$ pour tout j .

L'optimisation de boîte noire



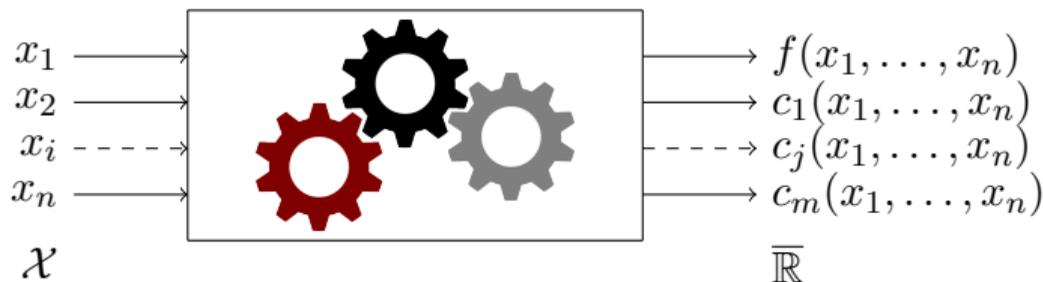
- ▶ $x \in \mathcal{X} \subset \mathbb{R}^n$.
- ▶ f et $c_j : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\} = \bar{\mathbb{R}}$.
- ▶ On cherche $x^* \in \mathcal{X}$ qui retourne la plus petite valeur de f tel que $c_j(x^*) \leq 0$ pour tout j .

L'optimisation de boîte noire



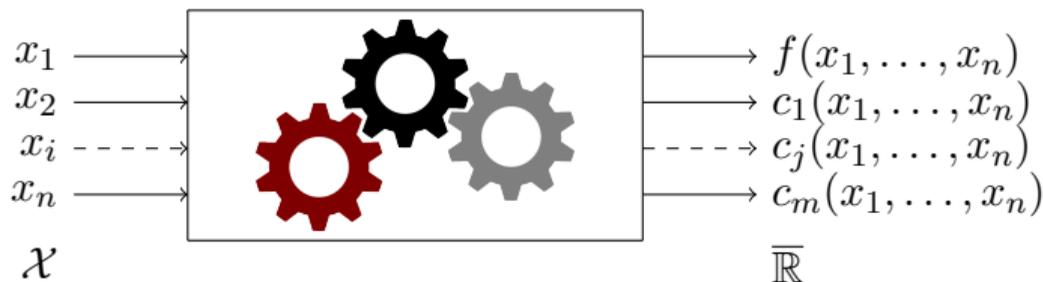
- ▶ $x \in \mathcal{X} \subset \mathbb{R}^n$.
- ▶ f et $c_j : \mathcal{X} \rightarrow \mathbb{R} \cup \{+\infty\} = \bar{\mathbb{R}}$.
- ▶ On cherche $x^* \in \mathcal{X}$ qui retourne la plus petite valeur de f tel que $c_j(x^*) \leq 0$ pour tout j .

L'optimisation de boîte noire



- ▶ Calculer une fois f et c_j est coûteux en temps et en ressources.
- ▶ Les dérivées ne sont pas accessibles, et même si elles existent (ce qui n'est pas garanti) elles ne seraient d'aucune utilité.
- ▶ f et c_j sont déterministes.

L'optimisation de boîte noire

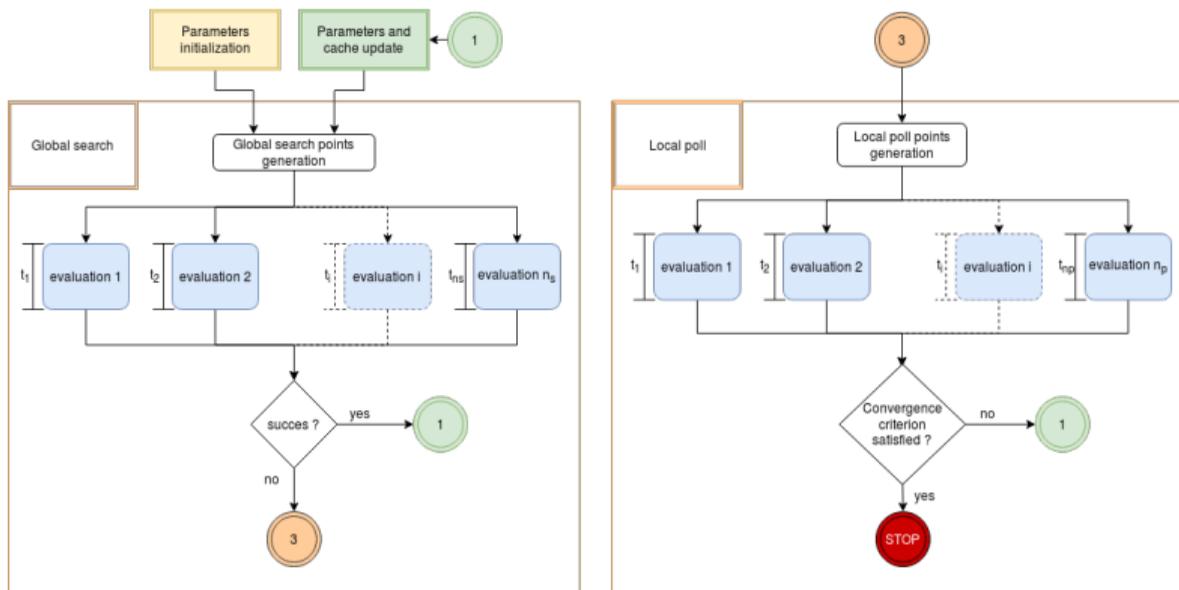


- ▶ Calculer une fois f et c_j est coûteux en temps et en ressources.
- ▶ Les dérivées ne sont pas accessibles, et même si elles existent (ce qui n'est pas garanti) elles ne seraient d'aucune utilité.
- ▶ f et c_j sont déterministes.

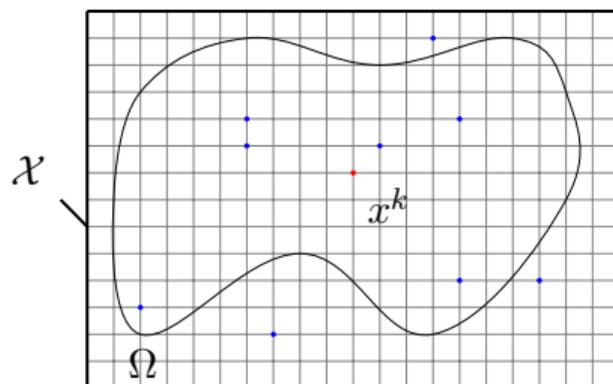
L'optimisation de boîte noire

$$\mathcal{P} : \begin{cases} \min_{x \in \mathcal{X}} & f(x) \\ \text{s.c} & c_j(x) \leq 0, \quad j \in \llbracket 1, m \rrbracket. \end{cases}$$

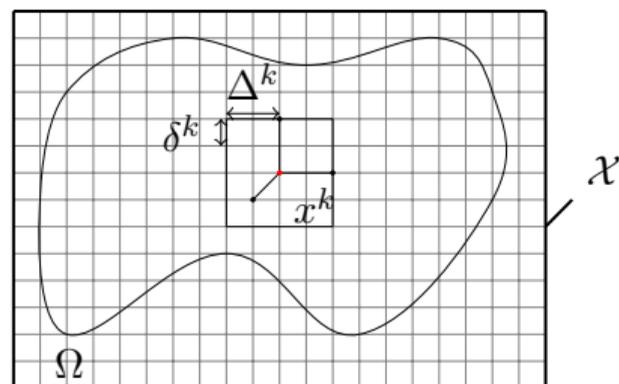
L'algorithme MADS [Audet and Dennis, Jr. (2006)]



L'algorithme MADS [Audet and Dennis, Jr. (2006)]



(a)



(b)

FIGURE – Si l'étape de recherche globale (a) ne parvient pas à améliorer f , on effectue l'étape de sonde locale (b).

L'algorithme MADS [Audet and Dennis, Jr. (2006)]

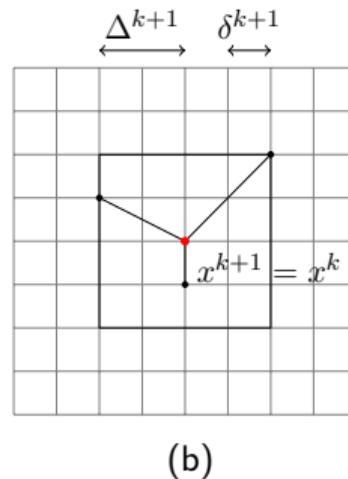
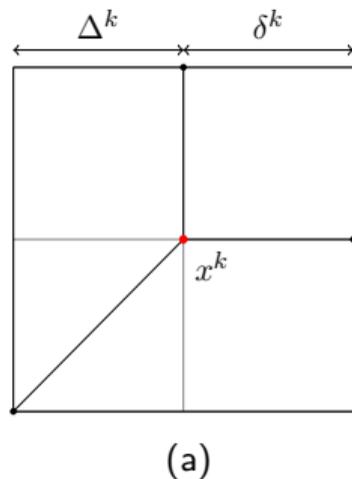


FIGURE – Si l'étape de sonde locale ne parvient pas à améliorer la valeur de f (a) alors δ^k et Δ^k sont diminués (b).

Les règles de génération de points dans MADS

Dans la recherche globale :

- ▶ N'importe où sur le maillage.
- ▶ En un nombre fini.

Les règles de génération de points dans MADS

Dans la sonde locale :

- ▶ Sur le maillage, à une distance maximale de Δ^k de la meilleure solution connue x^k .
- ▶ les directions entre les points de sonde et x^k doivent former un ensemble générateur positif.

Exploitation de ressources parallèles

Definition (Processeur)

Pour une boîte noire et un utilisateur donné, un *processeur* est un ensemble de ressources que l'utilisateur juge à priori capable de mener à terme n'importe laquelle des évaluations de cette boîte noire en une durée inférieure à une borne fixée par ce même utilisateur.

Exploitation de ressources parallèles

Definition (Processeur)

Pour une boîte noire et un utilisateur donné, un *processeur* est un ensemble de ressources que l'utilisateur juge à priori capable de mener à terme n'importe laquelle des évaluations de cette boîte noire en une durée inférieure à une borne fixée par ce même utilisateur.

- ▶ processeur \neq CPU : puce électronique dans un ordinateur.

Exploitation de ressources parallèles

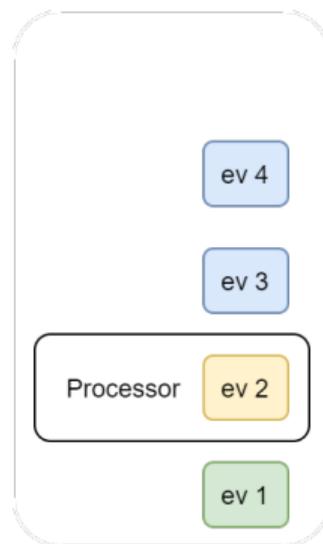


FIGURE – Pile d'évaluation séquentielle

Exploitation de ressources parallèles

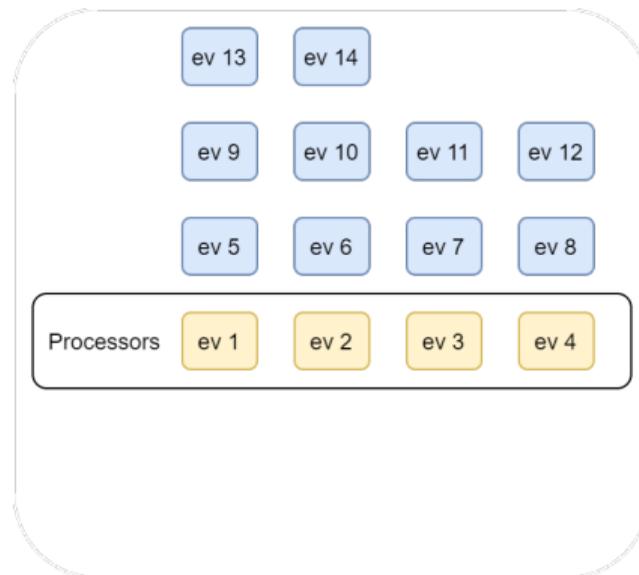


FIGURE – Pile d'évaluation parallèle avec plus d'évaluations que de processeurs

Exploitation de ressources parallèles

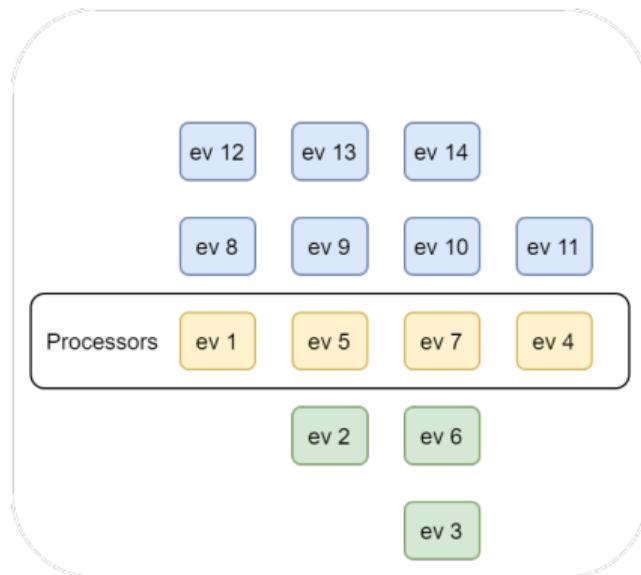


FIGURE – Pile d'évaluation parallèle avec plus d'évaluations que de processeurs

Exploitation de ressources parallèles

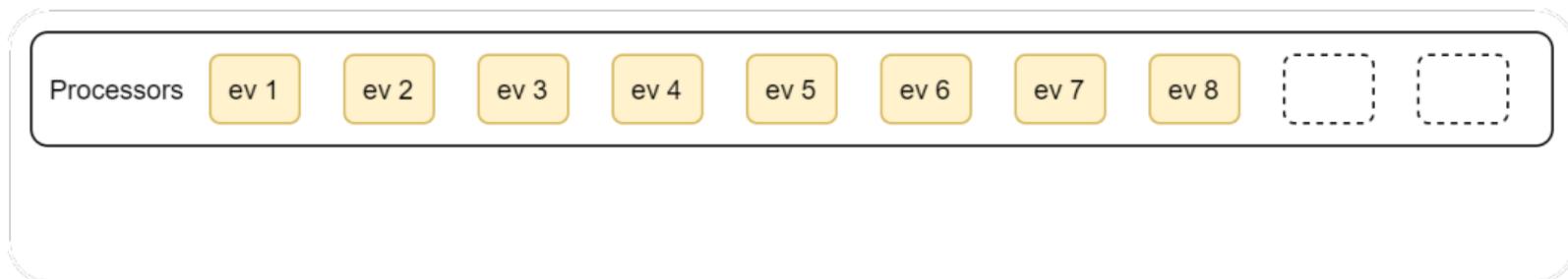


FIGURE – Pile d'évaluation parallèle avec plus de processeurs que d'évaluations

Exploitation de ressources parallèles

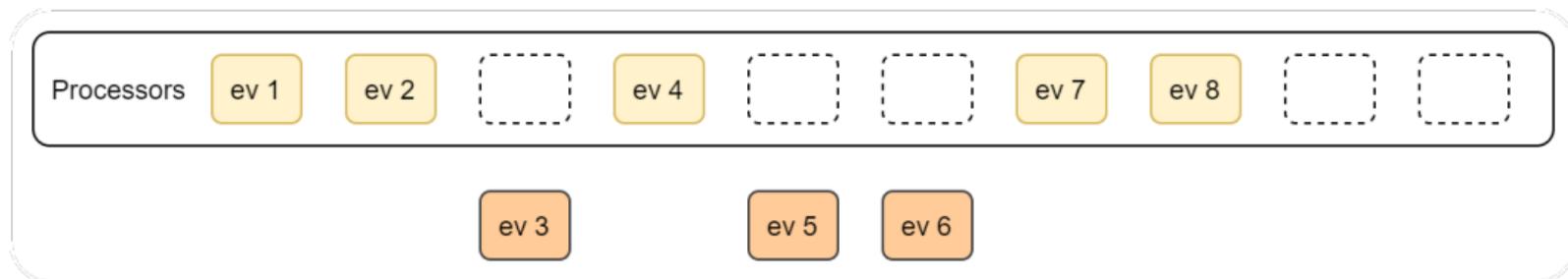
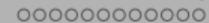


FIGURE – Pile d'évaluation parallèle avec plus de processeurs que d'évaluations



Motivations

Motivations

- ▶ Les environnements à fort degré de parallélisme (clusters, cloud computing) sont de plus en plus accessibles.
- ▶ Des problèmes avec un faible nombre de variables existent toujours.
- ▶ On espère qu'avec plus de ressources on puisse résoudre des problèmes plus efficacement.

Occupation des ressources parallèles

A l'étape de recherche globale

Dans la recherche globale

- ▶ Développées par Bastien Talgorn, Stéphane Alarie et Michael Kokkolaras Alarie (2019); Kokkolaras and Talgorn (2017).
- ▶ Effectue une optimisation sur un modèle dynamique du problème.
- ▶ Filtre les points générés à l'aide de six métriques différentes.
- ▶ Évalue les points filtrés sur la boîte noire.

Dans la recherche globale

$$\mathcal{S} : \begin{cases} \min_{x \in \mathcal{X}} & \tilde{f}(x) \\ \text{s.c} & \tilde{c}_j(x) \leq 0, \quad j \in \llbracket 1, m \rrbracket. \end{cases}$$

est un modèle de \mathcal{P} si il partage certaines similarités avec \mathcal{P} .

- ▶ Plus facile/rapide à résoudre que \mathcal{P}

Dans la recherche globale

Algorithme 1 Génération de points pour la recherche globale

[0]. Initialisation :

M^k : Maillage à l'itération k de MADS.
 S^k : Ensemble de points de recherche globale.
 $T \leftarrow \emptyset$.
 $\tilde{V} \leftarrow \emptyset$.
 \mathcal{F}_i : $i \in [1, 6]$: stratégies de filtrage.
 $i \leftarrow 1$.
 Choisir $\ell_{max} \in \mathbb{N}^*$: nb. max. de points à générer .
 Choisir V un ensemble de points déjà évalués sur \mathcal{P} .

[1]. Génération de \tilde{V} :

Construire S avec l'information donnée par V .
 Résoudre S et stocker les points évalués dans \tilde{V} .
 Aller à [2].

[2]. Filtrage des points de \tilde{V}

Tant que $\mathcal{F}_i(\tilde{V})$ retourne un point \bar{s} **faire** :

Si $|T| < \ell_{max}$ **faire** :

$T \leftarrow T \cup \{\bar{s}\}$.

Sinon :

 Go to [3].

Sinon :

$i \leftarrow i + 1$.

 Aller à [2].

[3]. Projection sur le maillage

Pour $\bar{s} \in T$ **faire** :

 Projeter \bar{s} sur M^k pour obtenir s .

$S^k \leftarrow S^k \cup \{s\}$

On utilise V pour construire

$$S : \begin{cases} \min_{x \in \mathcal{X}} & \tilde{f}(x), \\ \text{s.c} & \tilde{c}_j(x) \leq 0, \quad j \in \llbracket 1, m \rrbracket. \end{cases}$$

Dans la recherche globale

Algorithme 2 Génération de points pour la recherche globale

[0]. Initialisation :

M^k : Maillage à l'itération k de MADS.
 S^k : Ensemble de points de recherche globale.
 $T \leftarrow \emptyset$.
 $\tilde{V} \leftarrow \emptyset$.
 $\mathcal{F}_i : i \in [1, 6]$: stratégies de filtrage.
 $i \leftarrow 1$.
 Choisir $\ell_{max} \in \mathbb{N}^*$: nb. max. de points à générer .
 Choisir V un ensemble de points déjà évalués sur \mathcal{P} .

[1]. Génération de \tilde{V} :

Construire S avec l'information donnée par V .
 Résoudre S et stocker les points évalués dans \tilde{V} .
 Aller à [2].

[2]. Filtrage des points de \tilde{V}

Tant que $\mathcal{F}_i(\tilde{V})$ retourne un point \bar{s} **faire** :

Si $|T| < \ell_{max}$ **faire** :

$T \leftarrow T \cup \{\bar{s}\}$.

Sinon :

 Go to [3].

Sinon :

$i \leftarrow i + 1$.

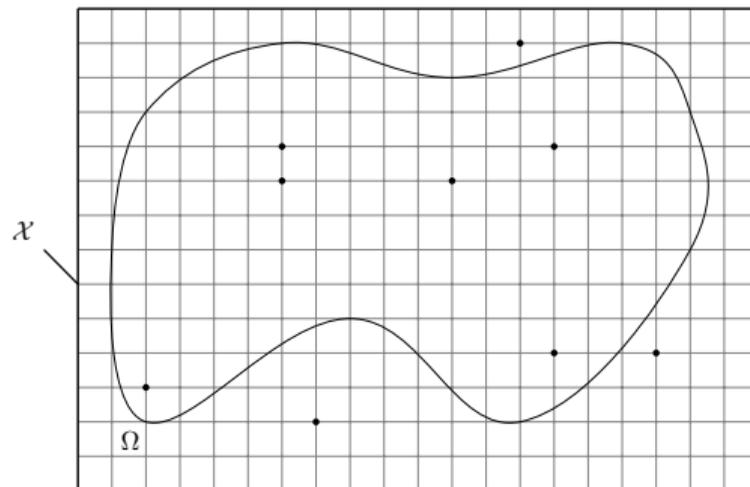
 Aller à [2].

[3]. Projection sur le maillage

Pour $\bar{s} \in T$ **faire** :

 Projeter \bar{s} sur M^k pour obtenir s .

$S^k \leftarrow S^k \cup \{s\}$



Dans la recherche globale

Algorithme 3 Génération de points pour la recherche globale

[0]. Initialisation :

M^k : Maillage à l'itération k de MADS.
 S^k : Ensemble de points de recherche globale.
 $T \leftarrow \emptyset$.
 $\tilde{V} \leftarrow \emptyset$.
 $\mathcal{F}_i : i \in [1, 6]$: stratégies de filtrage.
 $i \leftarrow 1$.
 Choisir $\ell_{max} \in \mathbb{N}^*$: nb. max. de points à générer .
 Choisir V un ensemble de points déjà évalués sur \mathcal{P} .

[1]. Génération de \tilde{V} :

Construire S avec l'information donnée par V .
 Résoudre S et stocker les points évalués dans \tilde{V} .
 Aller à [2].

[2]. Filtrage des points de \tilde{V}

Tant que $\mathcal{F}_i(\tilde{V})$ retourne un point \bar{s} **faire** :

Si $|T| < \ell_{max}$ **faire** :

$T \leftarrow T \cup \{\bar{s}\}$.

Sinon :

 Go to [3].

Sinon :

$i \leftarrow i + 1$.

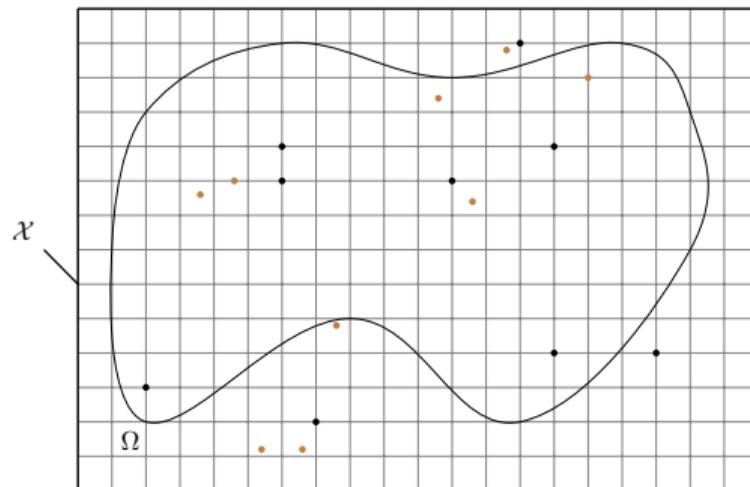
 Aller à [2].

[3]. Projection sur le maillage

Pour $\bar{s} \in T$ **faire** :

 Projeter \bar{s} sur M^k pour obtenir s .

$S^k \leftarrow S^k \cup \{s\}$



Dans la recherche globale

Algorithme 4 Génération de points pour la recherche globale

[0]. Initialisation :

M^k : Maillage à l'itération k de MADS.
 S^k : Ensemble de points de recherche globale.
 $T \leftarrow \emptyset$.
 $\tilde{V} \leftarrow \emptyset$.
 $\mathcal{F}_i : i \in [1, 6]$: stratégies de filtrage.
 $i \leftarrow 1$.
 Choisir $t_{max} \in \mathbb{N}^*$: nb. max. de points à générer .
 Choisir V un ensemble de points déjà évalués sur \mathcal{P} .

[1]. Génération de \tilde{V} :

Construire S avec l'information donnée par V .
 Résoudre S et stocker les points évalués dans \tilde{V} .
 Aller à [2].

[2]. Filtrage des points de \tilde{V}

Tant que $\mathcal{F}_i(\tilde{V})$ retourne un point \bar{s} **faire** :

Si $|T| < t_{max}$ **faire** :

$T \leftarrow T \cup \{\bar{s}\}$.

Sinon :

 Go to [3].

Sinon :

$i \leftarrow i + 1$.

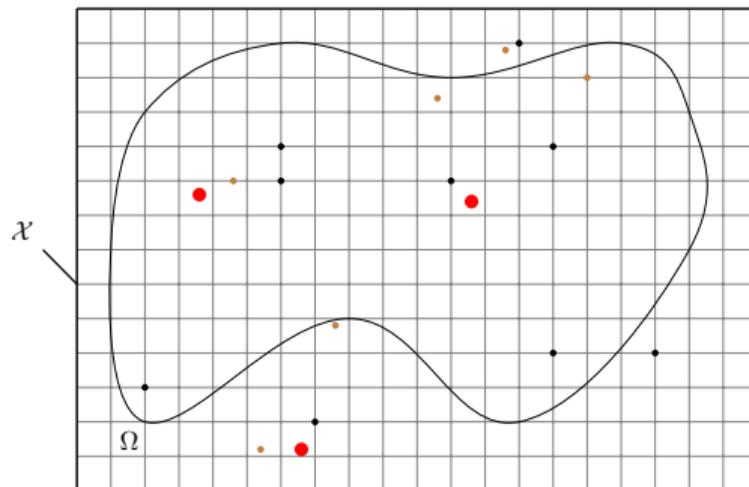
 Aller à [2].

[3]. Projection sur le maillage

Pour $\bar{s} \in T$ **faire** :

 Projeter \bar{s} sur M^k pour obtenir s .

$S^k \leftarrow S^k \cup \{s\}$



Occupation des ressources parallèles

A l'étape de sonde locale

Dans la sonde locale

- ▶ Trois stratégies
- ▶ Permet le dimensionnement en générant plus de $2n$ points à chaque étape de sonde locale
- ▶ Diffèrent dans la manière dont sont générés les points

sonde locale classique

Algorithme 6 sonde locale classique - Génération de points

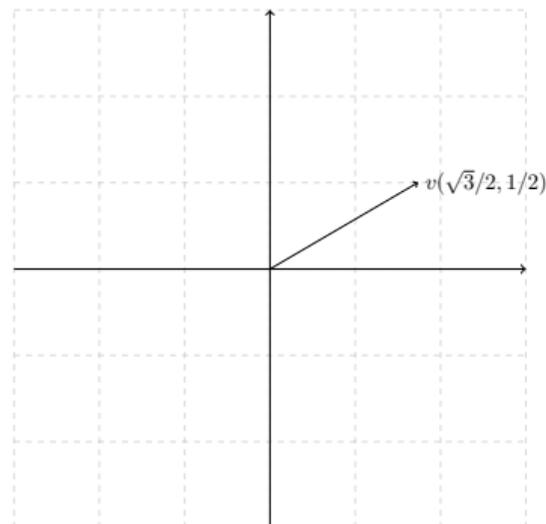
[0]. Initialisation :

 Δ^k : Pas de cadre. δ^k : Pas de maillage. x^k : Point donnant lieu au succès courant. $\mathbb{D} \leftarrow \emptyset$: Ensemble de direction de sonde locale. $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde locale.

[1]. Génération de directions de sonde :

Sélectionner $u \in \mathbb{R}^n$ non nulle.Poser $v = \frac{u}{\|u\|}$.Poser $H = I - 2vv^\top$. $\mathbb{D} \leftarrow \left\{ \pm \delta^k \text{round} \left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty} \right) : j \in \llbracket 1, n \rrbracket \right\} \subset \delta^k \mathbb{Z}^n$.

[2]. Génération de points de sonde :

 $\mathbb{P} \leftarrow \{x^k + d : d \in \mathbb{D}\}$.**RETOURNER** \mathbb{P} .

sonde locale classique

Algorithme 7 sonde locale classique - Génération de points

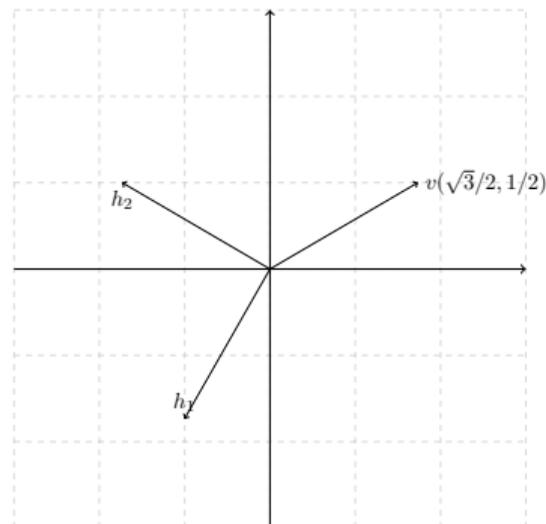
[0]. Initialisation :

 Δ^k : Pas de cadre. δ^k : Pas de maillage. x^k : Point donnant lieu au succès courant. $\mathbb{D} \leftarrow \emptyset$: Ensemble de direction de sonde locale. $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde locale.

[1]. Génération de directions de sonde :

Sélectionner $u \in \mathbb{R}^n$ non nulle.Poser $v = \frac{u}{\|u\|}$.Poser $H = I - 2vv^\top$. $\mathbb{D} \leftarrow \left\{ \pm \delta^k \text{round} \left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty} \right) : j \in \llbracket 1, n \rrbracket \right\} \subset \delta^k \mathbb{Z}^n$.

[2]. Génération de points de sonde :

 $\mathbb{P} \leftarrow \{x^k + d : d \in \mathbb{D}\}$.**RETOURNER** \mathbb{P} .

sonde locale classique

Algorithme 8 sonde locale classique - Génération de points

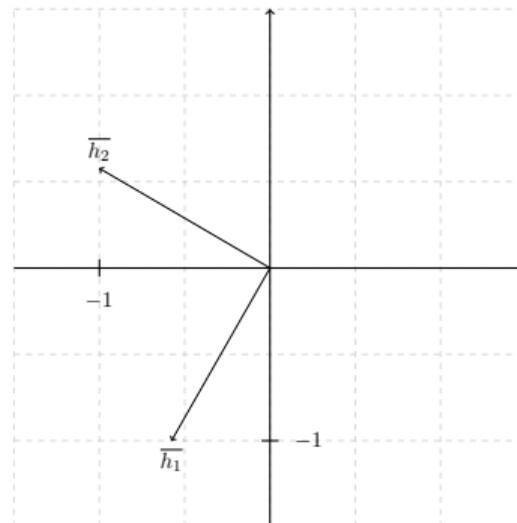
[0]. Initialisation :

 Δ^k : Pas de cadre. δ^k : Pas de maillage. x^k : Point donnant lieu au succès courant. $\mathbb{D} \leftarrow \emptyset$: Ensemble de direction de sonde locale. $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde locale.

[1]. Génération de directions de sonde :

Sélectionner $u \in \mathbb{R}^n$ non nulle.Poser $v = \frac{u}{\|u\|}$.Poser $H = I - 2vv^\top$. $\mathbb{D} \leftarrow \left\{ \pm \delta^k \text{round} \left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty} \right) : j \in \llbracket 1, n \rrbracket \right\} \subset \delta^k \mathbb{Z}^n$.

[2]. Génération de points de sonde :

 $\mathbb{P} \leftarrow \{x^k + d : d \in \mathbb{D}\}$.**RETOURNER** \mathbb{P} .

sonde locale classique

Algorithme 9 sonde locale classique - Génération de points

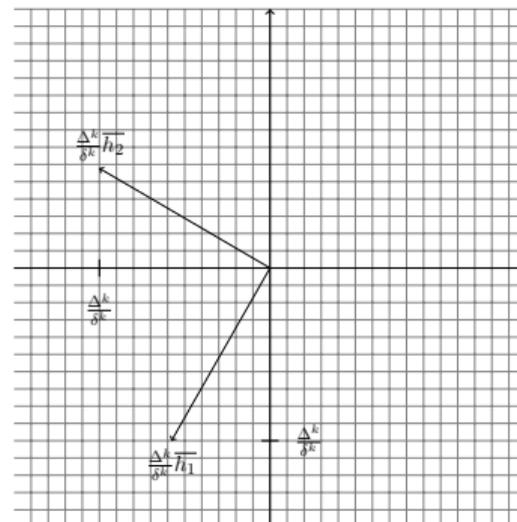
[0]. Initialisation :

 Δ^k : Pas de cadre. δ^k : Pas de maillage. x^k : Point donnant lieu au succès courant. $\mathbb{D} \leftarrow \emptyset$: Ensemble de direction de sonde locale. $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde locale.

[1]. Génération de directions de sonde :

Sélectionner $u \in \mathbb{R}^n$ non nulle.Poser $v = \frac{u}{\|u\|}$.Poser $H = I - 2vv^\top$. $\mathbb{D} \leftarrow \left\{ \pm \delta^k \text{round} \left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty} \right) : j \in \llbracket 1, n \rrbracket \right\} \subset \delta^k \mathbb{Z}^n$.

[2]. Génération de points de sonde :

 $\mathbb{P} \leftarrow \{x^k + d : d \in \mathbb{D}\}$.**RETOURNER** \mathbb{P} .

sonde locale classique

Algorithme 10 sonde locale classique - Génération de points

[0]. Initialisation :

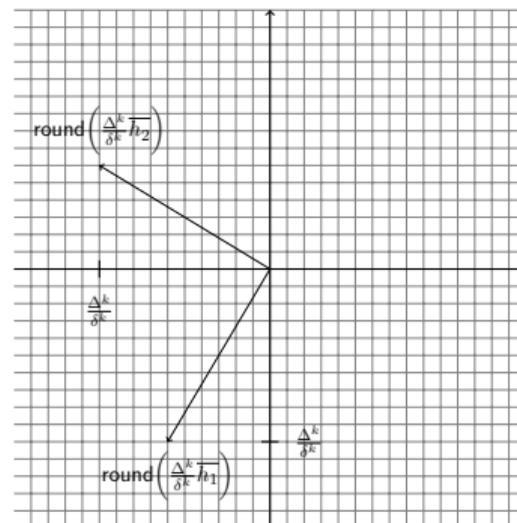
- Δ^k : Pas de cadre.
- δ^k : Pas de maillage.
- x^k : Point donnant lieu au succès courant.
- $\mathbb{D} \leftarrow \emptyset$: Ensemble de direction de sonde locale.
- $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde locale.

[1]. Génération de directions de sonde :

- Sélectionner $u \in \mathbb{R}^n$ non nulle.
- Poser $v = \frac{u}{\|u\|}$.
- Poser $H = I - 2vv^\top$.
- $\mathbb{D} \leftarrow \left\{ \pm \delta^k \text{round}\left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty}\right) : j \in \llbracket 1, n \rrbracket \right\} \subset \delta^k \mathbb{Z}^n$.

[2]. Génération de points de sonde :

- $\mathbb{P} \leftarrow \{x^k + d : d \in \mathbb{D}\}$.
 - RETOURNER** \mathbb{P} .
-



sonde locale classique

Algorithme 11 sonde locale classique - Génération de points

[0]. Initialisation :

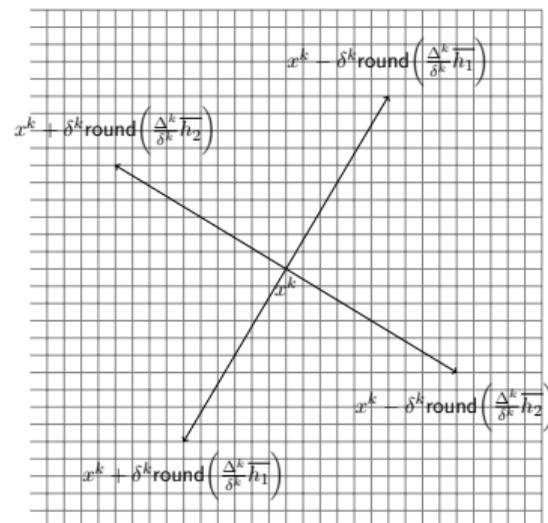
- Δ^k : Pas de cadre.
- δ^k : Pas de maillage.
- x^k : Point donnant lieu au succès courant.
- $\mathbb{D} \leftarrow \emptyset$: Ensemble de direction de sonde locale.
- $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde locale.

[1]. Génération de directions de sonde :

- Sélectionner $u \in \mathbb{R}^n$ non nulle.
- Poser $v = \frac{u}{\|u\|}$.
- Poser $H = I - 2vv^\top$.
- $\mathbb{D} \leftarrow \left\{ \pm \delta^k \text{round} \left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty} \right) : j \in \llbracket 1, n \rrbracket \right\} \subset \delta^k \mathbb{Z}^n$.

[2]. Génération de points de sonde :

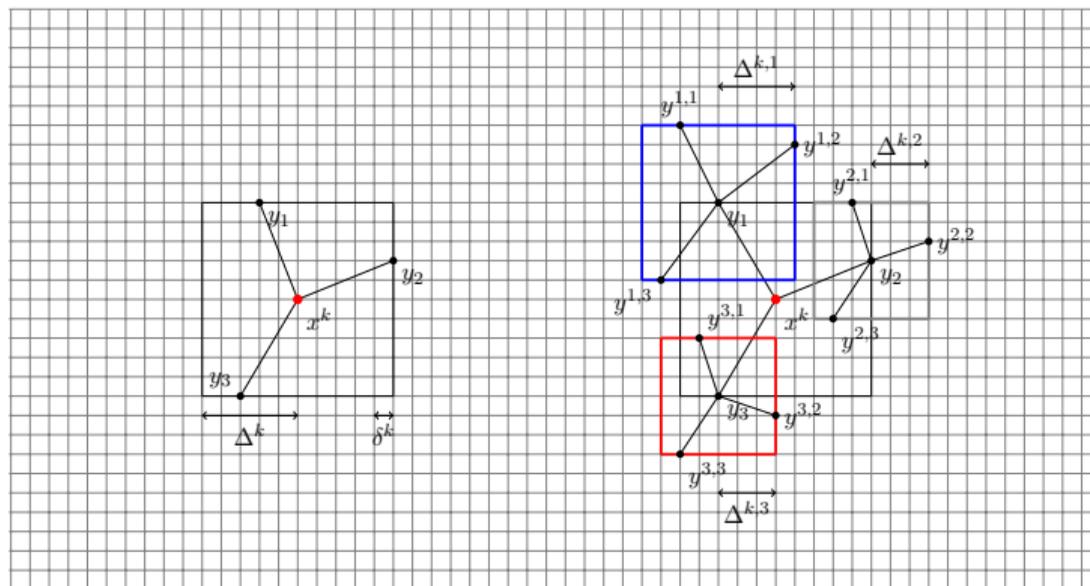
- $\mathbb{P} \leftarrow \{x^k + d : d \in \mathbb{D}\}$.
- RETOURNER** \mathbb{P} .



sonde locale classique

$$\begin{aligned} \text{CLASSICALPOLL} : \mathbb{R}_+^* \times \mathbb{R}_+^* \times \mathbb{R}^n &\rightarrow F^k \\ (\delta^k, \Delta^k, x^k) &\mapsto \mathbb{P} \end{aligned}$$

Multi sonde locale



Multi sonde locale

Algorithme 13 Multi-sonde locale - Génération de points

[0]. Initialisation :

δ^k : Pas de maillage.
 Δ^k : Pas de cadre.
 x^k : Meilleur point connu.
 $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde.

[1]. Génération de points de sonde primaire :

$\mathbb{P} \leftarrow \text{CLASSICALPOLL}(\delta^k, \Delta^k, x^k)$

[2]. Génération de points de sonde secondaire :

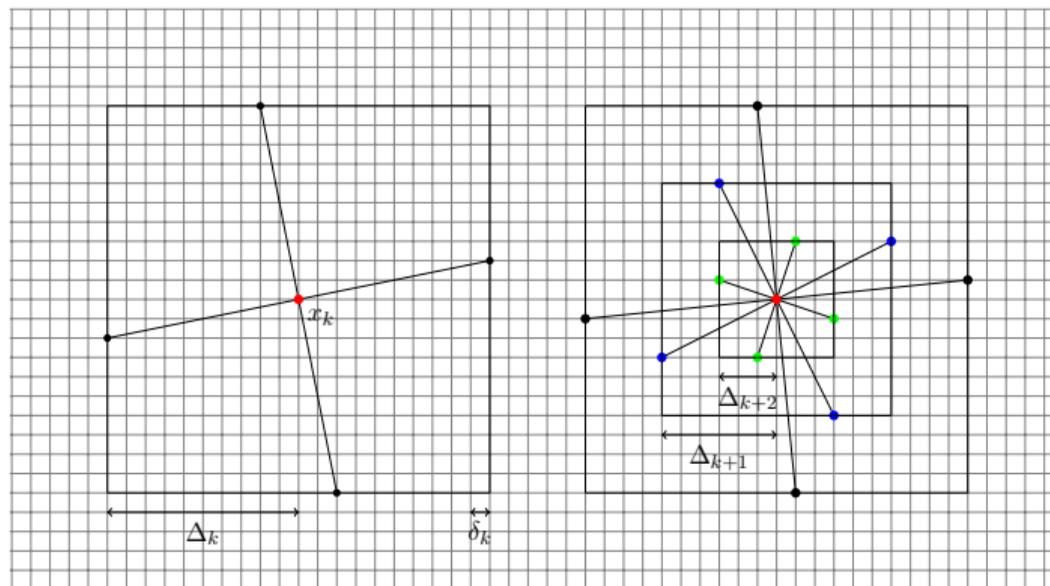
$T \leftarrow \emptyset$.
Pour $y \in \mathbb{P}$ **faire** :
 $T \leftarrow T \cup \text{CLASSICALPOLL}(\delta^k, \Delta^k, y)$.

Fin

$\mathbb{P} \leftarrow \mathbb{P} \cup T$.

RETOURNER \mathbb{P} .

sonde locale en oignon



sonde locale en oignon

Algorithme 14 sonde locale en oignon - Génération de points

[0]. Initialisation :

δ^k : Pas de maillage.
 Δ^k : Pas de sonde.
 x^k : Meilleur point connu.
 c : Nombre de couches.
 $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde.

[1]. Génération des sous cadres :

Choisir $\Gamma \subseteq \llbracket 1, \frac{\Delta^k}{\delta^k} \rrbracket$, $|\Gamma| \leq c$.

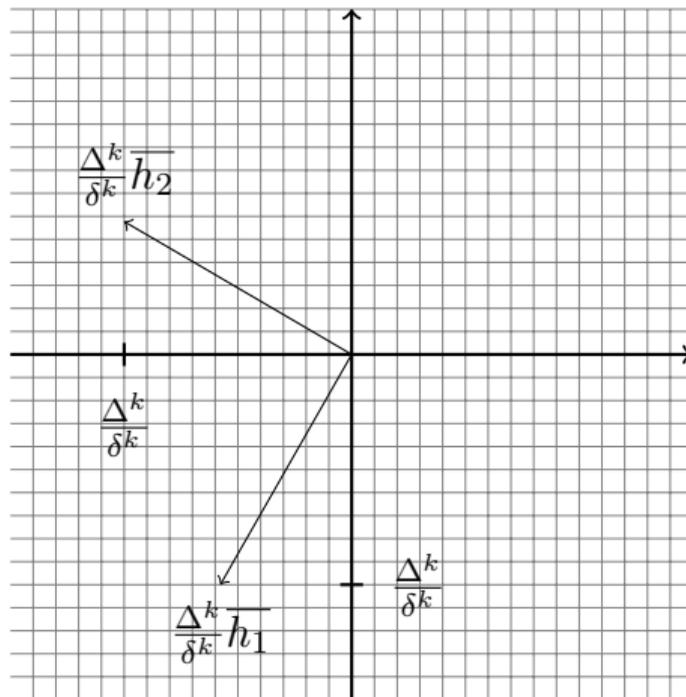
[2]. Génération de points de sonde :

Pour $\gamma \in \Gamma$ **faire** :

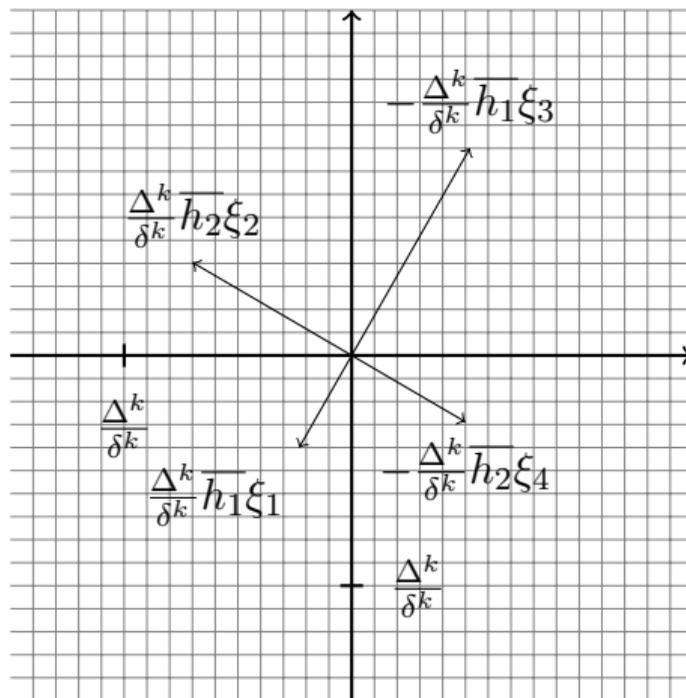
$\mathbb{P} \leftarrow \mathbb{P} \cup \text{CLASSICALPOLL}(\delta^k, \gamma\delta^k, x^k)$.

Fin**RETOURNER** \mathbb{P} .

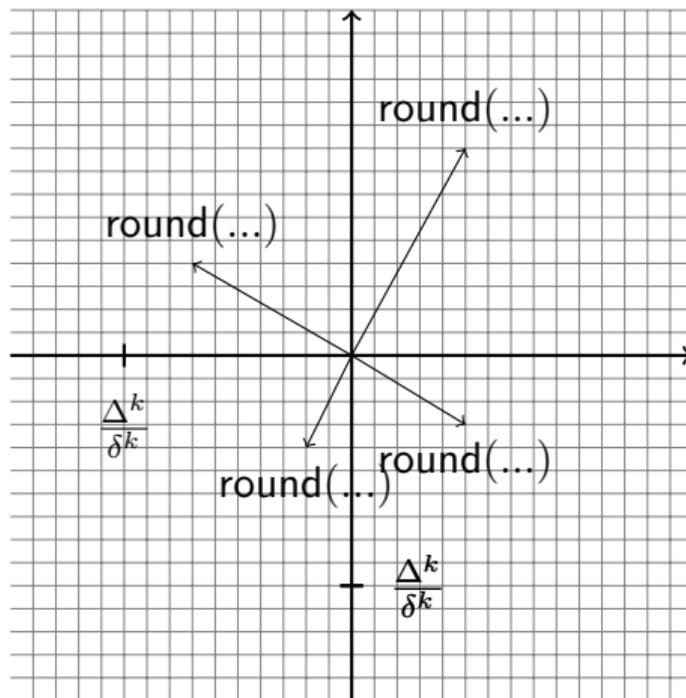
sonde locale enrichie



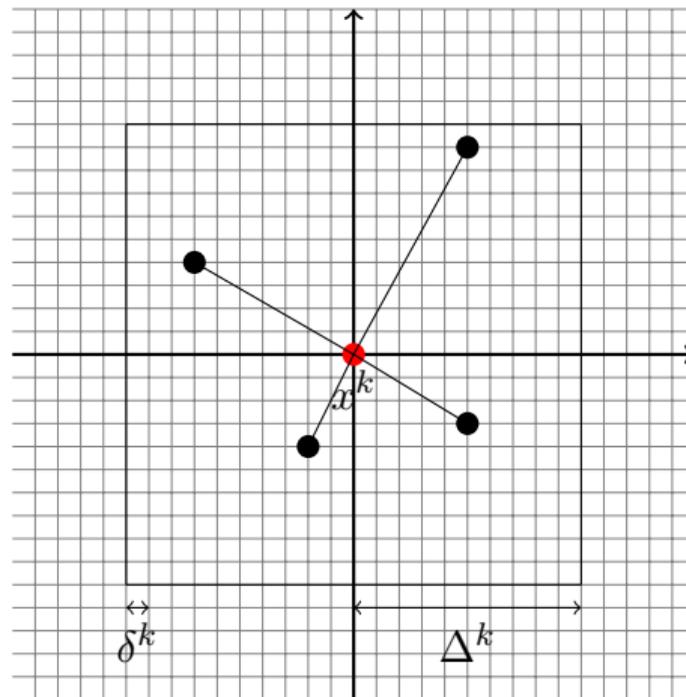
sonde locale enrichie



sonde locale enrichie



sonde locale enrichie



sonde locale enrichie

Algorithme 15 sonde locale enrichie - Génération de points

[0]. Initialisation :

Δ^k : Pas de cadre de sonde locale.
 δ^k : Pas de maillage.
 q : Nombre de bases positives à générer.
 x^k : Point donnant lieu au succès courant.
 $\mathbb{D} \leftarrow \emptyset$: Ensemble de direction de sonde locale.
 $\mathbb{P} \leftarrow \emptyset$: Ensemble de points de sonde locale.

[1]. Génération de direction de sonde locale :

Pour $i \in [1, q]$ **faire** :Sélectionner $u \in \mathbb{R}^n$ non nulle.Poser $v = \frac{u}{\|u\|_2}$.Poser $H = I - 2vv^T$.Sélectionner $\xi = (\xi_1, \dots, \xi_{2n}) \in [0, 1]^{2n}$ de manière aléatoire.
$$\mathbb{D} \leftarrow \mathbb{D} \cup \left\{ +\delta^k \text{round}\left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty} \xi_j\right) : j \in [1, n] \right\}$$

$$\cup \left\{ -\delta^k \text{round}\left(\frac{\Delta^k}{\delta^k} \frac{h_j}{\|h_j\|_\infty} \xi_{n+j}\right) : j \in [1, n] \right\}.$$
Fin

[2]. Génération de points de sonde locale :

 $\mathbb{P} \leftarrow \{x^k + d : d \in \mathbb{D}\}.$ **RETOURNER** \mathbb{P} .

Résultats numériques

Comparaison des stratégies

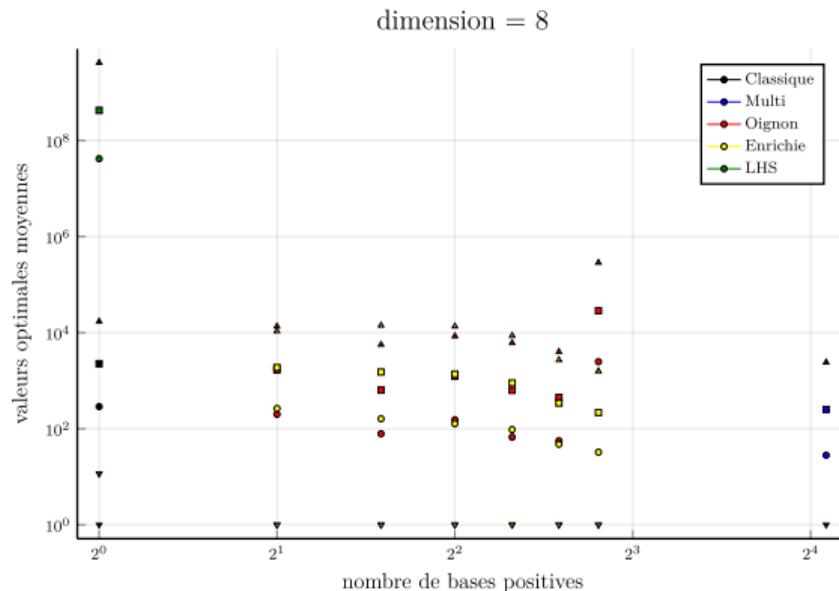


FIGURE – Valeurs moyennes, maximum, minimum et écart type

Comparaison des stratégies

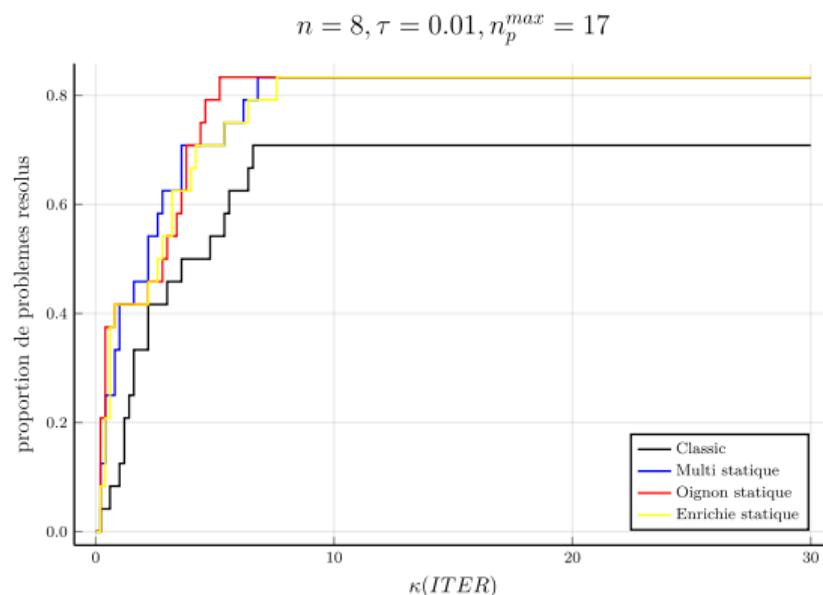


FIGURE – Profil de données Moré and Wild (2009) en itérations

Résultats numériques

- ▶ Intensifier la sonde locale améliore légèrement les résultats de l'optimisation.
- ▶ Pas de différence notable en terme de performance selon la méthode de génération de points.
- ▶ Peut-on faire mieux ?

Résultats numériques

- ▶ Intensifier la sonde locale améliore légèrement les résultats de l'optimisation.
- ▶ Pas de différence notable en terme de performance selon la méthode de génération de points.
- ▶ Peut-on faire mieux ?

Résultats numériques

- ▶ Intensifier la sonde locale améliore légèrement les résultats de l'optimisation.
- ▶ Pas de différence notable en terme de performance selon la méthode de génération de points.
- ▶ Peut-on faire mieux ?

sonde locale dynamique

- ▶ Intensifier la sonde locale que quand cela s'avère nécessaire.
- ▶
- ▶

sonde locale dynamique

- ▶ Intensifier la sonde locale que quand cela s'avère nécessaire.
- ▶ définir "nécessaire".
- ▶

sonde locale dynamique

- ▶ Intensifier la sonde locale que quand cela s'avère nécessaire.
- ▶ définir "nécessaire".
- ▶ quantifier "intensifier".

sonde locale dynamique

$$\text{DYNAMICPOLL}(\delta^k, \Delta^k, x^k, n_p^k) = \mathbb{P}^k,$$

sonde locale dynamique

$$\text{DYNAMICPOLL}(\delta^k, \Delta^k, x^k, n_p^k) = \mathbb{P}^k,$$

Comment définir n_p^k ?

sonde locale dynamique : historique sans mémoire

$$e^0 = \begin{cases} 1 & \text{Si l'étape de sonde locale à l'itération } 0 \text{ est un échec.} \\ 0 & \text{Sinon.} \end{cases}$$

$$e^k = \begin{cases} e^{k-1} + 1 & \text{Si l'étape de sonde locale à l'itération } k \\ & \text{est un échec.} \\ 0 & \text{Sinon.} \end{cases}$$

sonde locale dynamique : historique avec mémoire

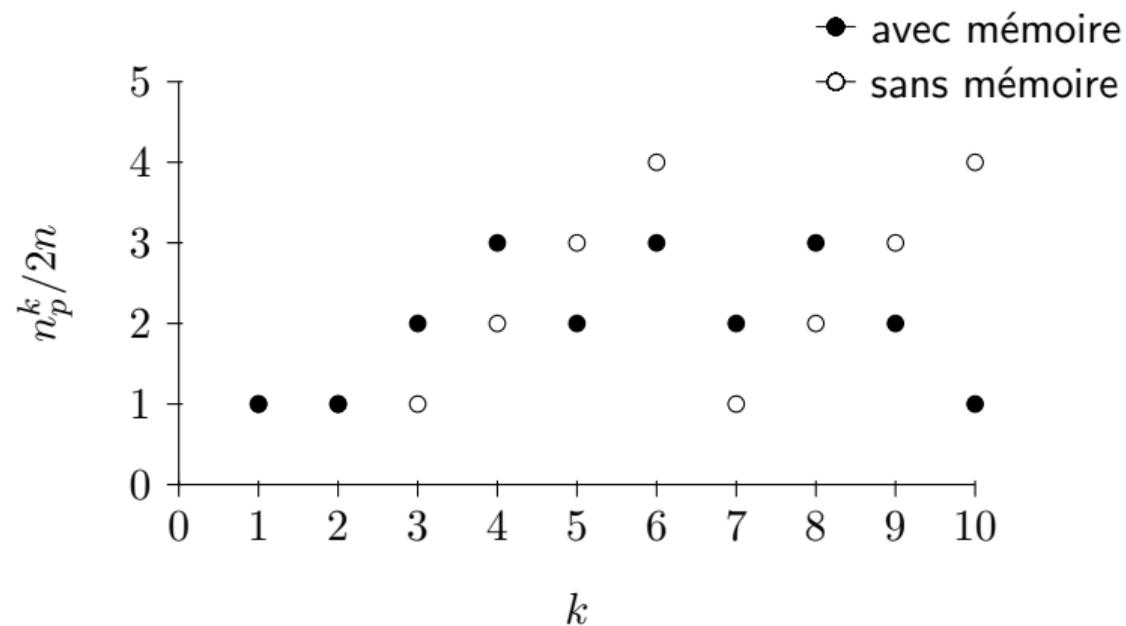
$$e^0 = \begin{cases} 1 & \text{Si l'étape de sonde locale à l'itération 0 est un échec.} \\ 0 & \text{Sinon.} \end{cases}$$

$$e^k = \begin{cases} e^{k-1} + 1 & \text{Si l'étape de sonde locale} \\ & \text{à l'itération } k \text{ est un échec.} \\ \max\{e^{k-1} - 1, 0\} & \text{Sinon.} \end{cases}$$

sonde locale dynamique

$$\begin{aligned}n_p^k &= g((e^\ell)_{1 \leq \ell \leq k-1}, n_p^{max}) \\ &= \min(1 + e^{k-1}, n_p^{max}) \\ &= \min(2^{e^{k-1}}, n_p^{max})\end{aligned}$$

Effet de la mémoire sur le nombre de points



Statique VS. dynamique sans mémoire

$$n = 4, \tau = 0.01, n_p^{max} = 8 \times 2n$$

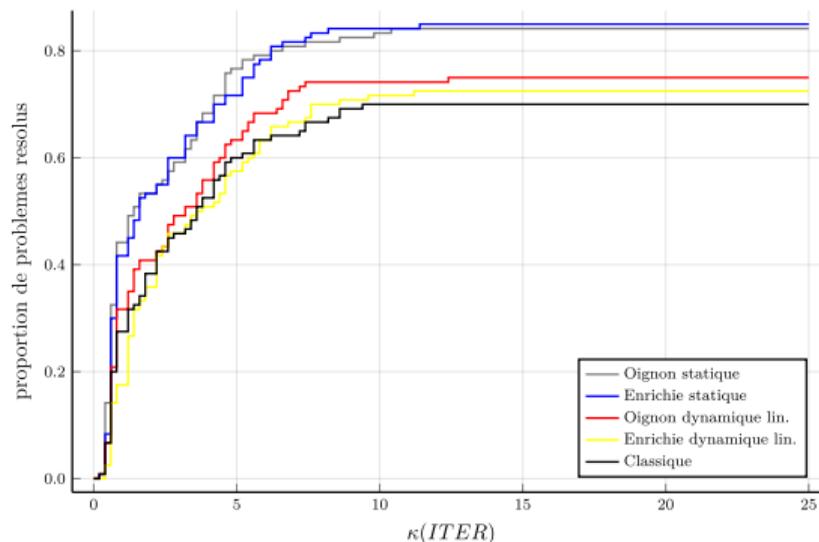


FIGURE – profile de donnée en évaluations

Travail à suivre

Travail à suivre

- ▶ Comparer les stratégies avec et sans mémoire.
- ▶ Comparer les stratégies avec différentes intensifications.
- ▶ Tracer des profils en temps.
- ▶ Étudier le comportement des stratégies sur des vrais problèmes (styrène).

Travail à suivre

- ▶ Comparer les stratégies avec et sans mémoire.
- ▶ Comparer les stratégies avec différentes intensifications.
- ▶ Tracer des profils en temps.
- ▶ Étudier le comportement des stratégies sur des vrais problèmes (styrène).

Travail à suivre

- ▶ Comparer les stratégies avec et sans mémoire.
- ▶ Comparer les stratégies avec différentes intensifications.
- ▶ Tracer des profils en temps.
- ▶ Étudier le comportement des stratégies sur des vrais problèmes (styrène).

Travail à suivre

- ▶ Comparer les stratégies avec et sans mémoire.
- ▶ Comparer les stratégies avec différentes intensifications.
- ▶ Tracer des profils en temps.
- ▶ Étudier le comportement des stratégies sur des vrais problèmes (styrène).

- Alarie, S. (2019). Use of surrogate-based model search for parallel blackbox optimization. Technical report, Hydro Québec.
- Audet, C. and Dennis, Jr., J. (2006). Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, 17(1) :188–217.
- Kokkolaras, M. and Talgorn, B. (2017). EngagePlus technical Report. Technical report, IREQ and McGill University.
- Moré, J. and Wild, S. (2009). Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1) :172–191.