Notes de travail sur la gestion du mesh avec CE

Romain Couderc

August 26, 2020

1 Cadre de travail

Afin de simplifier un peu les explications qui vont suivre, le cadre de travail sera le suivant:

- Nous travaillerons dans le cadre de l'algorithme GPS, il n'y aura donc qu'un seul treillis à considérer.
- La méthode CE sera utilisée seulement dans le but de modifier la taille du treillis à certains moments, elle ne générera aucun point.

2 Principe

Le principe de la modification de la taille du treillis par la méthode CE est le suivant: une fois qu'il y a suffisamment de point dans le cache pour calculer la moyenne et l'écart type des n_e meilleurs points, le but est de réinitialiser la taille du treillis à certaines itérations avec la valeur de la norme de l'écart type. Le point central de cette méthode est dans le choix des itérations lors desquelles la taille du treillis est modifiée.

Tout d'abord, il faut rappeler que dans le cas de CE-mads, on ne génère des points qu'à certaines itérations de l'algorithme (voir article CE-MADS pour les détails mais je rappelle l'essentiel ici). Ces itérations doivent obéir à la condition suivante: si on appelle σ_p l'écart type calculé lors du dernier passage où la recherche CE a généré des points alors la condition pour générer de nouveaux points est que :

$$||\sigma^k|| < ||\sigma^p|| \text{ avec } \sigma^k \text{ l'écart type à l'itération } k. \tag{1}$$

Dans notre cadre, la différence est que nous ne générons pas de points mais nous modifions la taille du treillis. La condition ci dessus est assez restrictive mais elle ne permet pas en l'état de garantir une convergence à l'algorithme. L'idée est donc de modifier l'actuelle condition ou de rajouter une condition C^k permettant de garantir la convergence. Le principe étant établi, le pseudo code de l'algorithme est donné ci dessous:

Algorithme 1 : Modified generalised pattern search

0. Initialisation:
   - $\delta_0 \in (0, +\infty)$ initial mesh size parameter
   - $D = GZ$ positive spanning matrix
   - $\tau \in (0, 1)$ with τ rational, the mesh size adjustement parameter
   - $\epsilon \in [0, +\infty)$ stopping tolerance
   - $k \leftarrow 0$ iteration counter
   - V^k the cache at iteration k
   - C^k a boolean at the iteration k which represents the condition to modify the size of the mesh with CE.
   - $\sigma^p = +\infty$ the previous standard deviation
1. CE step :

   if $n_e \geq card(V^k)$:

   Let E^k be the indices of the n_e best perfoming samples, then compute:

   $\mu^k = \frac{1}{n_e} \sum_{i \in E^k} x_i$

   $(\sigma^k)^2 = \frac{1}{n_e - 1} \sum_{i \in E^k} (x_i - \mu^k)^2$

   if $||\sigma^k|| < ||\sigma^p||$ and C^k:

   $\delta^k = ||\sigma^k||$

   $\sigma^p = \sigma^k$
2. Poll step:

   select a positive spanning set $\mathcal{D}^k \subset \mathcal{D}$

   if $f(t) < f(x^k)$ for some $t \in \mathcal{P}^k = \{x^k + \delta^k : d \in \mathcal{D}^k\}$:

   set $x^{k+1} \leftarrow t$ and $\delta^{k+1} \leftarrow \tau^{-1}\delta^k$

   Otherwise x^k is a mesh local optimizer:

   set $x^{k+1} \leftarrow x^k$ and $\delta^{k+1} \leftarrow \tau\delta^k$
3. Termination:

   if $\delta^{k+1} \geq \epsilon$:

   increment $k \leftarrow k + 1$ and go to 1.

   otherwise stop

3 Modification dans la preuve

3.1 Preuve en limitant le nombre de modifications du mesh par CE

Dans cette section, je m'appuie sur la preuve de convergence concernant l'algorithme GPS que tu donnes dans ton livre. Le premier aspect que cette modification concerne le theorème 7.4 de la page 123 qui traduit le fait que le treillis devient infiniment fin. La preuve de ce théorème repose sur le fait que pour n'importe quel entier $N \geq 1$, on a (je reprends les notations du livre):

$$x^N = x^0 + \sum_{k=0}^{N-1} \delta_k D z^k = x^0 + \delta^0 D \sum_{k=0}^{N-1} (\tau)^{r^k} z^k \tag{2}$$

Dans le cas où CE réinitialise la taille du treillis à certaines itérations, cette formule est remplacé par:

$$x^N = x^0 + \sum_{i=1}^{I} ||\sigma^i|| D \sum_{k=0}^{N_i-1} (\tau)^{r^k} \tag{3}$$

avec $I \in [0, N-1]$ le nombre de fois que la taille du treillis a été changée et les N_i tel que $\sum_{i=1}^{I} N_i = N$. Pour garantir la convergence de l'algorithme la manière la plus simple est de s'assurer que le nombre de fois qu'on modifie la taille du treillis avec CE est fini. En effet, dans ce cas là on retombe directement sur la convergence de l'algorithme GPS car on peut considérer que l'on a juste effectué une phase de recherche de solutions avant d'utiliser l'algorithme GPS non modifié. Pour garantir que I est fini, on peut choisir la condition C^k suivante:

$$C^k = k < N_c \tag{4}$$

avec N_c l'itération jusqu'à laquelle on peut effectuer des modifications de la taille du treillis avec CE. Une autre condition garantissant un nombre de modifications finis (trouvé lors de mon stage de fin d'étude) était:

$$C^k = ||\sigma_{stop}|| < ||\sigma|| \text{ avec } \sigma_{stop} \in [0; 1) \tag{5}$$

En effet, on avait réussi à montrer que la suite des écart types tendait vers 0. Cela limite donc fatalement le nombre de modifications de la taille du treillis. Cependant, ces deux solutions bien qu'utilisables ne sont pas mathématiquement très "esthétique".

3.2 Résultats numériques

L'idéal serait de conserver la propriété de convergence sans imposer une limite fini au passage dans l'étape de modification par CE. Est ce possible ? La réponse n'est pas évidente, néanmoins les résultats numériques de la Figure 1 incitent à y consacrer plus de temps.

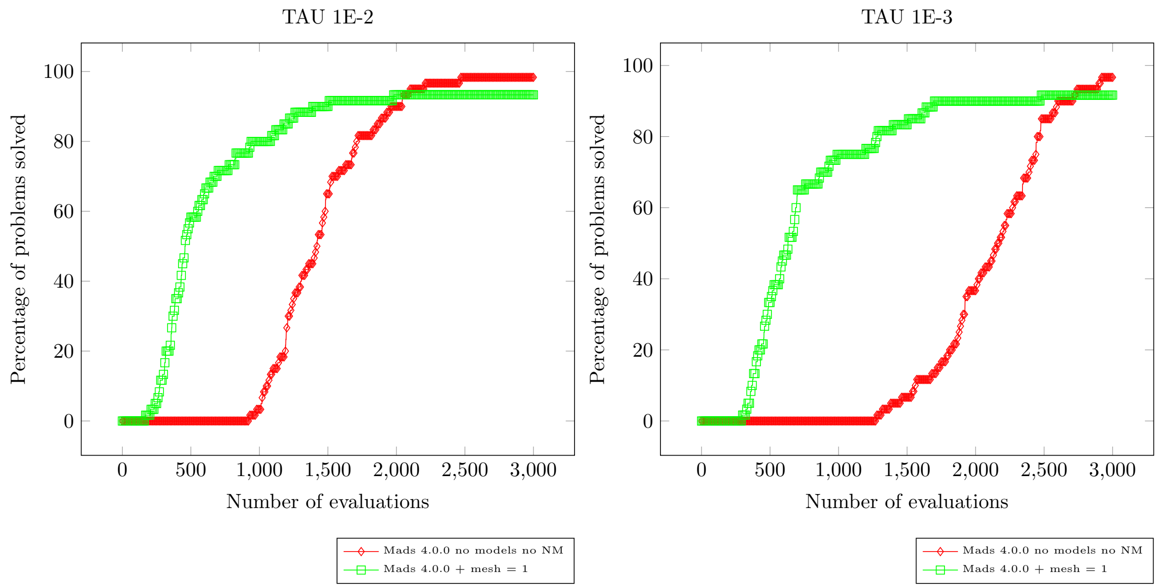

Figure 1 – Comparaison pour $\tau = 10^{-2}$ et $\tau = 10^{-3}$ entre mads et mads avec modification du mesh par CE (dans les deux cas il n'y a pas d'autres phases de recherche que la spéculative).

Pour tracer ces courbes, j'ai enlevé toutes les étapes de recherches possibles à part la spéculative. De plus, je n'ai pas rajouté de condition C^k, mais j'ai un peu modifié la condition de passage dans la modification du treillis par CE (en m'inspirant d'un résultat trouvé ci dessous) qui est devenu:

$$||\sigma^k|| < \frac{||\sigma^p||}{2} \tag{6}$$

Le résultat le plus impressionnant est obtenue quand on rajoute une phase initial où on génère des points avec la méthode de recherche CE, mais cela seulement jusqu'à ce que la taille du cache soit dépasse la dimension du problème. On obtient le résultat de la figure 2. Le fait de générer des points par CE juste au début de la minimisation du problème permet peut être de mieux ajuster la taille du treillis au départ ce qui améliore encore les résultats. Enfin, le dernier résultat figure 3 est obtenu avec l'utilisation de l'algorithme de mads avec la méthode de recherche CE. Les résultats restent une amélioration mais le fait de générer des points semblent nuire un peu à l'efficacité de l'algorithme, en effet pour MDO, il y a peu d'intérêt à explorer l'espace.

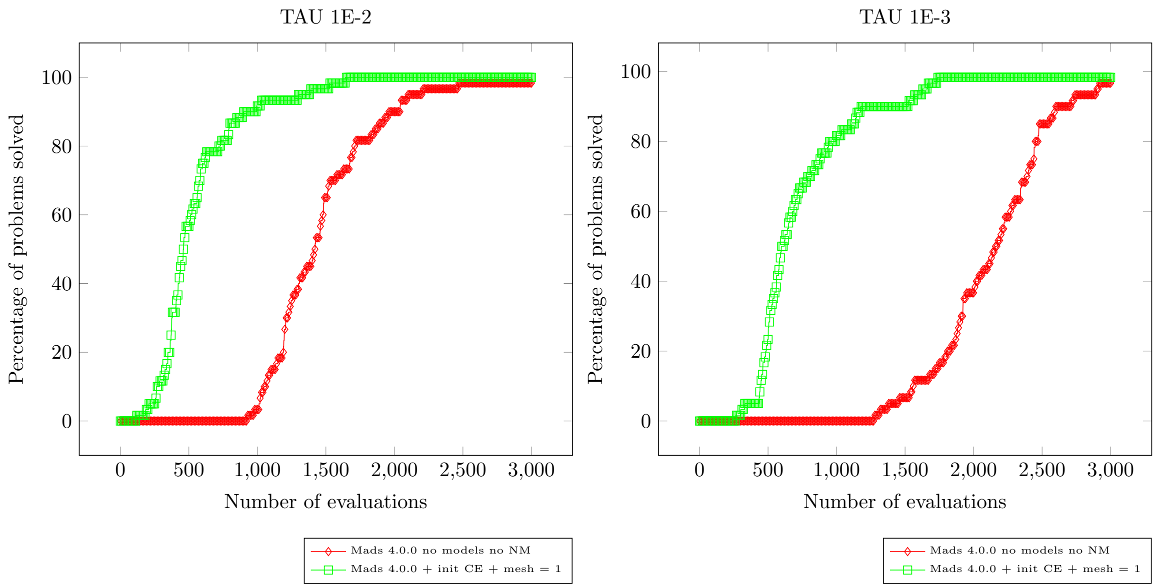

Figure 2 – Comparaison pour $\tau = 10^{-2}$ et $\tau = 10^{-3}$ entre mads et mads avec modification du mesh par CE et une recherche initial de CE.

3.3 Début de réflexion sur la preuve

Le premier résultat que nous perdons avec la modification du treillis par CE est le résultat du théorème 7.4 de la page 123 sur le fait que le treillis devienne infiniment fin. L'idée que j'ai pour conserver ce résultat est la suivante:

- Notons $i \in \mathbb{N}$ la ième modification du treillis par CE et $k_i \in \mathbb{N}$ l'itération à laquelle cela se produit.
- A cette itération, on a donc σ_i qui est l'écart type entre les meilleurs points à l'itération k_i et on a donc (puisqu'il y a modification du treillis à cette itération):

$$\delta^{k_i} = ||\sigma_i|| \tag{7}$$

- L'idée est alors de se dire que, la $i+1$ ème modification du treillis avec CE ne pourra avoir lieu que si :

$$\delta^{k_{i+1}} \leq \tau\delta^{k_i} \tag{8}$$

  où $\tau \in (0, 1)$ est le paramètre d'ajustement de GPS et $\delta^{k_{i+1}}$ est la taille du treillis au moment de la $i+1$ ème modification.

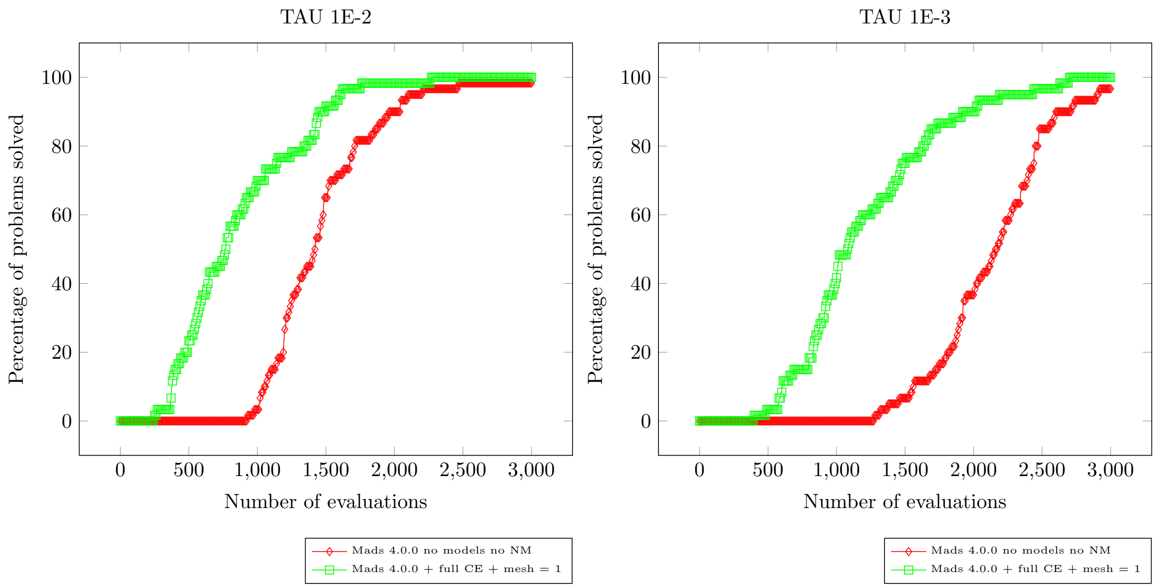

Figure 3 – Comparaison pour $\tau = 10^{-2}$ et $\tau = 10^{-3}$ entre mads et CE-mads (sans autre méthode de search) avec modification du mesh par CE.

Pour moi cette condition permettrait d'assurer que la taille du treillis tende bien vers 0, on peut appliquer le lemme 7.3 page 123, à chaque réinitialisation du treillis par CE. C'est à dire que pour tout $k \in [k_i + 1, ..., k_{i+1} - 1]$, on a:

$$\delta^k \leq \delta^{k_i}(\tau)^{\hat{z}_i} \tag{9}$$

avec $\hat{z}_i$ un entier négatif (il y a une petite erreur dans le livre je l'avais déjà signalé à Charles pendant le stage). En notant $M = \max\ (\tau)^{\hat{z}_i}$, on a, pour tout $k \in [k_i + 1, ..., k_{i+1} - 1]$;

$$\delta^k \leq M\delta^{k_i} \tag{10}$$

Ensuite, en utilisation la condition de passage que l'on a établi en 8, on a cette fois pour tout $k \in \mathbb{N}$:

$$\delta^k \leq M\delta^0\tau^i \tag{11}$$

En faisant, tendre i vers l'infini on a $\lim_{i\to\infty} \tau^i = 0$, donc par le théorème d'encadrement (δ^k étant positif), on a $\lim_{k\to\infty} \delta^k = 0$.

Le problème qui se pose désormais c'est que la condition 8 doit être remplie, néanmoins changer de taille à chaque fois qu'elle est remplie reviendrait à passer dans la méthode CE très souvent ce qui pourrait être problématique dans le cas où on l'utiliserait également pour générer des points. C'est pour cela que l'on aimerait condition respectant 8 mais plus restrictive. Le plus simple serait de trouver une condition proche de celle écrite dans l'algorithme 1. Prenons la condition suivante:

$$||\sigma^{k_{i+1}}|| \leq C_0 \tau ||\sigma_i^k|| \tag{12}$$

avec C_0 une constante (si elle existe) telle que pour tout $k \in [k_i + 1, ..., k_{i+1}]$:

$$C_0 \delta^k \leq ||\sigma^k||. \tag{13}$$

Dans ce cas là, la condition 8 serait remplie car, on aurait:

$$C_0 \delta^{k_{i+1}} \leq ||\sigma^{k_{i+1}}|| \leq C_0 \tau ||\sigma^{k_i}|| = C_0 \tau \delta^{k_i} \tag{14}$$

Pour prouver que la constante C_0 de la relation 13 existe on peut se baser sur le lemme 7.2 de la page 121 du livre, on a pour tout $k \in [k_i + 1, ..., k_{i+1}]$, i. e. pour toutes les itérations comprises entre les deux changement du treillis par CE, la distance minimale entre deux points du treillis qui est minoré par:

$$\min_{u \neq v \in M^k} ||u - v|| \geq \frac{\delta^k}{||G^{-1}||} \tag{15}$$

avec G une matrice inversible. Je ne vais écrire ici qu'une preuve dans le cas $n_e = 2$ mais en se basant sur la preuve du Lemme 7.2, je pense que c'est généralisable avec n_e plus grand. Pour $n_e = 2$, on a (les carrés ici sont à prendre composant par composant):

$$\begin{aligned}
(\sigma^k)^2 &= \left(X_1 - \frac{X_1 + X_2}{2}\right)^2 + \left(X_2 - \frac{X_2 + X_2}{2}\right)^2 && (16)\\
&= \frac{1}{2}(X_1 - X_2)^2 && (17)
\end{aligned}$$

Donc en prenant la racine composant par composant de $(\sigma^k)^2$, puis la norme et en utilisant le lemme 7.2, on obtient:

$$||\sigma^k|| = \frac{1}{\sqrt{2}} ||X1 - X2|| \geq \frac{1}{\sqrt{2}} \frac{\delta^k}{||G^{-1}||} \tag{18}$$