Optimisation des hyperparamètres d'un auto-encodeur variationnel pour la détection d'anomalies sur les données du Mars science laboratory

Dounia Lakhmiri
Sébastien Le Digabel
Ryan Shahrouz Alimo

GERAD
Jet Propulsion Laboratory

Table of contents

Introduction

Contexte

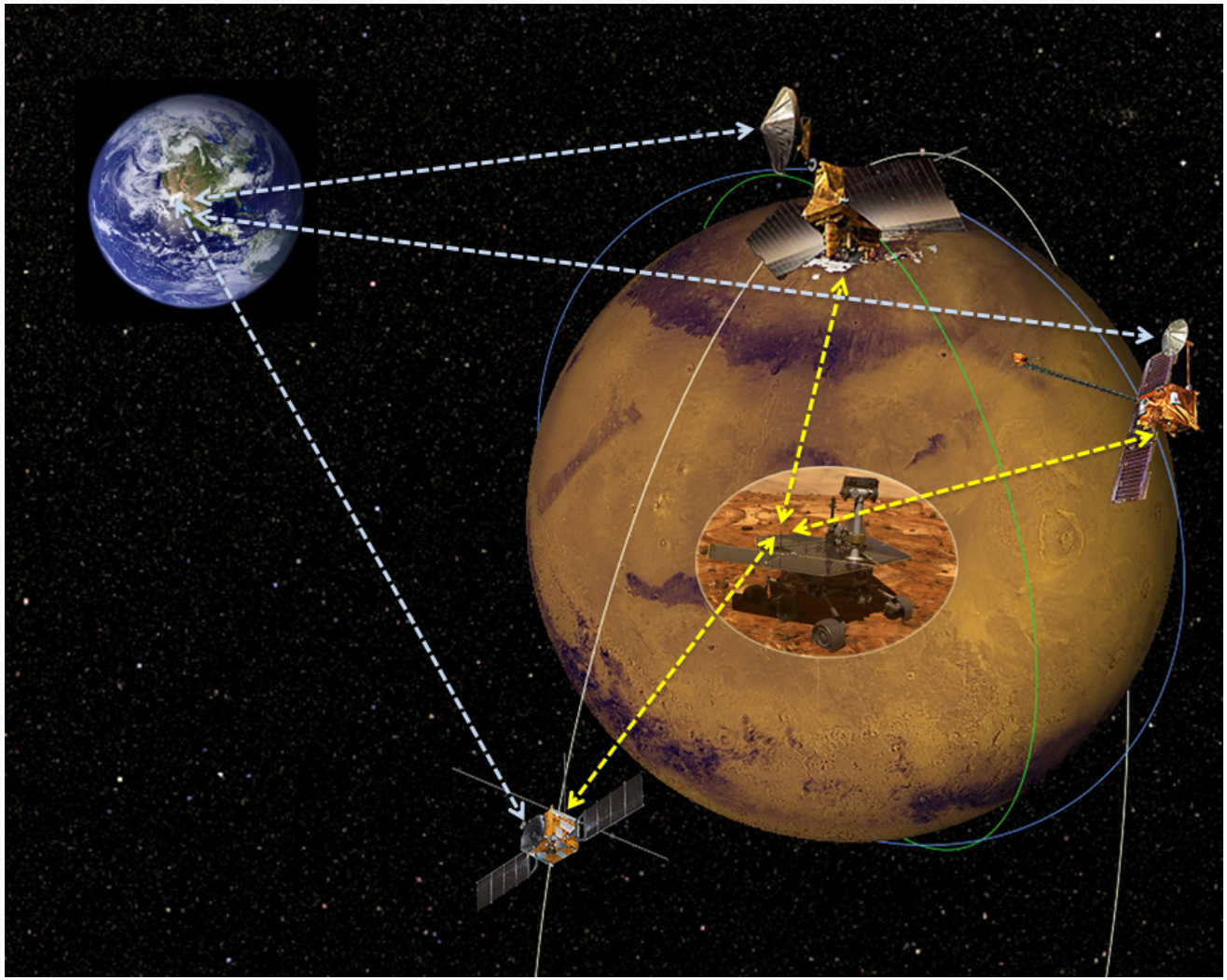
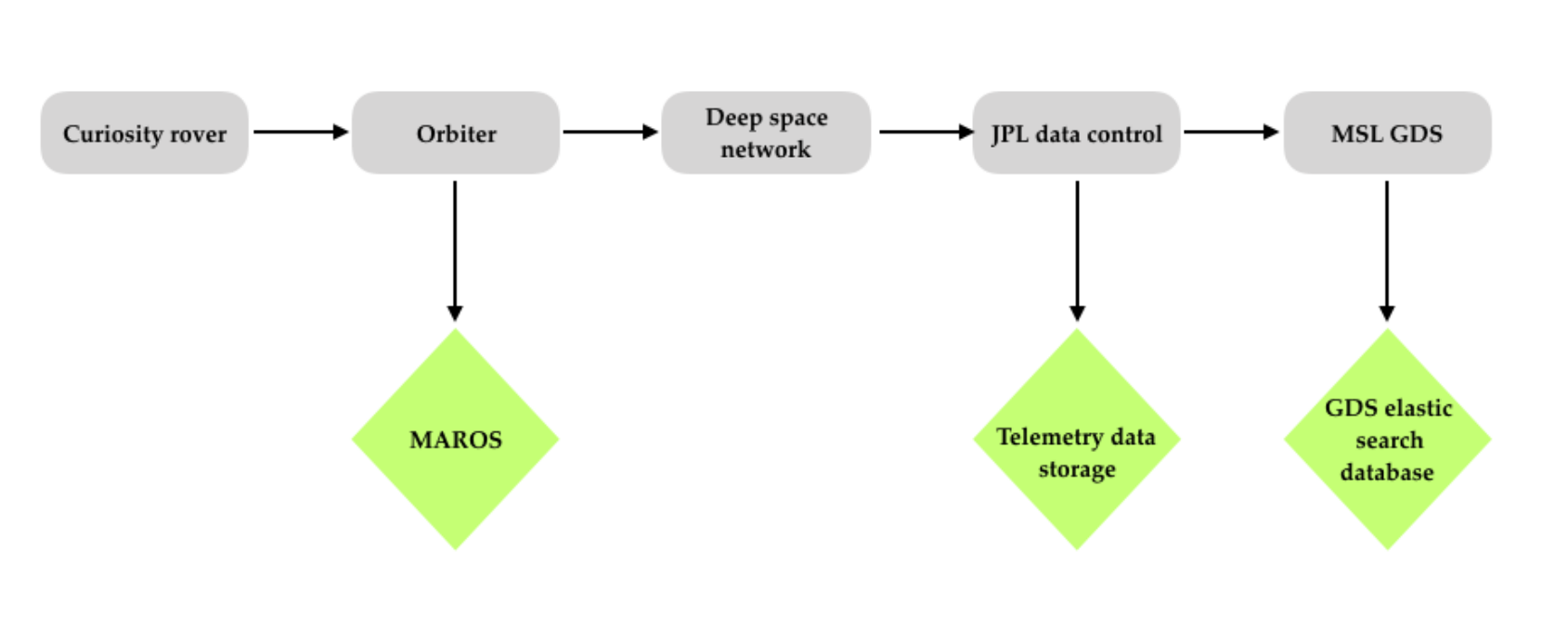


Image provenant de: NASA/JPL

Contexte

Une perte ou une corruption de données peut se produire lors de leur transfert du robot Curiosity au Mars Science Lab lorsqu'elles transitent par le pipeline suivant:



Procédure de détection

- Les données sont collectées et regroupées au MSL.
- Elles sont étiquetées manuellement par un expert.
- La détection d'anomalies se fait par un expert qui peut se faire aider des prédictions de RFs entrainnées sur des données étiquetées au préalable.

Objectif:
Trouver un moyen de détecter les passes incomplètes de manière non supervisée.

Base de données

Base de données

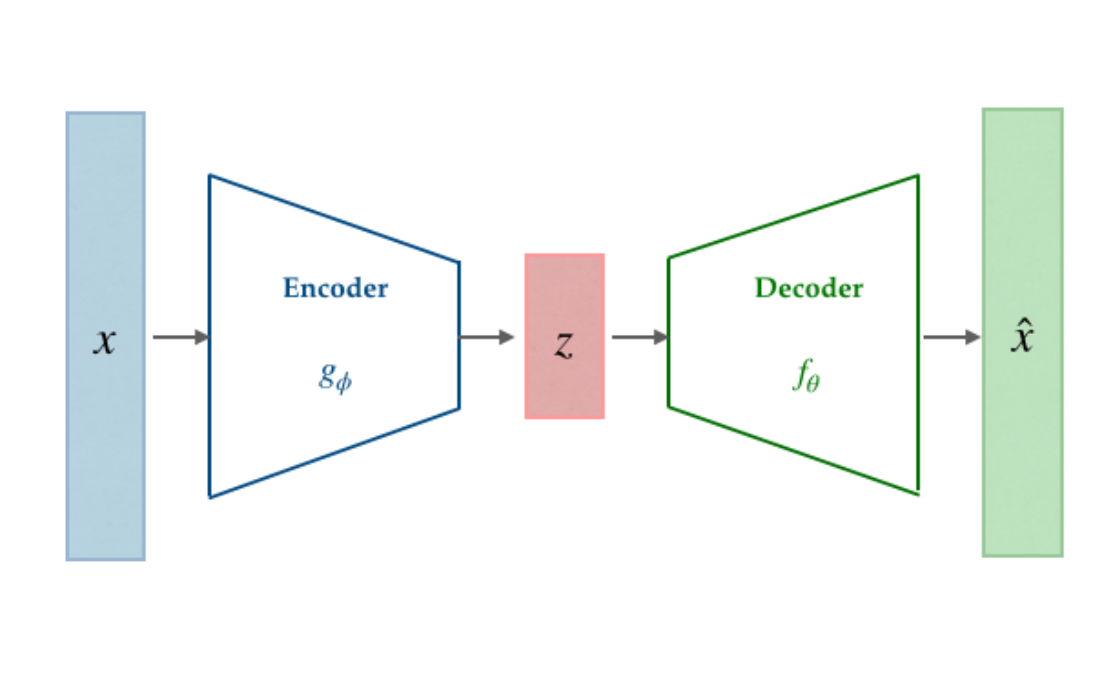
Aspects pertinents sur la base de données

- On a 6 satellites: MRO, ODY, MEX, MVN, TGO et la possibilité d'une transmission directe: DTE.
- Chaque vecteur d'entrée comprend 43 éléments: réels, entiers and booléens.
- Empiriquement on constate un taux de 13% de passes incomplètes.

Remarque:
Ce problème est équivalent à une classification non supervisée sur des données disproportionnées.

Apprentissage non supervisé

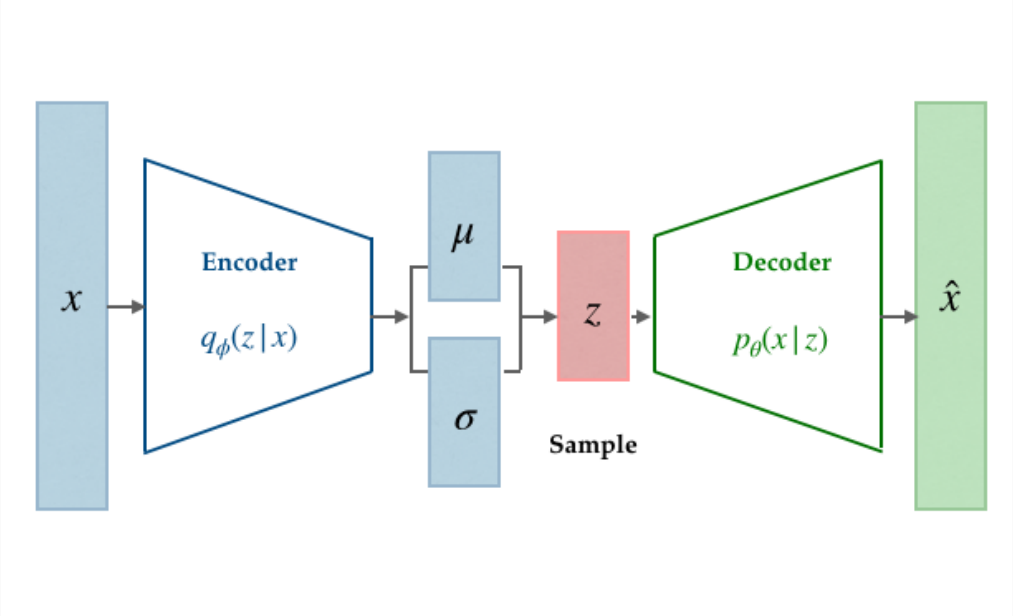
Auto-encodeur (AE)



Function de coût

$$L = ||\hat{x} - x||. \tag{1}$$

auto-encodeur variationnel (VAE)



Function de coût

$$L = ||\hat{x} - x|| + D_{KL}(q_\phi(z|x)||p(z)). \tag{2}$$

où $p(z) \sim N(0, I)$ et D_{KL} est la divergence de Kullback-Liebler qui mesure la dissimilarité entre deux distributions de probabilités:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) log \frac{p(x)}{q(x)} dx \tag{3}$$

Premiers tests

Détection d'anomalies avec AEs ou VAEs

Pour un seuil $\alpha \in \mathbb{R}$, $\forall x$, si $L(x) < \alpha$ alors x est une passe complète sinon x est incomplète.

Table 1: Performances de quelques méthodes non supervisées.

	GDS labeler		KMEAN		Gaussian mixture		AE		VAE	
	Cpl.	Inc.	Cpl.	Inc.	Cpl.	Inc.	Cpl.	Inc.	Cpl.	Inc.
Precision	0.94	0.74	0.08	0.40	0.26	0.30	0.55	0.74	0.84	0.75
Recall	0.97	0.55	0.01	0.86	0.15	0.46	0.63	0.62	0.79	0.80
$F1$ score	**0.95**	0.63	0.02	0.55	0.19	0.36	0.58	0.67	0.81	**0.77**

Hyperparamètres d'un VAE

Architecture d'un réseau de neurones

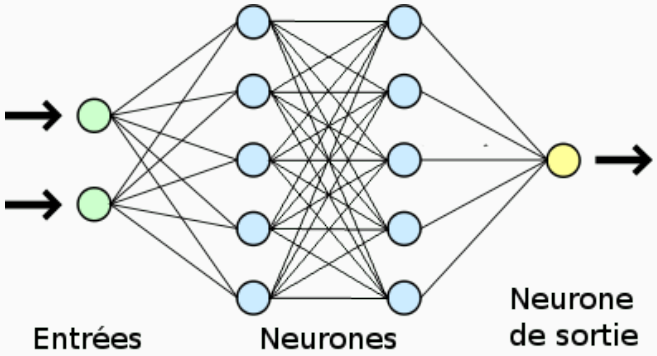


Figure 1: Architecture d'un réseau de neurones.

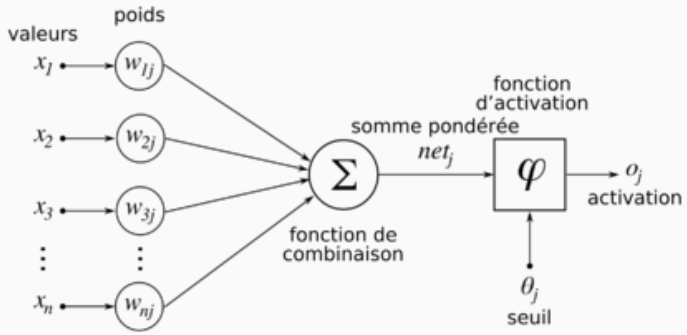
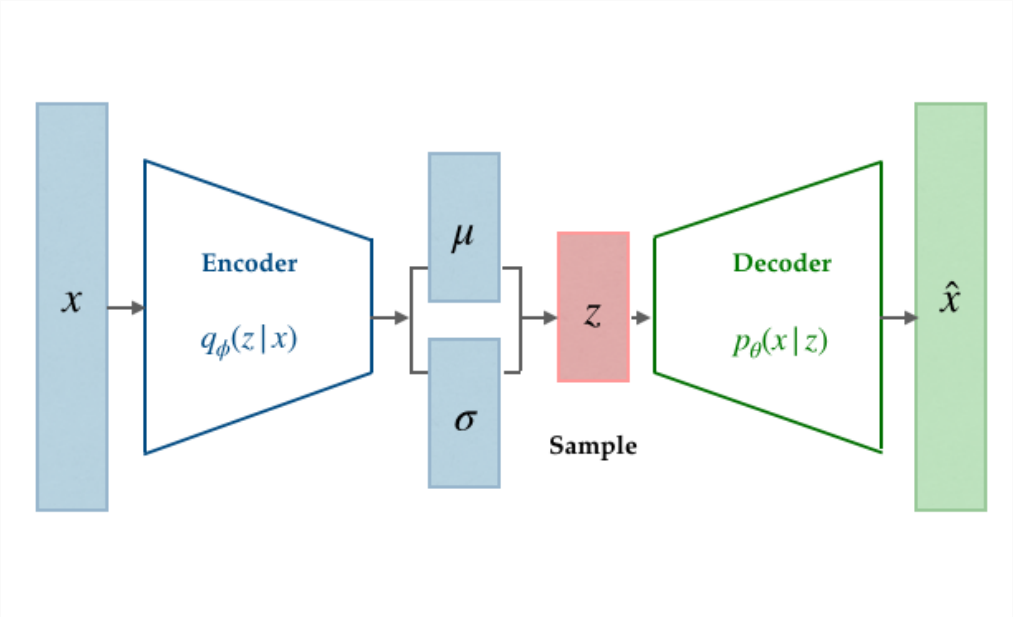
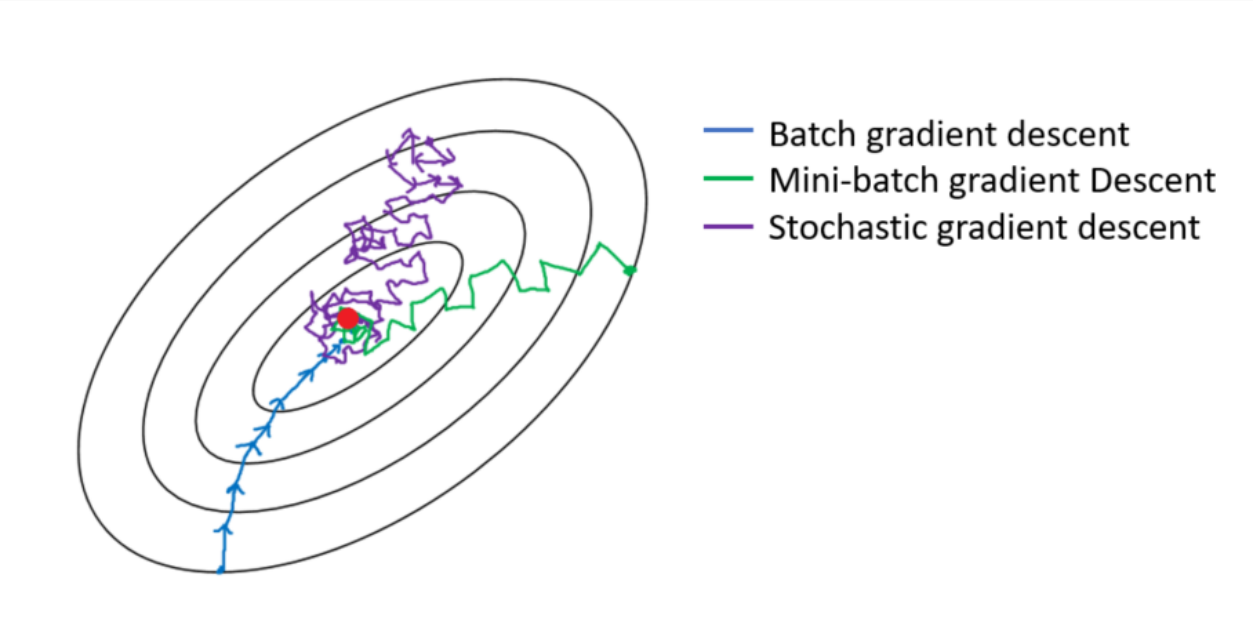


Figure 2: Fonctionnement d'un neurone.



Optimisation de la fonction de coût

Quelques variantes de la méthode du gradient



Il existe d'autres variantes : Adam, Adagrad, RMSProp...

Optimisation de la fonction de coût

Table 2: Hyperparameters related to the training of the VAE.

Optimizer	**Hyperparameter**	**Type**	**Range**
Stochastic Gradient Descent (SGD)	Initial learning rate	Réel	[0;1]
	Momentum	Réel	[0;1]
	Dampening	Réel	[0;1]
	Weight decay	Réel	[0;1]
Adam	Initial learning rate	Réel	[0;1]
	β_1	Réel	[0;1]
	β_2	Réel	[0;1]
	Weight decay	Réel	[0;1]
Adagrad	Initial learning rate	Réel	[0;1]
	Learning rate decay	Réel	[0;1]
	Initial accumulator	Réel	[0;1]
	Weight decay	Réel	[0;1]
RMSProp	Initial learning rate	Réel	[0;1]
	Momentum	Réel	[0;1]
	Smoothing constant	Réel	[0;1]
	Weight decay	Réel	[0;1]

Hyperparamètres

Table 3: Hyperparamètres considérés pour ce problème.

Hyperparameter	**Type**	**Range**
Nombre de couche d'encodage	Entier	$[1, 50]$
Dimension de la couche centrale	Entier	$[1, n_0[$
Taille du batch	Entier	$[10, 512]$
Fonction d'activation	De catégorie	1 : ReLU, 2 : Sigmoid, 3 : Tanh.
Dropout rate	Réel	$[0, 1]$
Choix de l'optimiseur	De catégorie	1 : SGD, 2 : Adam. 3 : Adagrad. 4 : RMSProp.
4 HPs de l'optimiseur	Réel	$[0, 1]$
Seuil α	Réel	$[0.50, 1]$

MADS

LE Mesh Adaptive Direct Search (MADS) est une méthode DFO de recherche directe où l'on définit un treillis M_k à chaque itération k:

$$M_k = \{x + \Delta_k^m Dz, z \in^{n_D}, x \in C\},$$

où

- C est la cache où sont enregistrées les évaluations précédentes.
- la matrice $D \in^{n \times n_D}$ a des colones qui forment un ensemble générateur positif,
- $\Delta_k^m \in^+$ est la taille du treillis à l'itération k.

À chaque itération
On effectue deux étapes : la recherche et la sonde.

MADS - Sonde

$$P_k = \{x_k + \Delta_k^m d \mid d \in D_k\} \text{ où } ||\Delta_k^m d|| \approx \Delta_k^p,$$

Avec

- Δ_k^p est le pas de sonde, sachant que $\Delta_k^m = \mathcal{O}(\sqrt{\Delta_k^p})$,
- les colones de la matrice D_k forment une base positive.

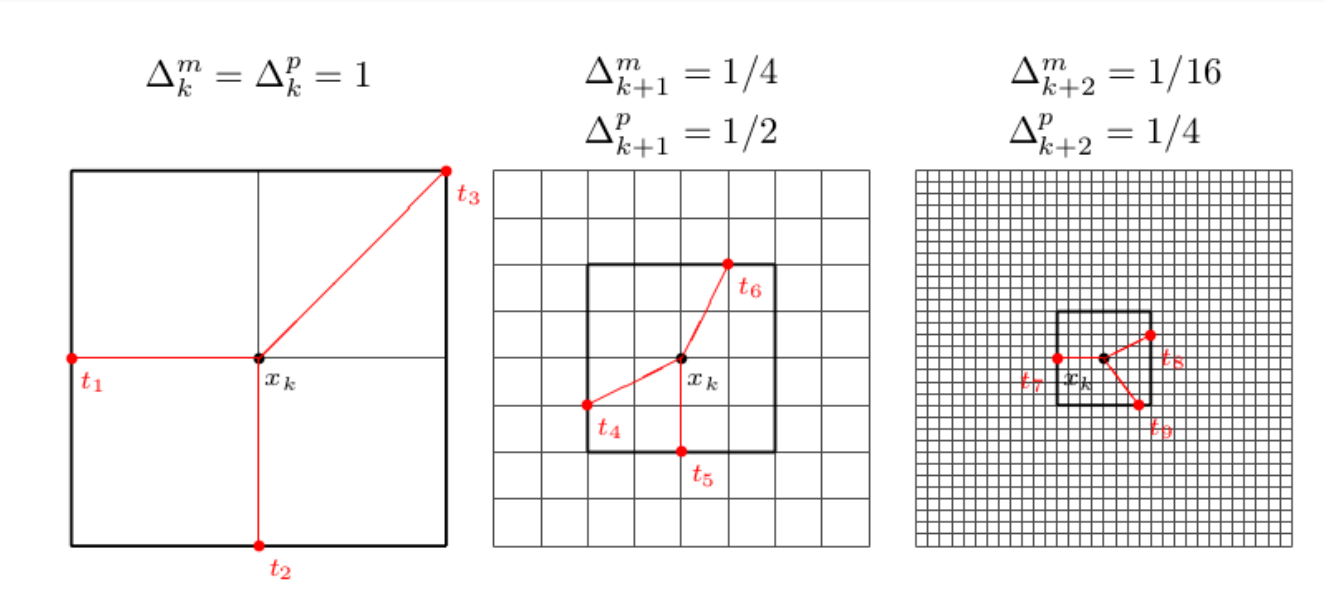


Figure 3: Exemple de sonde de MADS. [2]

Δ-DOGS

Delaunay-based derivative-free optimization via global surrogates est une famille de méthodes DFO basées sur les fonctions substituts.

Soit

- f la fonction objectif,
- p une fonction d'interpolation,
- e la fonction d'incertitude calculée sur la base de la triangularisation de Delaunay,
- f_0 la valeur cible,
- et la fonction de recherche $s = \begin{cases} \frac{p(x)-f_0}{e(x)} & \text{si } p(x) \geq f_0 \\ p(x) - f_0 & \text{sinon.} \end{cases}$

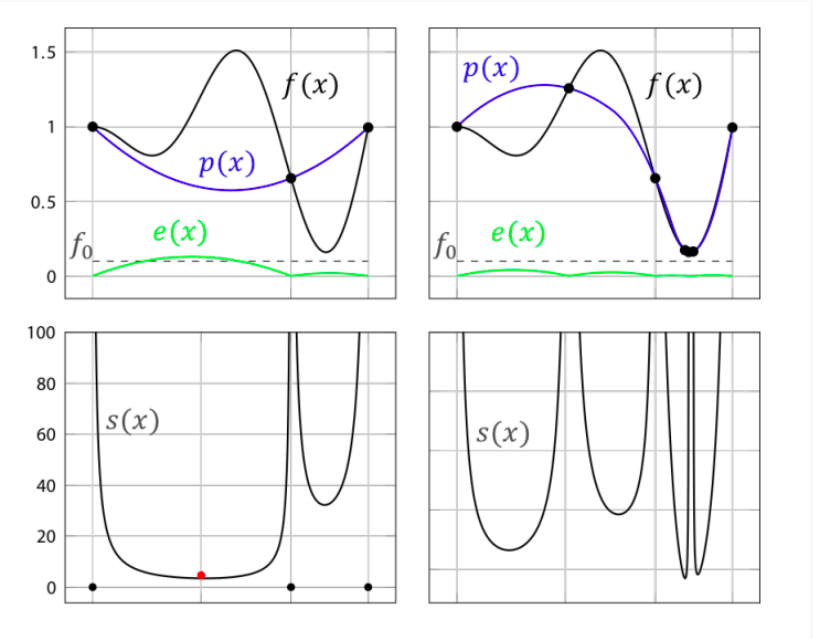


Figure 4: Mécanisme de Δ-DOGS. Image from [1]

Δ-DOGS

Algorithm 1: Δ-DOGS pour minimiser $f(x)$ en ciblant la valeur f_0.

0. Initialiser $k = 0$, l'ensemble de points S_0, et calculer $f(x_i)$ pour $x_i \in S_0$.
1. Mettre à jour la fonction d'interpolation $p_k(x)$ sur tous les points de S_k.
2. Mettre à jour la triangularisation de Delaunay Δ^k sur tous les points de S_k.
3. $\forall$ simplexe Δ_i^k de la triangularisation Δ^k :
   - Calculer le centre du cercle circonscrit z_i^k et son rayon r_i^k,
   - définir la fonction d'incertitude locale: $e_i^k(x) = (r_i^k)^2 - ||x - -z_i^k||^2$,
4. Minimiser la fonction de recherche pour obtenir $\hat{x}_k$ comme étant le point qui a la plus forte chance d'atteindre la valeur cible.
5. Si $\hat{x}_k \notin S_k$, $S_{k+1} = S_k \cup \hat{x}_k$, et évaluer $f(\hat{x}_k)$; sinon on incrémente .
6. Répéter les étapes jusqu'à ce qu'un point x tel que $f(x) \leq f_0$ est trouvé.

Δ-MADS

Algorithme Δ-MADS

Algorithm 2: Δ-MADS: Hybride entre MADS et Δ-DOGS

initialiser: $x_0, y_0, \epsilon \in]0, 1[$;

while *not stop* **do**

  Recherche: Fixer les valeurs entières et catégoriques du point courant x_k^N et utiliser Δ-DOGS sur le sous problème aux variables réelles x_k^R avec pour valeur cible y_k ;

  retourner le nouveau point $\hat{x_k} = x_k^N \cup x_k^R$;

  Sonde: Sonde d'HyperNOMAD (MADS) avec comme point de départ $\hat{x_k}$;

  Soit f_k la meilleure valeur de l'objectif ;

  if $f_k < y_k$ **then**

    $y_{k+1} = y_k - \epsilon$;

  else

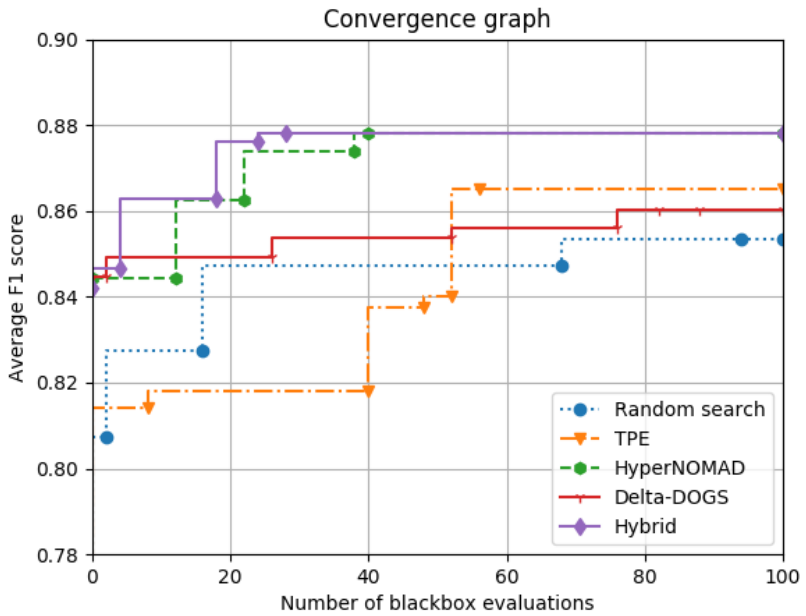
    $y_{k+1} = y_k + \epsilon$;

  end

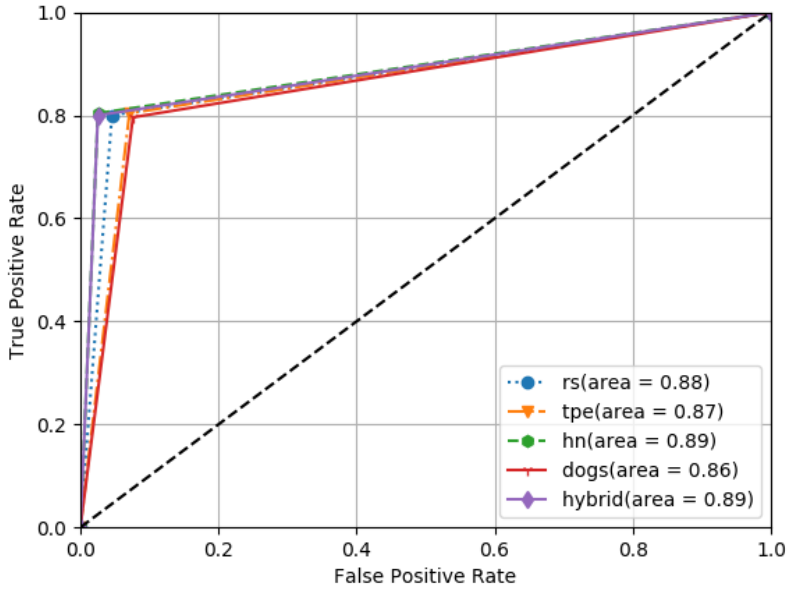
end

Résultats numériques

Résultats numériques



(a) Convergence de chaque méthode DFO.



(b) ROC curve de la meilleure configuration trouvée par chaque algorithme DFO.

Conclusion

Points forts

- Détection d'anomalies non supervisée,
- Δ-MADS obtient de meilleurs résultats plus rapidement.

Points faibles

- Les scores sur les passes complètes et incomplètes dépendent fortement de la valeur du seuil α. Il est donc difficile de dépasser la barre des 90% sur les deux simultanément.

Questions?

References

[1] S. R. Alimo, P. Beyhaghi, and T. Bewley. Optimization combining derivative-free global exploration with derivative-based local refinement. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2531–2538. IEEE, 2017.

[2] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):1–15, 2011.