

# Implémentation d'une variante régression symbolique de l'algorithme Black-DROPS

Vladislav Tempez

Université Rennes 1, École Normale Supérieure de Rennes

27 juin 2018

encadré par Jean-Baptiste Mouret (LORIA, équipe Larsen, Nancy)

# Apprentissage en robotique

- ▶ Donner aux robots la capacité d'apprendre et de s'adapter à leur environnement.
- ▶ Surmonter une panne partielle, un terrain non prévu, un changement de tâche à accomplir.
- ▶ Objectif : développer un algorithme qui permette à un robot d'apprendre une tâche spécifique ou de s'adapter à un environnement changeant



## Formulation du problème

# Apprentissage par renforcement

Apprentissage à partir de séquences d'états, de commandes et de récompenses.

$(x_t, u_t, r_t)_t$ .

- ▶ Spécification de la tâche via la récompense. Processus par essai erreur.
- ▶ But : trouver une fonction de commande du système (ou politique)  $\pi : x \mapsto u$  qui maximise l'espérance de la récompense  $J^1$ .

$$J = \mathbb{E}\left(\sum_{t=1}^T r_t\right)$$

- ▶ Paramétrage possible de cette politique par  $\theta \in \mathbb{R}^p$  noté  $\pi_\theta$  (e.g. poids d'un réseau de neurones)

---

1. Sutton et Barto 2011. "Reinforcement learning : An introduction".

Approche classique pour la recherche de  
politique basée modèles.

# Apprendre des modèles dynamiques

Hors simulation, chaque essai est coûteux.

- ▶ Nécessité de réduire leur nombre au strict minimum.
- ▶ Apprentissage d'un modèle du système pour y déléguer la majorité des essais.

Système dynamique  $S$  ; évolution est gouvernée par la fonction  $f$ .

- ▶  $x_{t+1} = x_t + f(x_t, u_t) + w$ .
- ▶ Bruit supposé gaussien noté  $w$ .

Apprendre un modèle de  $S$  : construire une approximation  $\hat{f}$  de  $f$ .

Puis optimiser la récompense selon le modèle :

$$J = \mathbb{E}\left(\sum_{t=1}^T r(x_{t-1} + \hat{f}(x_{t-1}, \pi_\theta(x_{t-1})) + w)\right)$$

# Recherche de politique basée modèle

- ▶ Apprentissage d'un modèle : supervisé
- ▶ Apprentissage de la politique : renforcement
- ▶ Alternance entre raffinement du modèle et recherche de politique

---

## **Algorithm 1** Recherche de politique basée modèle

---

Mouvements aléatoires. Collecte de données sur la dynamique  $X$  et la récompense  $R$

**while** Problème non résolu **do**

    Mettre à jour un modèle dynamique avec les données  $X$ .

    Mettre à jour un modèle de récompense avec les données  $R$ .

    Apprendre la politique  $\pi_\theta$  via le modèle.

    Essayer la nouvelle politique sur le robot et collecter des nouvelles données.

**end while**

---

## Processus gaussiens

Apprentissage paresseux (cf k-plus-proches-voisins).

Prédictions faites par combinaisons linéaires des exemples d'apprentissage similaires<sup>2</sup>.

- ▶ Prédiction = évolution du système et incertitude sur cette prédiction
- ▶  $f(x_t, u_t) = K((x_t, u_t), X)K_X^{-1}Y$
- ▶ Coûts importants :
  - ▶ prédition en  $O(\text{nombre d'exemples}^2)$
  - ▶ mémoire en  $O(\text{nombre d'exemples}^2)$
  - ▶ mise à jour du modèle = une inversion de matrice ( $O(\text{nombre d'exemples}^3)$ )

# L'algorithme Black-DROPS

L'algorithme Black-DROPS<sup>3</sup> (Black-box Data-efficient RObotics Policy Search) est une refonte de l'algorithme PILCO<sup>4</sup>.

- ▶ Modèle du système : processus gaussien.
- ▶ Politique : réseau de neurones ou processus gaussien.
- ▶ Tire parti de l'incertitude du modèle.
- ▶ Parralélisable (recherche de la politique avec CMA-ES<sup>5</sup>).
- ▶ Gère les récompenses non dérivables.

---

3. Chatzilygeroudis et al. 2017. "Black-Box Data-efficient Policy Search for Robotics".

4. Deisenroth, Fox et Rasmussen 2015. "Gaussian processes for data-efficient learning in robotics and control". *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

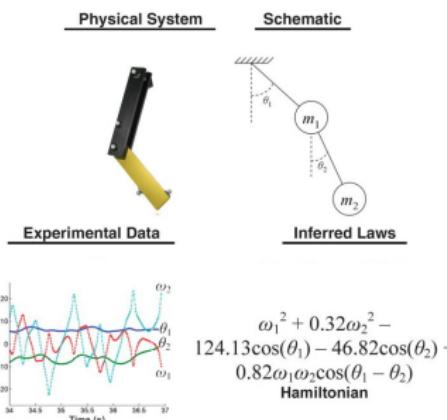
5. Hansen et Ostermeier 2001. "Completely derandomized self-adaptation in evolution strategies". *Evolutionary computation*.

# Une approche alternative pour l'apprentissage de modèles

# Régression symbolique

Processus gaussiens difficiles à lire par un expert. Pourquoi pas des formules comme modèle ?

- ▶ Riches et interprétables
- ▶ Évaluation rapide pour les opérateurs de base (+,-,×,÷,min,max,cos,etc.)
- ▶ Utilisées avec succès pour redécouvrir automatiquement des dynamiques connues<sup>5</sup>
- ▶ Pas évident d'explorer l'espace des formules
- ▶ Pas de gestion de l'incertitude des prédictions



Un double pendule utilisé dans (5)

5. Michael Schmidt et Hod Lipson (2009). "Distilling free-form natural laws from experimental data". In : *Science* 324.5923, p. 81–85

# Algorithmes évolutionnistes

Heuristiques d'optimisation de boites noires (i.e. la fonction à optimiser est une boite noire).

- ▶ Inspirés de la théorie darwinienne :
  - ▶ sélectionner les meilleures valeurs dans une population
  - ▶ les combiner pour obtenir une nouvelle population
  - ▶ recommencer jusqu'à satisfaction
- ▶ Gère des espaces sans métrique (e.g. l'espace des formules<sup>5</sup>)

Choix cruciaux :

- ▶ méthode de sélection
- ▶ manière de combiner les individus

---

5. Michael Schmidt et Hod Lipson (2009). "Distilling free-form natural laws from experimental data". In : *Science* 324.5923, p. 81–85

# Une version régression symbolique de Black-DROPS

Black-DROPS est l'état de l'art mais

- ▶ Requêtes de processus gaussiens coûteuses
- ▶ Processus gaussiens non interprétables

L'objectif de ce stage était de concevoir une variante à Black-DROPS

- ▶ Modèles interprétables
- ▶ Plus rapide

Pour cela

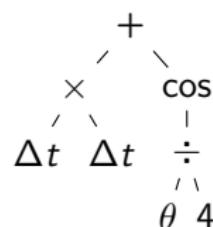
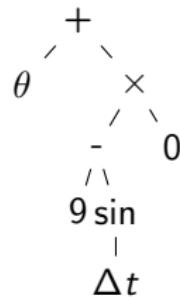
- ▶ Remplacer les processus gaussiens par des formules
- ▶ Explorer l'espace des formules à l'aide d'algorithmes évolutionnistes

## Détails de l'approche implémentée

# Formules

Les formules sont représentées par des arbres.

- ▶ Noeuds : opérateurs  $(+, -, \times, \div, \cos, \sin)$ .
- ▶ Feuilles : variables  $(x, u)$  et constantes  $(0-9, \Delta t, g)$ .
- ▶ Utilisées pour apprendre un modèle, supervisé
- ▶ Utilisées pour apprendre une politique (ajout de *if*, *min*, *max*), renforcement.



Arbre de  $\Delta t^2 + \cos(\frac{\theta}{4})$

# Algorithme évolutionniste, sélection des formules

NOMBREUSES FORMULES SÉMANTIQUEMENT ÉQUIVALENTES E.G.

$$\theta + 0 \times \Delta t = \frac{5\theta}{5} - u + u$$

- ▶ CONVERGENCE PRÉMATURE DE LA POPULATION.
- ▶ SÉLECTION DE FORMULES INUTILEMENT LONGUES.

UTILISATION DE PLUSIEURS OBJECTIFS ET DE TECHNIQUES DE MAINTIEN DE DIVERSITÉ DES INDIVIDUS.

- ▶ MINIMISER L'ERREUR (OU MAXIMISER LA RÉCOMPENSE POUR LES POLITIQUES).
- ▶ MINIMISER LA HAUTEUR DES ARBRES DE FORMULES.
- ▶ FAVORISER LES FORMULES DIFFÉRENTES (ERREUR POUR DES ÉTATS DIFFÉRENTS, SÉQUENCE DE COMMANDES DIFFÉRENTES).

UTILISATION DE L'ALGORITHME NSGA-II<sup>8</sup> POUR SÉLECTIONNER SELON CES 3 OBJECTIFS SANS COMPROMIS (FRONT DE PARETO).

# Protocole expérimental

Comparaison des modèles seuls avec les représentations :

- ▶ Processus gaussiens(GP)
- ▶ Réseau de neurones(NN)
- ▶ Formule(SR)

Évaluation sur un ensemble de test indépendant de l'ensemble d'apprentissage.

Comparaison de tous les couples modèle/politique (9) dans Black-DROPS.

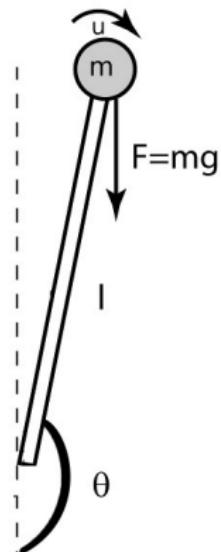
Deux tâches d'évaluation : pendule inversé et cart-pole (20 répliques pour chaque tâche).

Comparaison des politiques seules (via un modèle idéal) avec les représentations :

- ▶ Processus gaussiens(GP)
- ▶ Réseau de neurones(NN)
- ▶ Formule(SR)

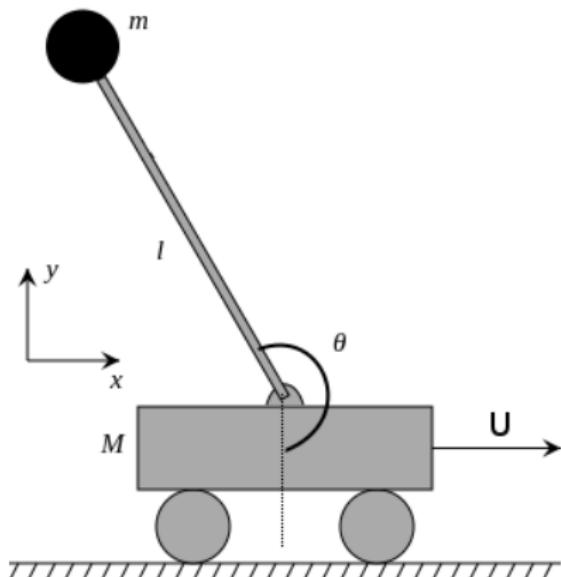
# Tâche d'évaluation - Pendule Inversé

- ▶ But : maintenir un pendule en position verticale  $\theta = \pi$ .
- ▶ Commande : moment angulaire appliqué au pendule  $u \in [-2.5, 2.5]$ .
- ▶ Sous actionné.
- ▶ État :  $(\cos(\theta), \sin(\theta), \dot{\theta})$ .
- ▶ Récompense :  
 $r=\exp(-\text{distance à l'objectif})$ .
- ▶ Tâche classique en apprentissage, utilisé par Black-DROPS.



## Tâche d'évaluation - Cart-pole

- ▶ But : maintenir un pendule monté sur un chariot à la verticale  $\theta = \pi$
- ▶ Commande : force  $f \in [-10, 10]$  appliquée sur le chariot en avant ou en arrière.
- ▶ Complètement actionné
- ▶ État :  $(x, \dot{x}, \cos(\theta), \sin(\theta), \dot{\theta})$
- ▶ Récompense :  $r = \exp(-\text{distance à l'objectif})$
- ▶ Tâche classique en apprentissage, utilisé par Black-DROPS



# Bibliothèques utilisées

- ▶ Implémentation en C++
- ▶ Black-DROPS implémenté dans limbo<sup>9</sup>
- ▶ NSGA-II implémenté dans Sferes2<sup>10</sup>
- ▶ Ajout d'un module de représentation arborescente à Sferes2
- ▶ Utilisation de tiny-dnn<sup>11</sup> pour comparaison avec des réseaux de neurones.
- ▶ Comparaison avec une autre bibliothèque de régression symbolique (fastsr<sup>12</sup>)

---

9. Cully et al. 2016. "Limbo : A Flexible High-performance Library for Gaussian Processes modeling and Data-Efficient Optimization". *Preprint*.

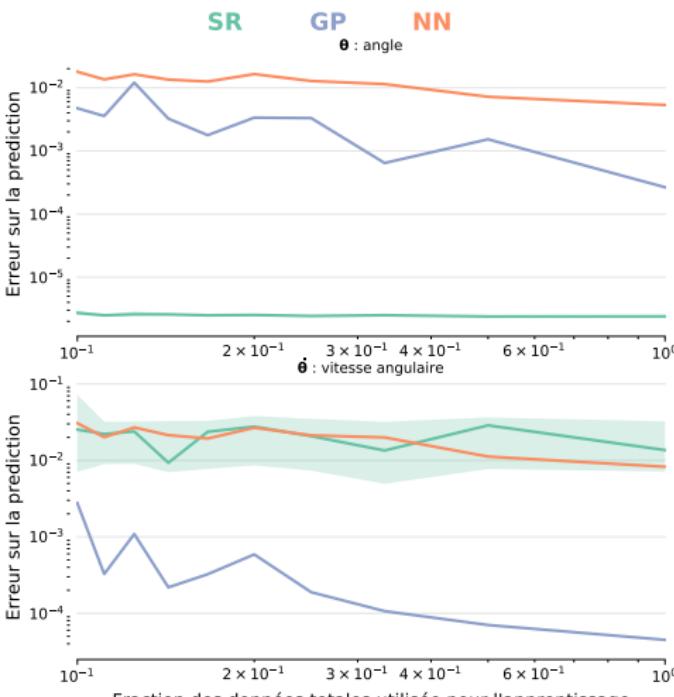
10. Mouret et Doncieux 2010. "SFERESv2 : Evolvin' in the Multi-Core World".

11. <https://github.com/tiny-dnn/tiny-dnn>

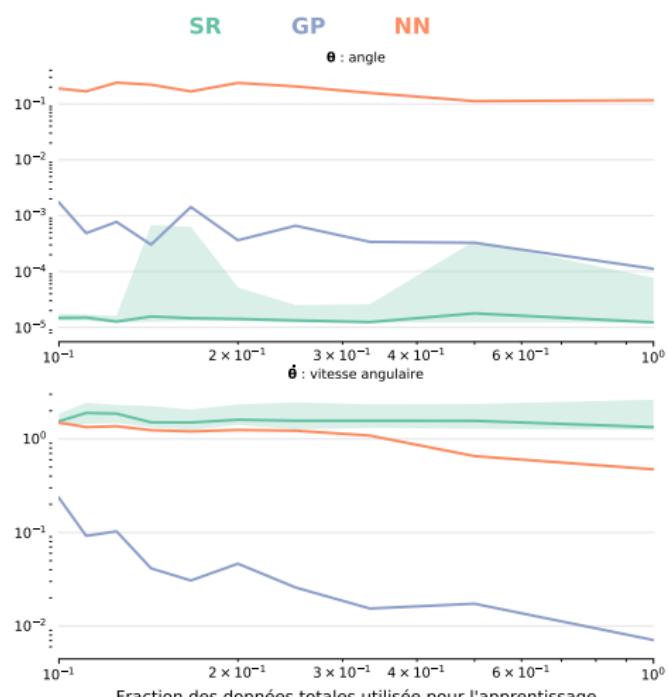
12. <https://github.com/cfusting/fast-symbolic-regression>

## Résultats obtenus

# Efficacité des modèles sur un ensemble de test indépendant

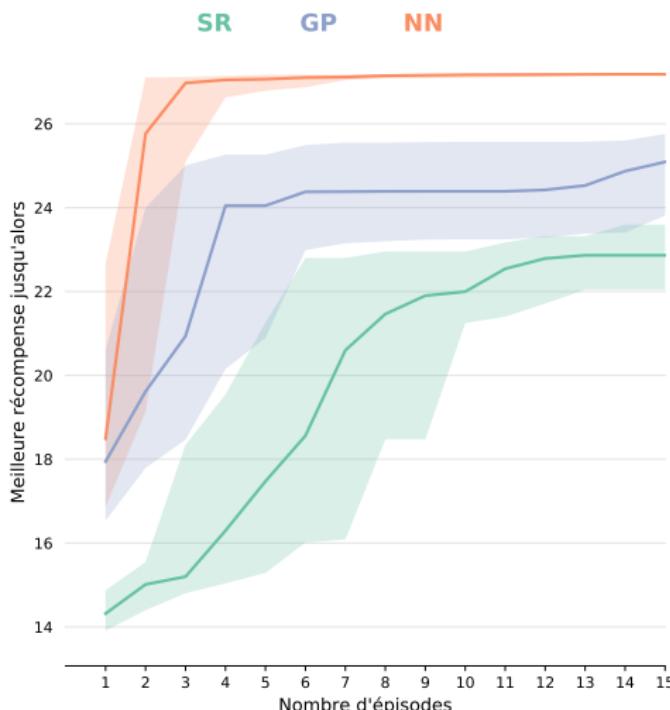


Pendule, 20 répliques

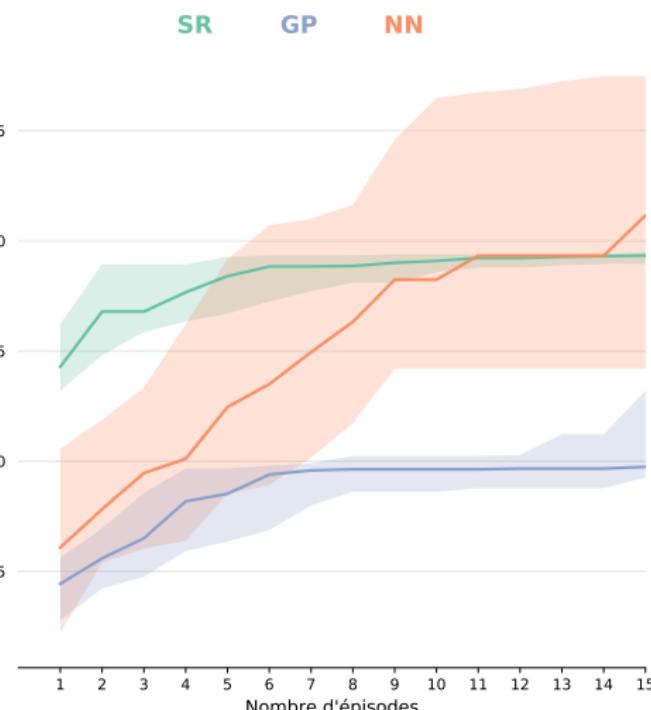


Cart-pole, 20 répliques

# Efficacité des politiques

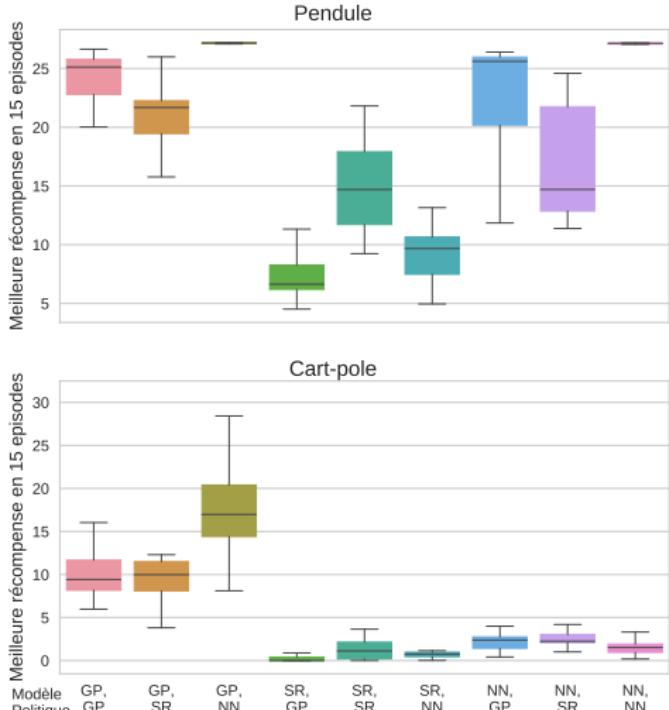


Pendule, 20 répliques

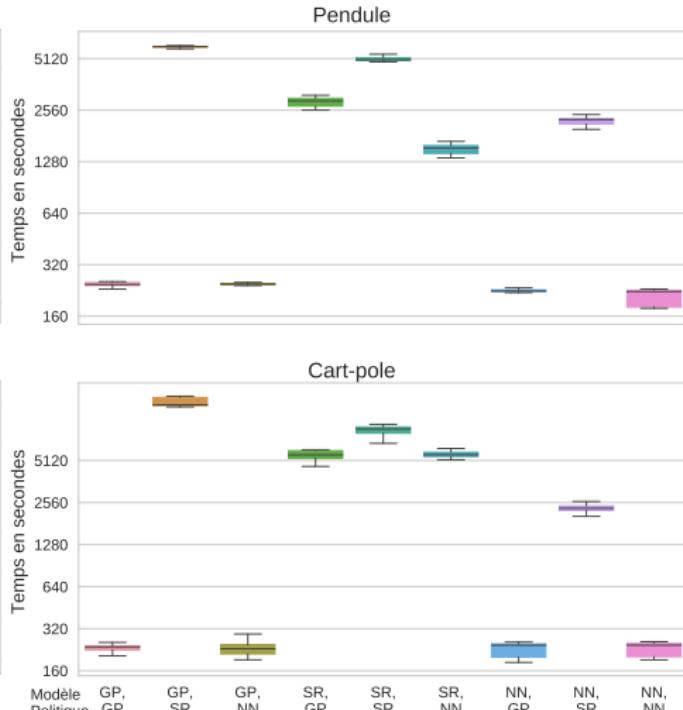


Cart-pole, 20 répliques

# Intégration dans Black-DROPS



Récompense, 20 répliques



Temps nécessaire, 20 répliques

## Formules trouvées

Variable	Pendule : $\Delta\dot{\theta}$
Formule trouvée	$\sin\left(\frac{u}{3}\right) - \frac{\cos(\theta)}{\cos(\sin(\cos(\frac{\sin(\frac{u}{\cos(7)}) - \sin(\frac{\sin(\cos(\theta))}{\cos(\sin(\theta))})}{3})))}$
Vraie formule	$\frac{-3ml\cdot\dot{\theta}^2\cdot\sin(\theta)\cdot\cos(\theta) - 6(M+m)g\cdot\sin(\theta) - 6(u - b\dot{x})\cdot\cos(\theta)}{4l(m+M) - 3ml\cdot\cos(\theta)^2}$

Variable	Pendule : $\Delta\theta$	Cartpole : $\Delta x$
Formule trouvée	$\Delta t \cdot \dot{\theta} + \frac{\Delta t(\dot{\theta} + \Delta t)}{8 \cdot 9 \cdot g^3 (\frac{\dot{\theta}}{3} - 6)}$	$\dot{x} \cdot \sin(\Delta t) - \sin(0)$
Vraie formule	$\dot{\theta} \cdot \Delta t$	$\dot{x} \cdot \Delta t$

# Conclusion

- ▶ Implémentation d'un version régression symbolique de Black-DROPS
- ▶ Utilisation des bibliothèques limbo<sup>13</sup> et Sferes2<sup>14</sup>
- ▶ Modèles seul : comparable ou meilleur que les réseaux de neurones
- ▶ Politiques : comparable ou meilleur que les réseaux de neurones

Mais

- ▶ Dans Black-DROPS : résultats significatifs mais inférieurs pour le pendule, résultats non significatifs pour le cart-pole.
- ▶ Temps de calcul très supérieur aux autres représentations
- ▶ Formules parfois peu interprétables

---

13. Cully et al. 2016. "Limbo : A Flexible High-performance Library for Gaussian Processes modeling and Data-Efficient Optimization". *Preprint*.

14. Mouret et Doncieux 2010. "SFERESv2 : Evolvin' in the Multi-Core World".

## Extensions possibles

- ▶ Simplifier les formules au cours de la recherche par des techniques de calcul formel<sup>15</sup>
- ▶ Faire une recherche locale pour les valeurs des constantes via une descente de gradient ou une technique d'optimisation<sup>16</sup>
- ▶ Affiner le processus de croisement et de construction des formules
- ▶ Introduire un mécanisme d'incertitude pour les prédictions

---

15. Worm et Chiu 2013. "Prioritized grammar enumeration : symbolic regression by dynamic programming".

16. Melo, Fowler et Banzhaf 2015. "Evaluating methods for constant optimization of symbolic regression benchmark problems".

# Contributions

- ▶ Implémentation d'un variante régression symbolique de Black-DROPS
- ▶ Utilisation des bibliothèques limbo<sup>17</sup> et Sferes2<sup>18</sup>
- ▶ Implémentation d'un module gérant les arbres pour Sferes2
- ▶ Comparaison de la variante avec d'autres représentations

---

17. Cully et al. 2016. "Limbo : A Flexible High-performance Library for Gaussian Processes modeling and Data-Efficient Optimization". *Preprint*.

18. Mouret et Doncieux 2010. "SFERESv2 : Evolvin' in the Multi-Core World".

## Annexe - CMA-ES

### Covariance MAtrix Evolution Strategy<sup>19</sup>

- ▶ Optimisateur de boite noire
- ▶ Ne fonctionne que dans des espaces vectoriels
- ▶ Cherche la valeur optimale d'une fonction en sélectionnant les meilleures valeurs parmi des échantillons tirés selon une gaussienne
- ▶ Les paramètres de cette gaussienne évoluent dynamiquement pour refléter la répartition des meilleurs échantillons sélectionnés

---

19. Hansen et Ostermeier 2001. "Completely derandomized self-adaptation in evolution strategies". *Evolutionary computation*.

# NSGA-II

- ▶ Chaque objectif est considéré à égale importance et sans compromis
- ▶ Sélection : rang puis diversité
- ▶ Tri non dominé : maintient de la diversité vis à vis des objectifs

---

## Algorithm 2 NSGA-II

---

```
Initialiser la population
while Tâche non résolue do
    Croisements et mutations
    Mise à jour de la fitness
    Tri non dominé (front de Pareto)
    Sélection élitiste
end while
```

---