Optimisation boîte noire discrète avec un algorithme d'apprentissage par renforcement invariant à l'ordre de génération

Olivier Goudet, Quentin Suire, Adrien Goëffon,
Frédéric Saubion et Sylvain Lamprier

LERIA, Université d'Angers
2 Boulevard Lavoisier, 49045 Angers, France

Mots-clés : Optimisation boîte noire, Algorithme à estimation de distribution, Apprentissage par renforcement.

Introduction

Dans ce travail, nous présentons un *framework* d'apprentissage par renforcement invariant à l'ordre de génération pour l'optimisation combinatoire de type boîte noire. Les algorithmes classiques d'estimation de distribution (EDA) [1], sous-classe des algorithmes évolutionnaires, résolvent de tels problèmes en apprenant et en échantillonnant un modèle probabiliste des solutions prometteuses. Ils s'appuient souvent sur l'apprentissage de graphes explicites de dépendance entre variables, ce qui peut s'avérer coûteux et ne pas permettre de saisir efficacement les interactions complexes entre les variables. Dans ce travail, nous paramétrons un modèle génératif autorégressif multivarié entraîné sans ordre fixe des variables. En échantillonnant des ordres de génération aléatoires pendant l'entraînement (une forme de dropout préservant les informations), le modèle est encouragé à être invariant par rapport à l'ordre des variables, ce qui favorise l'exploration de l'espace de recherche et façonne le modèle pour qu'il se concentre sur les dépendances variables les plus pertinentes, améliorant ainsi l'efficacité de l'échantillonnage de nouvelles solutions. Nous adaptons l'algorithme d'apprentissage par renforcement *Generalized Reinforcement Policy Optimization (GRPO)* [2] à ce contexte, ce qui permet de fournir des mises à jour stables du gradient des politiques à partir d'avantages invariants à toute transformation monotone de la fonction objectif. En comparaison avec un large éventail d'algorithmes de référence et sur un panel diversifié d'instances de problèmes de tailles variées, notre méthode atteint fréquemment les meilleures performances.

Algorithme EDA multivarié avec apprentissage par renforcement invariant à l'ordre de génération

Caractéristiques de l'approche (σ, σ')-RL-EDA :

- Le problème est vu comme un **MDP** où chaque état correspond à une solution partielle.
- La politique générative π_θ est paramétrée par des réseaux de neurones, un par variable.
- L'objectif est de maximiser la récompense globale (fitness) via des gradients de politiques, stabilisés par un algorithme de type GRPO.

Exemple jouet : La figure 1 présente un exemple de génération à l'instant t d'une population Γ_λ^t avec $\lambda = 2$ individus pour un problème de maximisation avec $N = 3$ variables. L'ordre de génération du premier individu est indiqué par des flèches bleues. Lorsque nous le construisons avec le MDP et l'ordre donné $\sigma_G^1 = (3, 1, 2)$, nous commençons par $x^1_{\sigma_G^1<1} = (0, 0, 0)$ donné en entrée au réseau de neurones g_{θ_3} qui génère $x_3 = 1$, puis $x^1_{\sigma_G^1<2} = (0, 0, 1)$ est donné en entrée à g_{θ_1} qui génère $x_1 = -1$, et enfin $x^1_{\sigma_G^1<3} = (-1, 0, 1)$ est donné en entrée à g_{θ_2} qui génère la valeur x_2 de la dernière variable et nous obtenons la solution complète $(-1, -1, 1)$. Lorsque tous les individus de la population sont échantillonnés, nous passons à la phase d'évaluation où les avantages sont calculés de telle sorte que $A_{\Gamma_\lambda^t}(x^i_{best}) = +1$ et $A_{\Gamma_\lambda^t}(x^i_{worse}) = -1$.

La phase d'entraînement s'appuie sur GRPO pendant E époques avec les $\lambda = 2$ solutions échantillonnées à cette génération. À chaque époque, de nouveaux ordres σ_G sont échantillonnés pour chaque individu. Les probabilités conditionnelles des actions sont ensuite calculées en fonction des masques causaux correspondants. Cela permet de mettre à jour les paramètres du modèle par descente de gradient.

Résultats expérimentaux

Nous avons considéré des problèmes discrets (optimisation quadratique (**QUBO**) et des instances basées sur le modèle classique **NK-landscape**). Notre approche (σ, σ')-RL-EDA est comparée avec la bibliothèque d'algo-

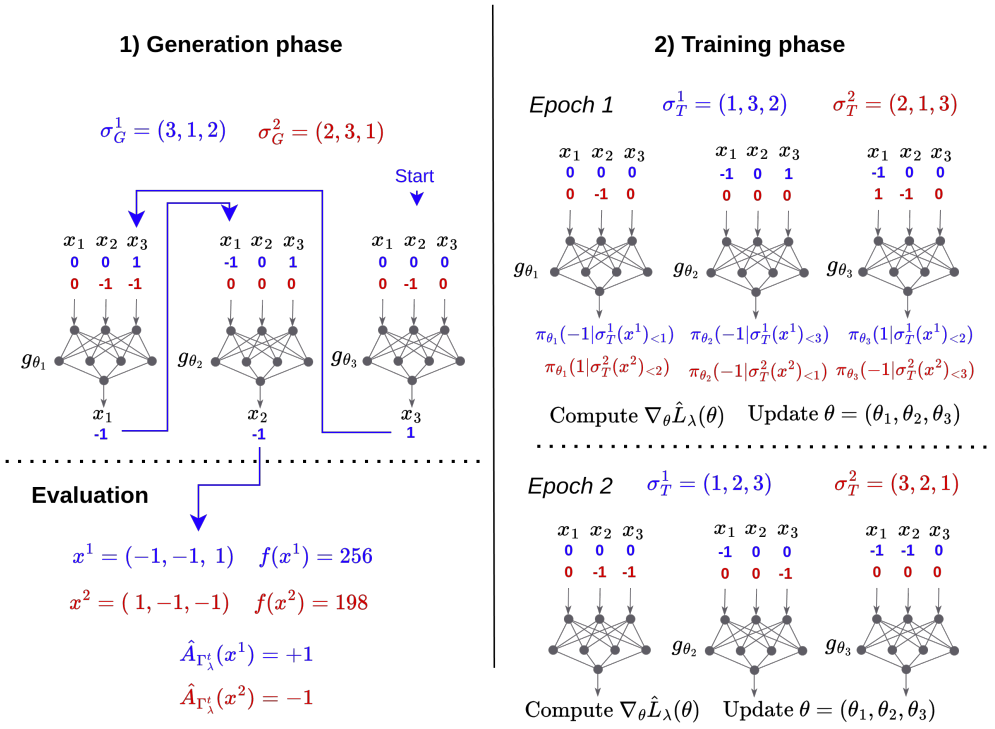
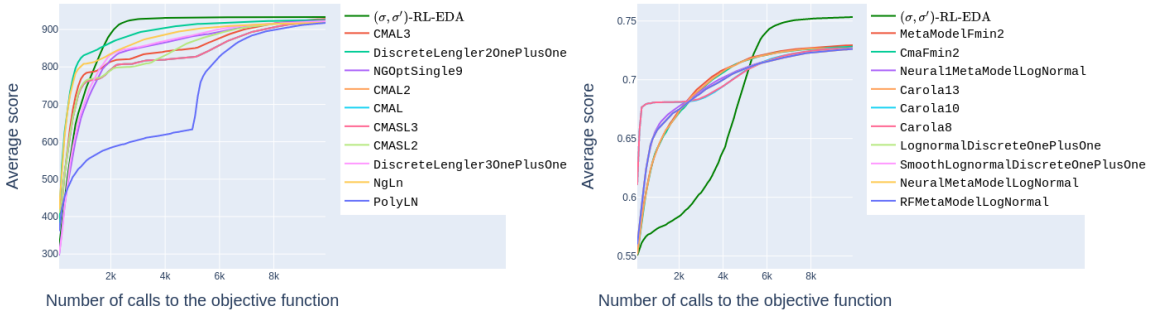


FIGURE 1 – Illustration du fonctionnement de (σ, σ')-RL-EDA

rithmes Nevergrad et des EDAs classiques (PBIL, MIMIC, BOA). Elle obtient de très bons résultats sur une grande variété d'instances. Elle permet de mieux explorer l'espace de recherche et de découvrir les interactions entre les variables grâce à la diversité induite par les ordres aléatoires.

Exemple de courbes d'évolution : À titre d'exemple, la figure 2 présente des graphiques illustrant l'évolution des meilleurs scores (moyenne sur 100 exécutions) en fonction du nombre d'évaluations de la fonction objective pour les instances QUBO de taille $N = 128$ et de type $K = 5$ et les instances NK de taille $N = 256$ et de type $K = 4$. Sur ce graphique, (σ, σ')-`RL-EDA` (courbe verte) est comparé aux 10 autres algorithmes concurrents les plus performants parmi un ensemble de 504 algorithmes.



(a) Instances QUBO avec $N = 128$ et $K = 5$.

(b) Instances NK avec $N = 256$ et $K = 4$.

FIGURE 2 – Axe X : nombre d'appels à la fonction objectif. Axe Y : évolution des scores moyens.

Références

[1] Martin Pelikan, Mark Hauschild, and Fernando G. Lobo. Estimation of distribution algorithms. In Janusz Kacprzyk and Witold Pedrycz, editors, *Springer Handbook of Computational Intelligence*, Springer Handbooks, pages 899–928. Springer, 2015.

[2] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath : Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv :2402.03300*, 2024.