Cahier des charges

Parallélisation de l'apprentissage par renforcement base sur des populations

Tuteur du projet :

Stéphane Doncieux

Johann Huber

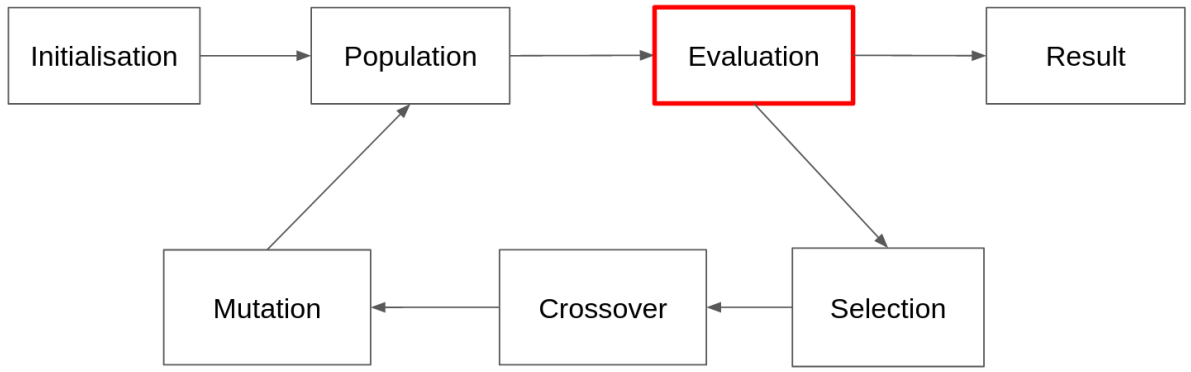
Membres de l'équipe :

Charles Lin

Zhe Wang

1 - Contexte et définition du projet

Le projet s'inscrit dans le domaine de l'apprentissage par renforcement, une technique d'apprentissage automatique où un agent interagit avec son environnement pour apprendre à prendre des décisions. Les algorithmes de recherche de nouveauté (NS) et de qualité-diversité (QD) sont des approches alternatives à l'optimisation basée sur la performance seule. Ce sont des algorithmes évolutionnaires qui demandent une importante capacité computationnelle, car leur leur apprentissage ne s'appui que sur l'expérience. Il est donc important d'utiliser des méthodes de parallélisation pour accélérer le temps d'exécution des expériences, dans notre projet on s'intéressera à l'optimisation de la phase d'évaluation qui sont les plus coûteuses pour nos deux algorithmes.



Différente étape d'un algorithme évolutionnaire

Actuellement la librairie est parallélisé sur CPU, et notre but serait de le réaliser sur GPU, qui ont été spécialement conçus pour le traitement parallèle massif de données.

JAX est une librairie d'apprentissage automatique développée par Google qui permet de paralléliser massivement des calculs sur GPU. BRAX est une librairie développée en JAX pour simuler des environnements standards en apprentissage par renforcement, notamment pour la robotique. La parallélisation mise en œuvre

dans les environnements BRAX permet de réduire significativement les temps d'exécution des algorithmes d'apprentissage.

2 - Objectif du projet

Le but principal de ce projet est d'intégrer les environnements de simulation BRAX à la librairie diversity_algorithms pour accélérer les temps de calcul et donc d'améliorer l'efficacité des algorithmes d'apprentissage par renforcement basés sur la recherche de nouveauté et de qualité-diversité. Les objectifs spécifiques incluent :

1. Modifier la librairie diversity_algorithm pour permettre l'utilisation d'environnements BRAX.
2. Tester les environnements BRAX, comparer les performances de diversity_algorithm avec BRAX et leurs équivalents dans les simulateurs précédemment utilisés, comme MUJOCO.
3. Si possible, implémenter des environnements jouets supplémentaires en utilisant JAX.

Le résultat attendu du projet est une version de la librairie diversity_algorithms qui intègre des environnements BRAX. L'analyse des choix de conception, les visualisations interprétables des résultats et les études comparatives en termes de temps et de coût computationnel, de complexité en usage mémoire et en calculs devront être présentées pour justifier les choix effectués pendant le projet.

3 - Périmètre du projet

Dans le cadre de ce projet, le périmètre comprend les tâches suivantes :

- Intégrer les environnements de simulation BRAX à la librairie diversity_algorithm pour améliorer les performances des algorithmes d'apprentissage par renforcement basés sur la recherche de nouveauté et de qualité-diversité.
- Effectuer des tests comparatifs des performances des environnements BRAX avec celles des simulateurs précédemment utilisés, tels que MUJOCO.
- Si possible, implémenter de nouveaux environnements jouets en utilisant JAX.

Les tâches qui ne sont pas incluses dans le périmètre du projet sont :

- Développer de nouveaux algorithmes d'apprentissage par renforcement basés sur la recherche de nouveauté et de qualité-diversité.
- Modifier BRAX ou la librairie diversity_algorithm de manière à changer leur fonctionnement fondamental.
- Effectuer des tests de performance sur des ordinateurs autres que ceux qui sont mis à disposition dans le cadre du projet.
- Réaliser une analyse détaillée de la consommation énergétique et des coûts liés à l'utilisation de l'algorithme.

4 - Besoins et contraintes

Besoins:

- Connaissance de base des algorithmes NS et QD.
- Connaissance en programmation en Python.
- Connaissance de la bibliothèque BRAX et de la parallélisation GPU.
- Connaissance de la librairie diversity_algorithms

Contraintes:

1. La structure de la librairie diversity_algorithms doit être préservée autant que possible pour garantir son intégrité.
2. L'intégration de BRAX doit être réalisée de manière cohérente
3. Les résultats doivent être facilement interprétables à travers des visualisations et une analyse des choix de conception