

REPUBLIQUE DEMOCRATIQUE DU CONGO
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET UNIVERSITAIRE
UNIVERSITE REVEREND KIM
SITE DE LINGWALA
FACULTE DE SCIENCES INFORMATIQUES

L3 LMD INFO/GENIE LOGICIEL
BP 171.KIN XX



TRAVAIL PRATIQUE DE L'IA

Fait par l'étudiant :

NGASA MOZIGA FLAVIEN

Dirigé par :

DOCTORANT DONATIEN KADIMA

ANNÉE ACADÉMIQUE : 2025– 2026

TRAVAIL PRATIQUE

Question

Parlez brièvement de l'algorithme évolutionnaire en IA : historique, Auteur, Avantage, Inconvénient et de l'Impact sur l'IA.

LES ALGORITHMES ÉVOLUTIONNAIRES

INTRODUCTION FONDAMENTALE

1. Définition et Principes de Base

Les algorithmes évolutionnaires (AE) constituent une famille de méthodes d'optimisation et de recherche inspirées par les mécanismes de l'évolution naturelle. Ils appartiennent au domaine plus large de l'intelligence computationnelle et s'appuient sur des principes darwiniens fondamentaux :

- **Sélection naturelle** : Les individus les plus adaptés ont une probabilité plus élevée de survivre et de se reproduire ;
- **Hérédité** : Les caractéristiques des parents sont transmises à leur descendance ;
- **Variation génétique** : Introduction de nouveautés grâce aux opérateurs de mutation et de recombinaison ;
- **Adaptation progressive** : Amélioration cumulative des populations sur plusieurs générations.

2. Paradigme Évolutionnaire en Informatique

Le paradigme évolutionnaire représente un changement fondamental dans l'approche de résolution de problèmes. Contrairement aux méthodes algorithmiques traditionnelles qui suivent une séquence d'instructions prédéterminée, les AE adoptent une approche stochastique et populationnelle où :

- La recherche est menée par une population de solutions candidates ;
- L'espace de recherche est exploré de manière parallèle ;
- L'information est échangée entre solutions via des opérateurs biologiques ;
- La direction de recherche émerge de l'interaction entre opérateurs et environnement.

CONTEXTE HISTORIQUE ET DEVELOPPEMENT

1. Précurseurs et Origines (Années 1950-1960)

Les premières tentatives d'appliquer les principes évolutionnaires à la résolution de problèmes remontent aux travaux de :

- **Nils Aall Barricelli** (1954) : Premières simulations numériques de processus évolutifs ;
- **Lawrence J. Fogel** (1960) : Développement de la programmation évolutive pour créer des machines à états finis ;
- **Hans-Joachim Bremermann** (1958) : Applications précoces en optimisation continue.

2. Fondations Théoriques (Années 1970)

2.1. John Holland et les Algorithmes Génétiques

John Holland de l'Université du Michigan établit les bases théoriques solides avec :

- **Schéma fondamental** : Publication de "Adaptation in Natural and Artificial Systems" (1975) ;
- **Théorème des schémas** : Fondation mathématique expliquant pourquoi les algorithmes génétiques fonctionnent ;
- **Opérateurs canoniques** : Définition des opérateurs de sélection, croisement et mutation ;

2.2. École Allemande : Stratégies d'Évolution

- **Ingo Rechenberg** et **Hans-Paul Schwefel** (Université de Berlin) ;
- Développement des **stratégies d'évolution** (Evolution Strategies) ;
- Application initiale à des problèmes d'ingénierie hydraulique ;
- Emphasis sur l'auto-adaptation des paramètres.

2.3. Consolidation et Diversification (Années 1980-1990)

- **Émergence de nouvelles variantes** : Programmation génétique, algorithmes évolutionnaires différentiels ;
- **Conférences spécialisées** : ICGA, PPSN, CEC ;
- **Applications industrielles** : Optimisation aérodynamique, planification, conception.

FONDEMENTS THEORIQUES ET MATHEMATIQUES

1. Cadre Formel des AE

Un algorithme évolutionnaire peut être défini formellement comme un 8-uplet :

$$AE = (P_0, \delta, l, \Phi, s, c, m, t)$$

Où :

- AE : Algorithme Evolutionnaire
- P_0 : Population initiale
- δ : Fonction d'évaluation (fitness)
- l : Représentation des individus
- Φ : Opérateur de sélection
- s : Opérateur de survie
- c : Opérateur de recombinaison
- m : Opérateur de mutation
- t : Critère de terminaison

2. Théorème des Schémas (Holland, 1975)

Le théorème des schémas fournit une explication théorique de la puissance des AE :

- **Définition d'un schéma** : Module ou patron décrivant un sous-ensemble de solutions partageant certaines similarités.
- **Équation fondamentale** :

$$E[m(H, t+1)] \geq m(H, t) \times (f(H)/f_{\text{moy}}) \times [1 - p_c \times (\delta(H)/(l-1)) - p_m \times o(H)]$$

Où :

- $m(H, t)$: Nombre d'instances du schéma H à la génération t
- $f(H)$: Fitness moyenne du schéma H
- f_{moy} : Fitness moyenne de la population
- p_c, p_m : Probabilités de croisement et mutation
- $\delta(H)$: Longueur définissante du schéma
- $o(H)$: Ordre du schéma

3. Paysages de Fitness et Difficulté

La performance des AE est intimement liée à la structure du paysage de fitness :

- **Épistasis** : Degré d'interaction entre les gènes
- **Rugosité** : Variabilité locale du fitness
- **Modalité** : Nombre d'optima locaux

- **Déceptivité** : Tendance à guider la recherche vers des optima locaux

🌈 ARCHITECTURES ET VARIANTES PRINCIPALES

1. Algorithmes Génétiques (AG)

1.1. Caractéristiques Distinctives

- **Représentation** : Généralement binaire, mais d'autres représentations possibles
- **Opérateurs** : Sélection proportionnelle, croisement à un ou deux points, mutation bit à bit
- **Emphase** : Maintien de la diversité génétique, exploration large

1.2. Processus Typique

```
Initialisation aléatoire de la population
  Évaluation de chaque individu
    Répéter jusqu'à convergence :
      Sélection des parents
      Application des opérateurs génétiques
      Évaluation des descendants
      Remplacement de la population
```

2. Stratégies d'Évolution (ES)

2.1. Caractéristiques Distinctives

- **Représentation** : Vecteurs de nombres réels
- **Paramètres stratégiques** : Pas de mutation auto-adaptatif
- **Schémas de remplacement** : (μ, λ) ou $(\mu + \lambda)$
- **Emphase** : Optimisation continue, exploitation locale

2.2. Mutation Gaussienne Auto-adaptative

```
 $\sigma_{i'} = \sigma_i \times \exp(\tau \times N(0,1) + \tau' \times N_i(0,1))$ 
 $x_{i'} = x_i + \sigma_{i'} \times N_i(0,1)$ 
```

Où σ représente le pas de mutation, adapté dynamiquement.

3. Programmation Évolutive (PE)

- **Origine** : Fogel, Owens et Walsh (1966)
- **Représentation** : Machines à états finis, programmes

- **Opérateurs** : Mutation comme opérateur principal
- **Application** : Optimisation continue, prédiction de séries temporelles

4. Programmation Génétique (PG)

- **Pionnier** : John Koza (1992)
- **Représentation** : Arbres syntaxiques
- **Application** : Découverte de programmes, modèles mathématiques
- **Opérateurs spéciaux** : Croisement de sous-arbres, mutation de nœuds

5. Algorithmes Évolutionnaires Différentiels (DE)

- **Inventeur** : Kenneth Price et Rainer Storn (1995)
- **Principe** : Utilisation de différences vectorielles pour la mutation
- **Avantage** : Simplicité, efficacité sur problèmes continus
- **Opérateur de mutation** :

$$v = x_{r1} + F \times (x_{r2} - x_{r3})$$

COMPOSANTS DÉTAILLÉES DES AE

1. Mécanismes de Représentation

1.1.Représentations Binaires

- **Avantages** : Simplicité théorique, couverture uniforme de l'espace
- **Inconvénients** : Problème de Hamming, précision limitée
- **Codage réel/binaire** : nécessité de définir une précision et un domaine

1.2.Représentations Réelles

- **Avantages** : Précision naturelle, opérateurs spécifiques au domaine
- **Inconvénients** : Espace de recherche potentiellement infini

1.3.Représentations Symboliques et Structurelles

- **Arbres** : Pour la programmation génétique
- **Graphes** : Pour les réseaux neuronaux
- **Permutations** : Pour les problèmes d'ordonnancement

2. Opérateurs de Sélection

2.1. Sélection Proportionnelle (Roulette)

$$p_i = f_i / \sum f_j$$

- **Avantage** : Pression de sélection naturelle
- **Problème** : Convergence prématurée pour les individus très performants

2.2.Sélection par Tournoi

- **Mécanisme** : Sélection de k individus, choix du meilleur
- **Avantage** : Contrôle simple de la pression de sélection
- **Paramètre** : Taille du tournoi k

2.3.Sélection par Rang

- **Principe** : Sélection basée sur le rang plutôt que la valeur absolue de fitness
- **Avantage** : Évite la domination des super-individus

3. Opérateurs de Recombination

3.1. Croisement à Un Point

```
Parent1: [AAAAA|AAAAA]  Parent2: [BBBBB|BBBBB]
Enfant1: [AAAAA|BBBBB]  Enfant2: [BBBBB|AAAAA]
```

3.2.Croisement à Deux Points

```
Parent1: [AAA|AAA|AA]  Parent2: [BBB|BBB|BB]
Enfant1: [AAA|BBB|AA]  Enfant2: [BBB|AAA|BB]
```

3.3.Croisement Uniforme

- **Principe** : Chaque gène est choisi aléatoirement d'un parent ou de l'autre
- **Masque de croisement** : Séquence binaire déterminant le parent donneur

3.4.Recombinaisons Spécialisées

- **Arithmétique** : Combinaison convexe pour les représentations réelles
- **BLX- α** : Extension de l'intervalle parental
- **SPX** : Simplex crossover pour la préservation de la diversité

4. Opérateurs de Mutation

1. Mutation Bit-Flip

- Pour les représentations binaires
- Inversion bit à bit avec probabilité p_m

2. Mutation Gaussienne

$$x' = x + N(0, \sigma)$$

- Adaptation possible du pas σ

3.2. Mutation Polynomiale

- Génération de nombres aléatoires suivant une distribution polynomiale
- Maintien de la solution dans les bornes

5. Stratégies de Remplacement

1. Remplacement Générationnel

- **(μ, λ)** : Les parents sont entièrement remplacés par les descendants
- **Élitisme** : Conservation du meilleur individu

2. Remplacement Steady-State

- Remplacement progressif de quelques individus à chaque génération
- Stratégies basées sur l'âge ou le fitness

6. Analyse des Avantages et Limitations

1. Avantages Détaillés

1.1. Robustesse et Universalité

- **Indépendance du domaine** : Peuvent s'appliquer à presque tout type de problème
- **Pas d'hypothèses fortes** : Continuité, dérivabilité non requises
- **Tolérance au bruit** : Fonctionnent même avec des fonctions d'évaluation bruitées

1.2. Capacités d'Exploration

- **Recherche parallèle** : Exploration simultanée de multiples régions

- **Évasion des optima locaux** : Maintien de la diversité populationnelle
- **Adaptation dynamique** : Capacité à suivre des optimums changeants

1.3.Flexibilité et Extensibilité

- **Hybridation facile** : Combinaison avec d'autres méthodes
- **Intégration de connaissances** : Contraintes et heuristiques intégrables
- **Optimisation multi-objectifs** : Extension naturelle aux problèmes Pareto

2. Limitations et Défis

2.1.Complexité Computationnelle

- **Nombre d'évaluations** : Peut être prohibitif pour les problèmes coûteux
- **Paramétrage** : Nombreux paramètres à régler
- **Convergence lente** : Comparé aux méthodes déterministes pour problèmes convexes

2.2.Problèmes Théoriques

- **Conditions de convergence** : Pas toujours garanties
- **Analyse de complexité** : Difficile pour des problèmes généraux
- **Certification des solutions** : Qualité difficile à prouver

2.3.Défis Algorithmiques

- **Détermination des paramètres** : Règles de réglage souvent empiriques
- **Échelle des problèmes** : Difficulté avec les problèmes de très grande dimension
- **Déceptivité** : Échec sur les problèmes spécialement conçus

7. Applications et Domaines d'Impact

1. Ingénierie et Conception

1.1. Conception Aérodynamique

- **Optimisation de profils** : Ailes, pales de turbines
- **Exemple notable** : Optimisation de la tuyère du moteur de la fusée Ariane
- **Gains** : Améliorations de 10-15% en efficacité

1.2. Conception Électronique

- **Placement et routage** : Circuits intégrés

- **Syntonisation de filtres** : Optimisation de caractéristiques fréquentielles
- **Antennes** : Conception de formes complexes

2. Planification et Ordonnancement

2.1.Problèmes Industriels

- **Flow-shop** : Optimisation de la séquence de production
- **Problème de voyageur de commerce** : Logistique et transport
- **Affectation de ressources** : Allocation optimale

2.2. Applications Réelles

- **Airlines** : Planification des équipages
- **Manufacturing** : Ordonnancement de production
- **Transport** : Optimisation de flottes

3. Apprentissage Automatique et IA

3.1.Optimisation d'Architectures

- **Réseaux de neurones** : Topologie, fonctions d'activation, hyperparamètres
- **Systèmes flous** : Optimisation des bases de règles
- **Sélection de caractéristiques** : Réduction de dimension

3.2.Apprentissage par Renforcement Évolutionnaire

- **Combinaison AE/AR** : Politiques d'action apprises par évolution
- **Avantage** : Moins sensible au crédit assignment problem
- **Applications** : Robotique, jeux vidéo

IMPACT SUR LE DEVELOPPEMENT DE L'IA

1. Contribution aux Fondements de l'IA

1.1. Paradigme Alternatif

Les AE ont fourni une alternative aux approches symboliques et connexionnistes :

- **Recherche populationnelle** vs recherche locale
- **Émergence** vs conception explicite
- **Robustesse** vs optimalité garantie

1.2. Inspiration Biologique

- **Validation du computationnalisme** : Preuve que les processus naturels peuvent être calculés
- **Ouverture vers d'autres métaphores** : Essaims, immunité, culture

2. Synergies avec l'Apprentissage Profond

2.1. Neuro-évolution

- **NEAT** (NeuroEvolution of Augmenting Topologies) : Évolution conjointe des poids et topologies
- **CoDeepNEAT** : Extension pour les architectures profondes
- **Evolution stratégique** : Pour l'entraînement des réseaux profonds

2.2. Optimisation d'Hyperparamètres

- **Recherche d'architecture** : Découverte automatique de modèles performants
- **Réglage d'optimiseurs** : Paramètres de learning rate, batch size
- **NAS (Neural Architecture Search)** : Utilisation intensive des AE

3. Impact sur les Méthodologies de Recherche

3.1. Approche Expérimentale

- **Culture du benchmark** : Développement de problèmes tests standards
- **Analyse statistique** : Importance des tests rigoureux
- **Reproductibilité** : Standards de comparaison

3.2. Interdisciplinarité

- **Biologie computationnelle** : Échanges fructueux avec la biologie théorique
- **Économie évolutionnaire** : Applications aux sciences sociales
- **Robotique évolutive** : Conception de contrôleurs adaptatifs

4. Tendances Actuelles et Recherche Avancée

1. AE à Grande Échelle et Parallélisme

1.1. Modèles Massivement Parallèles

- **Algorithmes cellulaires** : Populations structurées spatialement
- **Îlots** : Sous-populations avec migration
- **GPU computing** : Évaluation parallèle des fitness

1.2.Défis des Hautes Dimensions

- **Malédiction de la dimensionnalité** : Dégradation des performances en grande dimension
- **Techniques de réduction** : Embeddings, projections
- **Décomposition cooperative** : Coevolution, CCGA

5. AE Hybrides et Méta-optimisation

5.1.Mémoïques

- **Combinaison AE/recherche locale** : Exploitation/exploration équilibrée
- **Auto-adaptation** : Apprentissage des paramètres pendant l'exécution
- **Transfert de connaissances** : Utilisation d'expériences passées

5.2.Hyper-heuristiques

- **Sélection d'opérateurs** : Choix adaptatif des opérateurs
- **Paramétrage automatique** : F-Race, iRace
- **Configuration algorithmique** : SMAC, ParamILS

Références Clés :

- ✚ Holland, J.H. (1975). Adaptation in Natural and Artificial Systems
- ✚ Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning
- ✚ Eiben, A.E. & Smith, J.E. (2015). Introduction to Evolutionary Computing
- ✚ Deb, K. (2001). Multi-Objective Optimization using Evolutionary Algorithms
- ✚ Fogel, D.B. (2006). Evolutionary Computation: Toward a New Philosophy of Machine Intelligence