

REPUBLIQUE DEMOCRATIQUE DU CONGO
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET UNIVERSITAIRE
UNIVERSITE REVEREND KIM
FACULTE D'INFORMATIQUE



B .P 171 KIN XX
KINSHASA/LINGWALA

INTELLIGENCE ARTIFICIELLE

Fait par l'étudiant : IRUNG TSHIKWAT DIDA

Prof. Dr. Ir. : Elisée ILUNGA MBUYAMBA
DOCTORANT Donatien KADIMA MUAMBA

ANNEE ACADEMIQUE : 2025 – 2026

PARLEZ BRIÈVEMENT DES L'ALGORITHME ÉVOLUTIONNAIRE EN I.A

Les **algorithmes évolutionnistes** ou **algorithmes évolutionnaires** (*evolutionary algorithms* en anglais), sont une famille d'algorithmes dont le principe s'inspire de la théorie de l'évolution pour résoudre des problèmes divers. Ce sont donc des méthodes de calcul bioinspirées. L'idée est de faire évoluer un ensemble de solutions à un problème donné, dans l'optique de trouver les meilleurs résultats. Ce sont des algorithmes dits stochastiques, car ils utilisent itérativement des processus aléatoires.

Les algorithmes évolutionnaires s'inspirent de l'évolution des êtres vivants, en considérant que celle-ci tend à produire des organismes plus *adaptés* à leur environnement.

Selon la théorie de l'évolution, plusieurs mécanismes sont à l'œuvre pour ce faire. Schématiquement :

Les caractéristiques d'un organisme sont en grande partie codées dans ses gènes, chaque population d'organismes est composée d'individus tous différents, les individus sont plus ou moins adaptés à leur environnement, les organismes transmettent une partie de leurs caractéristiques à leurs descendants, les individus les plus adaptés se reproduisent plus « efficacement », leurs caractéristiques ont donc tendance à davantage se répandre dans la population.

I. HISTORIQUE

Historiquement, trois grandes familles d'algorithmes ont été développées indépendamment, entre le milieu des années 1960 et 1970. Les premières méthodes furent les stratégies d'évolution[2], proposées par I. Rechenberg en 1965, pour résoudre des problèmes d'optimisations continus. L'année suivante, Fogel, Owens et Walsh conçoivent la programmation évolutionnaire[3] comme une méthode d'intelligence artificielle pour la conception d'automates à états finis. Enfin, en 1975, J. H. Holland propose les premiers algorithmes génétiques[4], pour l'optimisation combinatoire. La parution en 1989 du livre de D. E. Goldberg sur les algorithmes génétiques rendra ceux-ci particulièrement populaires.

1. 1952 : premiers travaux sur l'utilisation de méthodes stochastiques pour l'optimisation^[6].
2. 1954 : Barricelli effectue les premières simulations du processus d'évolution et les utilise sur des problèmes d'optimisation généraux^[7].
3. 1965 : Rechenberg conçoit le premier algorithme utilisant des *stratégies d'évolution*^[2].
4. 1966 : Fogel, Owens et Walsh proposent la *programmation évolutionnaire*^[3].
5. 1970 : John Horton Conway conçoit le jeu de la vie, l'automate cellulaire le plus connu à ce jour.

6. 1975 : travaillant sur les automates cellulaires, Holland propose les premiers *algorithmes génétiques*^[8].
7. 1980 : Smith utilise la *programmation génétique*^[9].
8. 1986 : Farmer, Packard et Perelson travaillent sur les systèmes immunitaire artificiels.
9. 1988 : la première conférence sur les algorithmes génétiques est organisée à l'université de l'Illinois à Urbana-Champaign.
10. 1988 : Koza dépose son premier brevet sur la programmation génétique^[10].
11. 1989 : Goldberg publie un des livres les plus connus sur les algorithmes génétiques^[9].
12. 1989 : *Evolver*, le premier logiciel d'optimisation par algorithmes génétiques est publié par la société *Axcelis*.
13. 1989 : le terme *algorithme mémétique* apparaît^[11].
14. 1993 : le terme « *Evolutionary Computation* » (« calcul évolutionnaire » en français) se répand, avec la parution de la revue éponyme, publiée par le Massachusetts Institute of Technology.
15. 1996 : Mühlenbein et Paaß proposent les *algorithmes à estimation de distribution*^[12].
16. 1997 : Storn et Price proposent un algorithme à évolution différentielle^[13].
17. 2000 : premiers algorithmes génétiques interactifs^[14].

II. AUTEUR

Les algorithmes évolutionnaires regroupent plusieurs familles (algorithms genitures, strategies devolution, programming gentium...). Ils sont le fruit de travaux de plasters churches qui, chacun, ont posé des foundations essentials pour ce Domaine

- a) John Holland – Le père des Algorithmes Genesius (GA)
- John Henry Holland (1929–2015) était chercheur à l'university du Michigan.
C'est lui qui a officiellement posé les bases mathématiques et conceptuelles des algorithmes génétiques
Ses contributions majeures :

Formalisation des Algorithmes Génétiques (GA) dans les années 1960–1970.

Introduction des concepts de :

Population de solutions

Chromosomes et gènes artificiels

Opérations de croisement et mutation

Sélection basée sur une fonction de fitness

- b) Ingo Rechenberg – Pionnier des Stratégies d' Évolution (ES)
- Ingo Rechenberg (né en 1934) est un chercheur allemand, connu comme le **fondateur des Evolution Strategies (ES)**.
- A mené des expériences dans les années 1960 pour optimiser l'aérodynamisme des structures.

- · A développé la méthode (1+1) ES, une forme primitive mais très efficace d'optimisation.
- · A introduit le principe de self-adaptation (la mutation s'adapte toute seule).

c) Hans-Paul Schwefel - Cofondateur des ES et théoricien majeur
 Collaborateur de Rechenberg, Schwefel a joué un rôle déterminant dans l'évolution des ES.

Apports principaux :

A développé les (μ,λ) -ES et les $(\mu+\lambda)$ -ES, plus avancés et efficaces.

A optimisé l'utilisation de la mutation gaussienne.

A fourni une base théorique solide pour les stratégies d'évolution moder

d) John Koza - Le père de la Programmation Génétique (GP)

John R. Koza (né en 1944) est un chercheur américain connu pour avoir inventé la programmation génétique (Genetic Programming — GP).

Ses contributions :

A appliqué les principes des algorithmes génétiques pour faire évoluer des programmes entiers, pas seulement des vecteurs de paramètres.

A proposé l'utilisation des arbres syntaxiques comme représentation (très innovant).

III. AVANTAGE

Avantages des Algorithmes Évolutionnaires en Intelligence Artificielle

1. Résolution de problèmes complexes

Les algorithmes évolutionnaires peuvent gérer des problèmes non linéaires, difficiles à modéliser et sans solution analytique claire.

2. Absence de connaissance préalable

Ils ne nécessitent pas de dérivées, de gradient ou d'hypothèses mathématiques strictes.

Ils fonctionnent même dans des environnements mal définis.

3. Recherche parallèle de solutions

Grâce à l'utilisation d'une population, ils explorent plusieurs pistes en même temps, ce qui évite les solutions locales et améliore la qualité globale.

4. Flexibilité et adaptabilité

Ces algorithmes s'adaptent à divers contextes : optimisation, robotique, classification, conception automatique, etc.

5. Solutions innovantes

Les mutations et croisements donnent parfois des solutions originales et impossibles à imaginer manuellement.

IV. INCONVÉNIENT

Puisque le Solveur Évolutionnaire peut, en principe, trouver des solutions aux problèmes avec des relations non lisses, non linéaires, et même linéaires, pourquoi ne pas l'utiliser pour tous les problèmes ? En pratique, tout algorithme génétique ou évolutif présente certains inconvénients, qui vont de pair avec ses avantages : Un algorithme évolutif est beaucoup plus lent que des alternatives telles que les méthodes

GRG et Simplex — souvent par des facteurs de cent fois ou plus.

À mesure que la taille du problème augmente (de, disons, dix à cent ou mille variables de décision), un algorithme évolutif est souvent submergé par la dimensionnalité du problème et est incapable de trouver une solution qui s'approche de l'optimalité. Mais il est toujours possible de résoudre de tels problèmes importants avec le GRG ou le Solver de la méthode Simplexe.

Dans un algorithme évolutif, une solution n'est « bonne » qu'en comparaison avec d'autres solutions découvertes précédemment. Un algorithme évolutif n'a en réalité aucun concept de « solution optimale », ni aucun moyen de tester si une solution donnée est optimale (même localement optimale).

V. IMPACT SUR L'I.A

- A. Resolution de problèmes complexes
- B. Les algorithmes évolutionnaires peuvent gérer des problèmes non linéaires, difficiles à modéliser et sans solution analytique claire.
- C. Absence de connaissance préalable
- D. Ils ne nécessitent pas de dérivées, de gradient ou d'hypothèses mathématiques strictes. Ils fonctionnent même dans des environnements mal définis.
- E. Recherche parallèle de solutions
- F. Grâce à l'utilisation d'une population, ils explorent plusieurs pistes en même temps, ce qui évite les solutions locales et améliore la qualité globale.
- G. Flexibilité et adaptabilité
- H. Ces algorithmes s'adaptent à divers contextes : optimisation, robotique, classification, conception automatique, etc.
- I. Solutions innovantes
- J. Les mutations et croisements donnent parfois des solutions originales et impossibles à imaginer manuellement.
- K. Robustesse face aux données imparfaites
- L. Ils supportent bien le bruit, les erreurs et les environnements changeants, ce qui les rend fiables pour les systèmes réels.
- M. Parallélisation facile
- N. Leur structure permet une exécution distribuée ou multi-cœur, améliorant la rapidité et l'efficacité.

TROUVEZ LA SOLUTION OPTIMALE POUR QUITTER DE ARAT

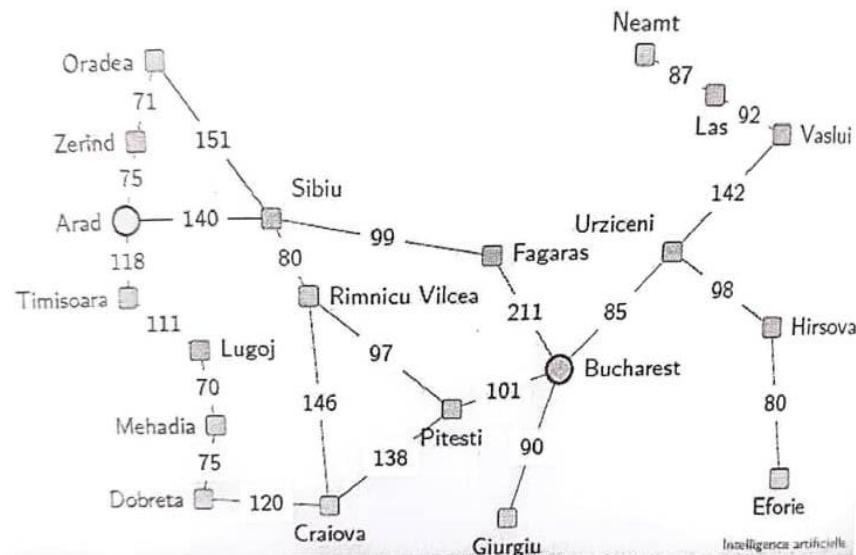


Figure 4.6. Une carte de route simplifié de la Roumanie.

VERS BUCHAREST

La résolution optimal pour quitter de ARAT vers BUCHAREST

J'ai trouvé 7 chemin à prendre pour quitter de notre point initial vers notre point but

Le solution (chemin) il l'aura de ARAT vers BUCHAREST pour une solution plus optimal