Algorithmes génétiques pour décrypter des mots de passes

Université Claude Bernard Lyon 1

Claire Pillet - 11422645, Cyril Reymond - 11300239
Groupe 4

***Abstract*—Les algorithmes génétiques visent à trouver une solution approchée à un problème d'optimisation et ce pour le résoudre dans un temps limité. Ces algorithmes utilisent des principes tirés de la nature qui sont la sélection naturelle de populations et des générations suivantes. Nous utilisons ces algorithmes pour retrouver un mot de passe que nous avions perdu.**

***Index Terms*—Algorithmes Génétiques, Intelligence Artificielle, Algorithmes Evolutionnistes.**

I. Introduction

Dans ce papier, nous étudions les différents aspects des algorithmes génétiques pour optimiser au maximum la découverte d'un mot de passe. Le fonctionnement repose sur l'évolution d'une population d'individus, qui, au cours des générations voient leurs génômes s'améliorer. Dans le problème qui nous intéresse ici, le phénotype de chaque individu représente un mot de passe. Ainsi, chaque individu correspond à un essai pour découvrir le bon mot de passe.

II. Principes généraux

Les algorithmes génétiques se décomposent en trois étapes clés : la sélection, le cross-over et la mutation.

A. Sélection

Ce processus est similaire à une sélection naturelle d'individus. Selon l'environnement, les individus développent des caractéristiques propres leur permettant une meilleur adaptabilité et survie. Les individus présentant les meilleurs attributs génômiques auront plus de chances de faire parti des reproducteurs pour engendrer la génération suivante.

B. Cross-over

Cette étape, appelée enjambement ou croisement, consiste en l'association de différentes façons de deux chromosomes pour créer de nouveaux chromosomes. Dans les algorithmes génétiques, cette méthode sert de reproduction entre individus pour mettre ne place une nouvelle génération. La probabilité d'apparation d'un cross-over est un paramètre à faire varier et dépend du problème rencontré.

C. Mutation

Dans la nature, des mutations d'un gène surviennent de manière aléatoire. Ces mutations apportent une diversité génétique suplémentaire pouvant, dans certains cas, améliorer grandement les performances d'un individus. De plus, de par son caractère aléatoire, ce paramètre permet d'accentuer la diversité au sein d'une population.

III. Découverte de mots de passes

L'objectif de notre papier est de découvrir un mot de passe perdu composé de douze à dix-huits caractères, des nombres et des lettres majuscules. Le génome d'un individu est au centre de l'algorithme.

A. Codage du génome

Nous prenons en compte deux aspects des génomes des êtres vivants : le phénotype et le génotype.

1) Phénotype: Les gènes codent les caractéristiques apparentes d'un individus tel que la couleur des yeux. Dans notre étude, les individus ont un seul objectif qui est la découverte de mot de passe. Ainsi, leur phénotype est simplement un mot de passe.

2) Génotype: Le génotype est l'ensemble des gènes et la manière dont ils sont codés. Nous avons choisi d'exprimer les gènes sous forme de code ASCII puisque deux caractères qui se suivent, par exemple les lettres A et B, ont leur code consécutif. Ainis, dès lors qu'une mutation se produit, il est possible d'obtenir un gène proche de l'original.

B. Fonction de fitness

Pour connaître la valeur d'un mot de passe, il est comparé au mot de passe à découvrir. Un score, la fitness, est attribué à l'individu en fonction de la proximité entre les deux mots de passes.

Figure 1. Mot de passe à découvrir par les individus de la population

La table 1 fourni quelques exemples de mots de passes et de la fitness associée. Il est intéressant de voir que pour une même longueur, la fitness varie grandement. Dans le dernier

Table I
COMPARAISON DE MOTS DE PASSES ET LEURS FITNESS

Mot de passe	Longueur	Fitness
0L897BLSZE1HDRD0O	17	0.88323
0L877BLSZEIRD1D03	17	0.93289
U0L8BLSZEIRD1DDO0	17	0.96078

cas, deux mutations sont nécessaires pour obtenir le bon mot de passe. Ces exemples soulignent l'importance d'un choix judicieux des hyper-paramètres.

IV. ALGORITHME NAÏF

La première réalisation a consisté en une implémentation naïve concernant les différents aspects mentionnés précédemment.

A. Sélection

La première solution implémentée consiste en la sélection des n meilleurs individus, n étant un hyper-paramètre à faire varier en fonction de la taille de la population initiale.

Algorithm 1: Algorithme de sélection naïve

```
Result: n meilleurs individus
listeFitness initialisée;
n = 20;
selection = choisir(trier(listeIndividus, listeFitness), n);
```

Le principe de l'algorithme est de trier par ordre décroissant selon la fitness les individus. Ensuite, les n meilleurs sont choisis. L'avantage principal de cette méthode est que les moins bons individus sont écartés, ce qui rappel le fonctionnement de l'évolution naturelle. Toutefois, en procédent de la sorte, il est possible d'obtenir les mêmes individus ayant des schèmes similaires. Un schème est un ensemble de gènes qui sont partagés par plusieurs individus. La diversité de la population peut en être affectée. Nous verrons par la suite quels moyens nous avons mis en oeuvre pour pallier à ce problème.

B. Cross-over

Une position est choisie aléatoirement au niveau du plus petit chromosome parent. Les chromosomes sont coupés en ce point pour former deux enfants possèdant une partie de chaque génotype des parents.

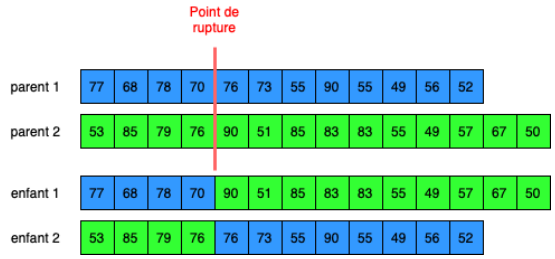

Figure 2. Exemple de cross-over avec un seule point de rupture

De la même manière que précédemment, cette méthode rappelle la fonction dont la reproduction sexuée s'effectue dans le monde réel. L'ADN de l'enfant provient à 50% de la mère et à 50% du père. Dans notre cas, il a été préférable de choisir le point de rupture de manière aléatoire afin de favoriser la diversité au sein de la population.

C. Mutation

Les mutations s'effectuent sur le génotype de l'individu. Chaque individu peut faire l'objet de trois mutations distinctes : modification d'un gène, ajout d'un gène et suppression d'un gène. Tout d'abord, chaque gène à une probabilité d'être modifié. Ainsi, le génotype est composé de parties indépendantes les unes des autres. Une nouvelle fois, l'avantage est d'augmenter la diversité des individus. Cependant, la moyenne des fitness sans voit diminuée.

D. Résultats

Cette implémentation naïve n'a pas apporté de résultats convainquants. Lors de nos expérimentations avec cet algorithme, nous n'avons pas réussi à trouver le mot de passe. Après quelques essais, ajouter de l'élitisme à apporter de meilleurs résultats. Lors de l'étape de reproduction, le meilleur individu de la génération précédente est conservé. Cet individu ne fait pas l'objet de mutations. L'objectif est d'assurer que la génération suivante évolute dans le bon sens.

Table II
HYPER-PARAMÈTRES UTILISÉS

Taille Population	N meilleurs	P(cross-over)	P(mutation)
300	50	30%	9%

La probabilité de cross-over joue un rôle important dans la convergence de l'algorithme. En effet, les hyper-paramètres de la table 2 permettent d'atteindre un minimum local en peu de générations. Toutefois, il est difficile de sortir des ces plateaux comme le montre les figures suivantes.

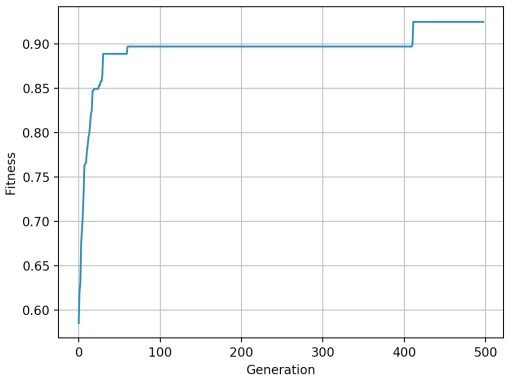

Figure 3. Evolution de la fitness maximum au cours des générations

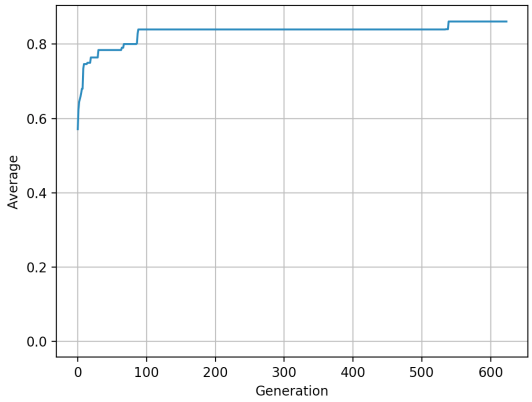

Figure 4. Evolution de la moyenne des fitness au cours des générations

En étudiant les deux figures, nous constatons que pour sortir d'un minimum local, plus de 400 générations sont nécessaires. Malgré le fait que la solution ne soit pas trouvée en temps acceptable, i.e environ 500 générations, la moyenne des fitness de la population reste importante.

V. Piste d'amélioration de l'algorithme naïf

En partant de l'algorithme réalisé, nous analysons l'impact de la sélection des individus dans la convergence de l'algorithme et pour l'amélioration des résultats.

A. Sélection

Au lieu de choisir simplement les n meilleurs individus, les meilleurs individus appartenant à des familles différentes sont sélectionnés. On reprend ici la notion de schème. Un individu est rattaché à une famille si il présente un taux de similitude dépassant les 75%.

B. Résultats

La modification apportée à l'algorithme naïf concernant la sélection n'offre pas de meilleurs performances, voire les amoindries.

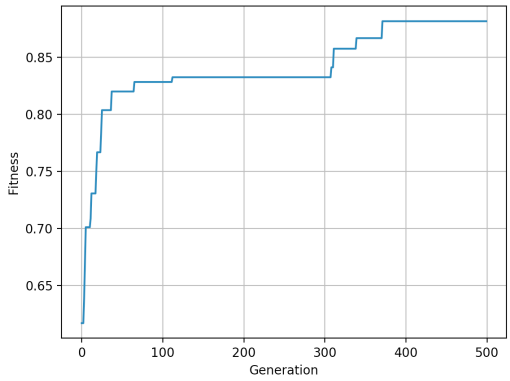

Figure 5. Evolution de la fitness maximum pour les individus de familles différentes

Nous pouvons conclure de la figure 5 que l'absence de bons résultats ne provient pas de la méthode de sélection. Ainsi, dans les itérations suivantes de notre algorithme, nous explorons d'autres méthodes concernant le cross-over et les mutations.

VI. Algorithme avec sélection par le rang

Dans cette nouvelle partie, nous testons de nouvelles approches afin d'améliorer les performances de l'algorithme.

A. Sélection par le rang

Cette fois-ci, nous implémentons l'algorithme de la sélection par le rang. Tout d'abord, les individus sont classés par ordre décroissant selon leurs scores de fitness. Ensuite, à chaque individu est attribué un poids. Pour calculer ce poids, nous utilisons une sélection exponentielle où r est le rang, N est le nombre total d'individus et c une constante à fixer :

$$w(r) = \frac{c-1}{c^N - 1} c^{N-r} \tag{1}$$

B. Cross-over uniforme

Deux parents sont sélectionnés pour produire un seul enfant. La particularité de ce cross-over est que les gènes des parents ont la même chance d'être exprimés dans le génotype de l'enfant. Cette technique apporte une plus grande diversité et une équité entre les individus. Ainsi, le meilleur individu aura le même impact reproducteur que les autres membres de la population.

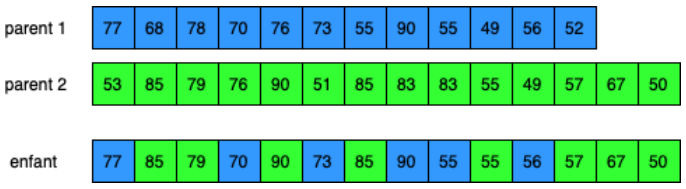

Figure 6. Exemple de cross-over uniform générant un enfant.

C. Résultats

L'évolution de la fitness est non constante et la valeur maximale est très faible. Ce phénomène peut s'expliquer par le fait que lors de la sélection par le rang, les meilleurs individus ont une importance moindre, ce qui donne plus de poids aux moins bons individus. L'effet est accentué par le cross-over uniforme.

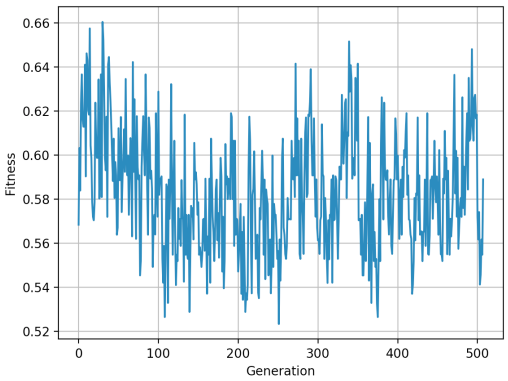

Figure 7. Evolution de la fitness maximum avec une sélection par le rang

VII. Algorithme avec différentes mutations et ajout de populations

Dans cette dernière partie, nous conservons les meilleures méthodes des itérations précédentes qui sont une sélection simple des meilleurs individus et un cross-over uniforme.

A. Nouvelles mutations

Au lieu de faire une mutation possible par gène, nous faisons une mutation par génotype. Ainsi, nous augmentons le seuil initial de mutation puisque le nombre de mutations sera limité. Une fois le seuil passé, nous attribuons une probabilité à chaque mutation possible. Une fois de plus, afin que l'algorithme reste proche du monde vivant, nous donnons plus d'importance à une simple modification de gène. De plus, nous ajoutons le fait que deux gènes puissent intervertir leur place. D'après les observations des expérimentations précédentes, certains mots de passes étaient pratiquement corrects à l'exception près que deux gènes étaient inversés.

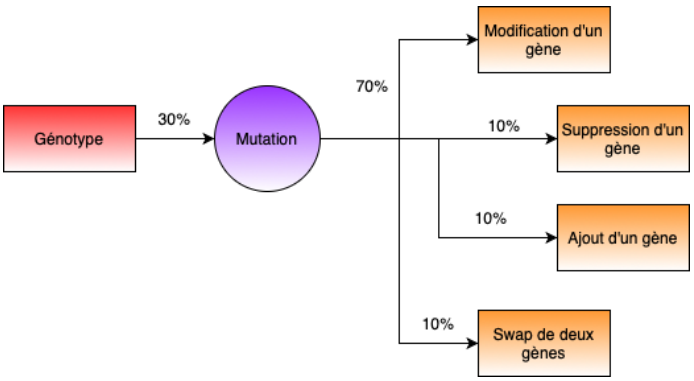

Figure 8. Flowchart des mutations possibles pour un génotype.

B. Multiples populations

Pour notre dernière expérimentation, nous voulions également nous éloigner des méthodes traditionnelles. Dans cette nouvelle version de l'algorithme, le nombre de populations devient un hyper-paramètre. n populations indépendantes sont créées puis rassemblées en un seule au bout de k générations. L'avantage apporté par cette technique est de pouvoir explorer l'espace plus largement. Le merge des populations favorise le mélange des meilleurs solutions.

C. Résultats

Finalement, les résultats sont meilleurs et les populations arrivent à découvrir le mot de passe en temps acceptable. La table 3 résume nos expérimentations suivant les hyper-paramètres choisis. En moyenne, un solution est trouvée autour de 200 générations. Le score correspond au pourcentage de réussite sur 100 itérations avec un nombre maximal de 1000 générations.

Table III
Résultats en fonction des hyper-paramètres

P(cross-over)	# Individus	# Populations	Merge	Score
30%	200	5	50	50%
30%	100	5	50	45%
30%	100	10	50	80%
30%	100	20	50	90%

VIII. Discussion

Les algorithmes génétiques sont des outils intéressants qui cherchent à reproduire l'évolution des espèces vivantes. L'ADN et les chromosomes sont au centre du fonctionnement de ces algorithmes. Dans notre étude, les algorithmes génétiques ont montré qu'il n'y a pas une seule manière de procéder. Il est nécessaire de faire évoluer l'algorithme en fonction du problème à résoudre. Nous avons tout d'abord, implémenter un algorithme naïf avec un cross-over simple point de rupture et une mutation possible par gène. Ensuite, nous avons testé diverses méthodes qui ont apporté des résultats plus ou moins bons. Tout au long de notre étude, nous travaillons itérativement en réalisant des modifications minimes afin de souligner précisément quelles sont les techniques pertinentes pour la découverte de mot de passe.