

# Étude de l'uniformisation de la population des algorithmes génétiques

Philippe Preux\*

Laboratoire d'Informatique du Littoral, Calais, France  
et

LIFL, URA CNRS 369, Villeneuve d'Ascq, France

Publié dans les actes de "Évolution Artificielle'94"  
Toulouse, France, Sep. 1994

## Abstract

*Dans cet article, nous considérons les algorithmes génétiques comme des heuristiques utiles à la résolution de problèmes compliqués. Nous étudions le processus d'uniformisation génotypique de la population d'un algorithme génétique. Des expériences montrent que cette uniformisation est très rapide quelle que soit la fonction objectif, après quoi la diversité génotypique de la population demeure constante. Bien que prévue par les modèles théoriques, une diversification de la population n'est pas observée "dans un futur proche". Nous concluons que si l'on veut utiliser les algorithmes génétiques comme des heuristiques, cette phase d'uniformisation doit être étudiée plus avant en cherchant, par exemple, à l'allonger pour que l'optimum ait une plus grande chance d'être trouvé.*

---

\*. [Philippe.Preux@lifl.fr](mailto:Philippe.Preux@lifl.fr). Toute correspondance doit être envoyée à l'adresse suivante : Cité Scientifique, LIFL – Bâtiment M3, 59655 Villeneuve d'Ascq Cedex, France. Ce papier est dédié à la mémoire de Lenneke Dekker.



# 1 Introduction

Les algorithmes génétiques<sup>1</sup> sont couramment proposés comme des heuristiques, utiles à la résolution de problèmes non polynômiaux. Même si de réels succès ont été obtenus, force est de constater que leur utilisation demeure pour l'essentiel intuitive. Des expériences ont donné quelques pistes pour fixer des valeurs robustes de paramètres comme la taille de la population ou le taux d'application des opérateurs (voir par exemple [13, 22]). Cependant, nous ne disposons d'aucune aide en ce qui concerne la définition d'opérateurs (robustes) qui pourraient se révéler intéressants, l'écriture de la fonction objectif, le codage de l'espace de recherche, donc des chromosomes, du choix de la méthode de sélection, ... Des travaux théoriques, initiés par J. Holland [16] (théorème des schémas, parallélisme intrinsèque) ont essayé d'expliquer le fonctionnement des AG. Ces travaux ont effectivement été utilisés pour construire des problèmes difficiles à résoudre pour un AG en utilisant l'hypothèse des briques de base<sup>2</sup> [10, p. 41–45] (problème décevant [9, 29]) ou prédire la taille optimale de la population [11], avec quelques succès. Cependant, le théorème des schémas a été récemment critiqué ainsi que l'hypothèse des briques de base ([15], [18], [8]). Ainsi, J. Grefenstette a montré que les problèmes décevants ne sont difficiles que si l'on se restreint à la version initiale des AG [15]. Même si cette version initiale, qualifiée de *canonique*, autorise une formalisation, a priori relativement simple du fonctionnement des AG, il semble fort dommage de se restreindre à cette version si l'on veut utiliser les AG comme des heuristiques. Même en demeurant robustes, les AG peuvent, et doivent selon nous, différer de la version canonique. Si besoin est, il faut chercher de nouvelles formes des AG plus adaptées à leur utilisation comme des heuristiques, plutôt que de s'en tenir à l'algorithme canonique par principe et parce que les résultats théoriques obtenus ne sont valables que pour lui ou presque.

Sur un autre plan, des modèles ont été développés, généralement basés sur une analyse à l'aide de chaînes de Markov [7, 20, 5, 27, 17, 26], montrant que les AG trouvent l'optimum avec une probabilité égale à 1. Si ces résultats sont intéressants dans leur principe, il faut noter que la découverte de l'optimum n'est démontrée qu'au bout d'un temps infini. Si cela est rassurant de savoir que les AG peuvent découvrir l'optimum, cela nous laisse démunis quand il s'agit d'utiliser un AG pour résoudre un problème. Dans ce cas, nous aimerions savoir à partir de quand une solution (l'optimum?) a une certaine probabilité d'être effectivement découverte. Une autre question digne d'intérêt est celle de connaître, si on arrête l'évolution de l'AG après un certain nombre d'itérations, la qualité de la solution trouvée à ce moment-là. Ces questions nous paraissent d'autant plus légitimes que les résultats théoriques ne nous disent rien sur l'état à un certain instant de la population de l'AG avant un temps infini, on ne sait donc jamais si l'AG a trouvé l'optimum ou non. Tout ce que

l'on peut utiliser est un critère de convergence de la population qui ne nous donne aucune information sur la qualité de la solution trouvée. Si l'on compare les AG à une simple procédure énumérant tous les points de l'espace de recherche et retenant, à tout instant, le meilleur trouvé jusqu'alors, on sait que cette procédure s'arrête et qu'elle trouve l'optimum. Pour leur part, les AG ne s'arrêtent pas, si l'on désire être sûrs d'obtenir l'optimum.

À la suite de ces réflexions, nos travaux visent à comprendre le comportement "accessible" des AG. Par accessible, nous voulons signifier que nous nous intéressons à savoir comment s'uniformise la population. En principe, la population s'uniformise au bout d'un certain temps. Puis, elle se diversifie, se ré-uniformise, etc, infiniment, convergeant "alors" sur l'optimum. Aussi, nous désirons connaître comment et vers quel point de l'espace de recherche la population se concentre. Les buts de ces travaux sont multiples :

- comprendre le comportement d'une procédure d'optimisation robuste
- comprendre l'influence des opérateurs sur le comportement
- maîtriser la pression de sélection
- éviter les convergences prématurées
- guider le choix du génotype
- guider l'écriture de la fonction objectif
- avoir un modèle réellement utilisable

Il faut également indiquer que nous utilisons le mot AG pour désigner, en général, toute procédure stochastique reposant sur la reproduction, la sélection et l'application d'opérateurs sur les membres d'une, ou plusieurs, populations, communicantes ou non. Nous ne nous restreignons pas à la seule utilisation des opérateurs génétiques classiques, ni à une représentation binaire des chromosomes. Ainsi, stratégies évolutives, programmations évolutive et génétique entrent clairement dans le champ de nos travaux. Dans le présent article, nous traiterons du cas des chromosomes composés d'allèles binaires. Nous utiliserons généralement une stratégie de reproduction SUS (*Stochastic Universal Sampling*) [2].

Dans la suite de cet article, nous décrivons nos premiers résultats. Ils concernent une étude de l'uniformisation de la population d'un AG. Pour cela, nous étudions l'évolution au cours du temps de la diversité des génotypes présents dans la population. Ce cadre de travail est présenté dans la section 2 où l'évolution prévisible est décrite. Dans la section 3, des expériences sont décrites, montrant l'évolution de cette diversité pour l'optimisation d'une fonction numérique. La section 4 complète cette étude expérimentale avec l'optimisation de fonctions conçues pour tester certains points. Enfin, nous discutons nos résultats et indiquons les suites de ce travail.

1. désormais abrégés par AG

2. *building block hypothesis*

## 2 Méthodologie

### 2.1 Phénotype et génotype

Nous désirons étudier l'uniformisation au cours du temps de la population d'un AG. Pour cela, nous devons définir une mesure. L'uniformité, ou la variété, d'une population peut se définir à deux niveaux, génotypique ou phénotypique. Il peut être utile de bien noter ce qui se passe à chacun de ces niveaux. Le niveau phénotypique est le plus apparent pour l'utilisateur car :

- les chromosomes y sont vus comme des structures de données ;
- la fonction objectif évalue les chromosomes à ce niveau ;
- le processus de sélection agit à ce niveau.

Le niveau génotypique est le niveau profond du processus où :

- le codage des chromosomes est défini ;
- les opérateurs génétiques agissent ;
- le processus évolutif agit à ce niveau totalement mécaniquement en manipulant des chaînes de bits.

Ce qui se passe au niveau phénotypique est toujours le reflet de ce niveau profond qu'est le niveau génotypique. Aussi, il faut comprendre ce qui se passe à ce niveau profond pour comprendre et prédire ce qui se passe au niveau phénotypique.

Avant d'entrer plus avant dans les détails, nous voudrions insister sur un point fondamental. Il y a deux espaces de recherche, l'espace phénotypique et l'espace génotypique. Le premier est l'espace où la fonction doit être optimisée. Le second est l'espace réellement parcouru par l'AG. Le codage des chromosomes définit la transformation à opérer pour passer d'un espace à l'autre. Aussi, pour simplifier, on peut souhaiter que les deux espaces soient isomorphes. Il faut, en tout cas, que l'optimum de l'espace de recherche phénotypique (la solution au problème qui nous intéresse) corresponde à l'optimum de l'espace génotypique. Nous reviendrons plus tard sur ce codage de l'espace phénotypique. Nous voulons simplement indiquer ici que lorsque nous parlerons de l'espace de recherche  $\mathcal{E}$  dans cet article, il s'agira de l'espace de recherche génotypique, à moins que cela ne soit spécifier explicitement différemment.

### 2.2 La diversité de la population

Dans le cadre de cet article, nous ne considérons que des chromosomes binaires. Aussi, nous définissons la *diversité génétique* comme une mesure de la diversité de la population au niveau génotypique. A ce niveau,

la distance entre deux chromosomes est le nombre d'allèles respectifs qui diffèrent ; c'est donc simplement la distance de Hamming entre les deux chaînes de bits. Afin de mesurer la distance globale dans la population, nous prenons la somme sur toutes les paires de chromosomes  $(\chi_i, \chi_j)$  de la population  $P$  de cette distance  $\Delta_H(\chi_i, \chi_j)$  :

$$\Delta_H = \sum_{\{\chi_i, \chi_j\} \in P \times P, i > j} \Delta_H(\chi_i, \chi_j)$$

Nous noterons  $\Delta_H^t(P)$  la diversité génotypique de la population  $P$  à l'instant  $t$ , ou simplement  $\Delta_H(P)$  si  $t$  n'importe pas.

### 2.3 L'évolution de la diversité

Nous présentons comment varie la diversité génotypique de la population au cours des différentes phases de l'évolution d'un AG. A chaque génération  $t$ , la population  $P$  est caractérisée par  $\Delta_H^t(P)$ . Initialement, à  $t = t_0$ , une population de  $\nu$  individus, chacun composé de  $\lambda$  gènes binaires est générée aléatoirement. Aussi,  $\Delta_H^{t_0}$  est :

$$\Delta_H^{t_0} = \frac{\nu(\nu - 1)\lambda}{4}$$

Au cours d'une génération, l'AG effectue les étapes suivantes :

**reproduction** La population des ancêtres  $P$  donne une population de descendants  $P'$ . En utilisant la reproduction SUS, un individu peut donner deux enfants, un enfant ou aucun. En moyenne, un chromosome donne  $\varphi(i)$  enfants, avec  $\varphi(i)$  défini par :

$$\varphi(i) = \eta_{\max} - (\eta_{\max} - \eta_{\min}) \frac{i - 1}{\lambda - 1}$$

où  $\eta_{\max} = 2.0 - \eta_{\min}$  et  $0.0 \leq \eta_{\min} \leq 1.0$ .  $\varphi(i)$  est arrondi stochastiquement pour donner un nombre entier d'enfants. L'effet est que la diversité génotypique est réduite par SUS puisque deux enfants jumeaux ont leur distance de Hamming qui est nulle. Donc,  $\Delta_H(P') < \Delta_H(P)$

**application des opérateurs génétiques** Nous nous intéressons ici uniquement au crossover et à la mutation.

**crossover** Le crossover ne modifie en rien la diversité génotypique. En effet, le crossover agit par un simple échange d'allèles entre deux chromosomes. Aussi, la distance de Hamming entre les deux allèles correspondant de deux chromosomes demeure-t-elle identique, que les allèles soient échangés ou non. Notons que ce raisonnement est vrai quel que soit le type de crossover utilisé (1-, 2-, m- points ou uniforme) ;

**mutation** Considérant que la mutation agit en forçant un allèle avec une valeur choisie au hasard, le forçage par un 0 ou un 1 est équi-probable. Aussi, en moyenne,  $\Delta_H$  demeure inchangé. Plus précisément, on peut déterminer des valeurs minimales et maximales statistiques d'une population  $P'$  mutée  $\Delta_H(\mu(P'))$ , où  $\mu$  est l'opérateur de mutation :

$$\begin{aligned} \Delta_H(P') - \frac{R_\mu \nu (\nu - 1) \lambda}{2} \\ \leq \Delta_H(\mu(P')) \leq \\ \Delta_H(P') + \frac{R_\mu \nu (\nu - 1) \lambda}{2} \end{aligned}$$

où  $R_\mu$  est le taux de mutation.

**sélection** tant que l'écart entre les générations<sup>3</sup>  $\gamma$  vaut 1, la population de la génération suivante consiste dans la population  $P'$  des descendants, après application des opérateurs. Aussi, nous avons :

$$\Delta_H^{t+1}(P) = \Delta_H^t(\mu(X_{\text{Over}}(P')))$$

où  $X_{\text{Over}}$  dénote l'opérateur de crossover.

Dans le cas général où  $\gamma \neq 1.0$ , nous n'avons pas d'expression analytique correcte de  $\Delta_H^{t+1}(P)$ . Des expériences concernant ces cas sont présentées plus loin.

Ces quelques remarques nous indiquent que la population tend à s'uniformiser. Seule la mutation peut diversifier la population. Il faut remarquer que nous n'avons pas caractérisé la vitesse de processus d'uniformisation

$$\frac{d\Delta_H^t(P)}{dt}$$

Nous n'avons pas d'expression analytique de cette vitesse. Nous pensons cependant pouvoir en obtenir une rapidement. Plus difficile est la caractérisation du point vers lequel la population s'uniformise.

De nombreuses expériences nous ont montré que la vitesse d'uniformisation est élevée. Nous les décrivons brièvement dans la section suivante.

### 3 Expérimentation

Nous avons étudié expérimentalement le comportement de  $\Delta_H$  au cours du temps. Pour cela, nous avons optimisé une fonction numérique nommée  $f6$  dans [22] :

$$f6(x, y) = 0.5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$$

3. generation gap en anglais

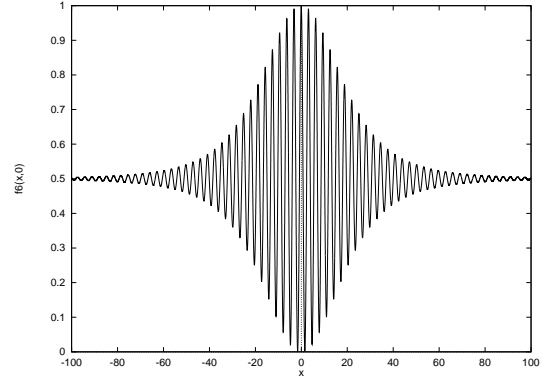


FIG. 1 - Fonction  $f6(x, y)$  quand  $y$  vaut 0

Cette fonction est symétrique par rapport à l'axe  $z$ . On recherche son maximum qui atteint au point  $(0, 0)$  pour lequel  $f6(0, 0) = 1$ . Elle est représentée à la figure 1. On voit qu'elle comporte un grand nombre d'optima locaux qui rendent son optimisation difficile.

Toutes nos simulations sont réalisées à l'aide de notre simulateur EGG (pour *Enhanced Grefenstette's Genesis*[19]) qui, comme son nom l'indique, est une extension du simulateur *genesis* v5.0 de J. Grefenstette [14].

Nous avons utilisé les paramètres standards de *genesis* excepté la stratégie de reproduction SUS : taille de la population  $\nu = 50$ , nombre de gènes par chromosome  $\lambda = 60$ , taux de crossover  $R_\chi = 0.6$ , taux de mutation  $R_\mu = 0.001$ , écart entre les générations  $\gamma = 1.0$ , et SUS avec rang minimum  $\eta_{\min} = 0.75$  et rang maximum  $\eta_{\max} = 2 - \eta_{\min}$ .

Un chromosome est composé de deux parties de même taille, l'une codant la valeur de  $x$ , l'autre la valeur de  $y$ . Chaque valeur varie entre  $-5.12$  et  $5.12$ . On utilise un codage binaire de Gray. L'intervalle  $[-5.12, 5.12]$  est uniformément découpé en  $2^{\frac{5}{2}}$  morceaux. Chacune des extrémités de ces morceaux correspond à un point de  $\mathcal{E}$ .

La fonction objectif est écrite très simplement : elle retourne la valeur de  $f6(\chi)$  où  $\chi$  est un chromosome à évaluer.

#### 3.1 L'uniformisation

La figure 2 représente la variation de  $\Delta_H(P)$  au cours du temps. On constate que sa valeur décroît très rapidement et que la courbe peut être caractérisée en deux parties :

- une décroissance rapide de  $\Delta_H(P)$  durant les 300 premières générations ;
- une phase stable "éternelle".

Nous avons réalisé de nombreuses expériences en faisant varier les paramètres systématiquement, ce, sur de nombreuses fonctions. Il est frappant de constater

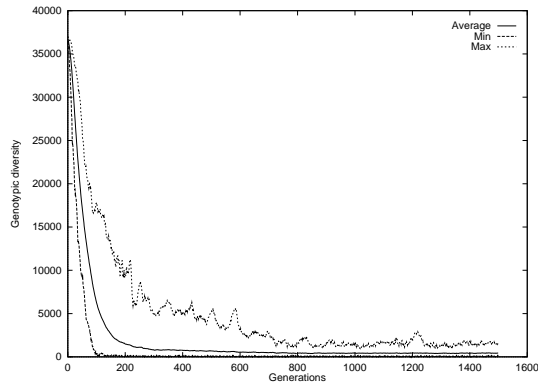


FIG. 2 - Évolution de  $\Delta_H$  lors de la résolution de  $f_6$ . Les trois courbes sont des moyennes sur 100 expériences. Elles indiquent la valeur moyenne observée ainsi que les bornes inférieure et supérieure. Seules les 1500 premières générations sont représentées et le début de la phase stable. Durant cette dernière phase,  $\Delta_H$  garde une valeur autour de 420. Les paramètres de cette simulation sont ceux indiqués précédemment.

que  $\Delta_H(P)$  évolue toujours de la même manière, avec une phase de décroissance rapide qui se termine autour de la 1000<sup>ème</sup> génération, suivie d'une phase stable "éternelle".

Nous appelons la première phase *uniformisation*, la deuxième *phase stable*. Ces deux phases peuvent être assimilées aux phases d'exploration et d'exploitation. Nous avons placé des guillemets autour de "éternelle" car la théorie des AG nous indique que cette phase, même si elle dure longtemps, n'est pas éternelle tant que l'optimum n'a pas été découvert. Cependant, une diversification de la population n'a jamais été observée durant nos expériences, bien qu'elles aient été poussées très loin dans le temps (plusieurs millions de générations).

La phase stable est caractérisée par ce que nous nommons un *bruit de fond*, qui est cette valeur résiduelle de  $\Delta_H(P)$ . Pour l'expérience décrite ici,  $\Delta_H^{t > 1000} = 420$ . Cette valeur caractérise la diversité de la population atteinte à la phase stable. Elle doit être comparée à la valeur initiale  $\Delta_H^{t_0} = 36750$ . La diversité est donc réduite d'un facteur 100.

Comme nous l'avons indiqué plus haut, la mutation seule (dans le cadre que nous nous sommes fixés) peut diversifier la population. D'ailleurs, dès que le taux de mutation est relativement élevé, la diversité demeure importante durant la phase stable (cf. figure 3). Une uniformisation de la population n'est plus alors possible, du fait des mutations continues. Plutôt que d'utiliser un taux de mutation fixe durant toute l'évolution, nous avons réalisé des expériences où le taux de mutation est normal ( $10^{-3}$ ) avec, à intervalles réguliers (100 générations), une fois la population relativement uniformisée, un très fort taux de mutation (0.5) pendant 10 générations. Alliée à une

stratégie élitiste, cette méthode réalise une espèce de ré-initialisation de l'ensemble de la population, tout en conservant le meilleur individu trouvé jusqu'alors. Avec les paramètres que nous utilisons, le maximum de  $f_6$  est normalement trouvé dans 8% des expériences. Avec cette stratégie de forte mutation à intervalles réguliers, le maximum est trouvé dans 80 % des expériences (pour des expériences de durées égales).

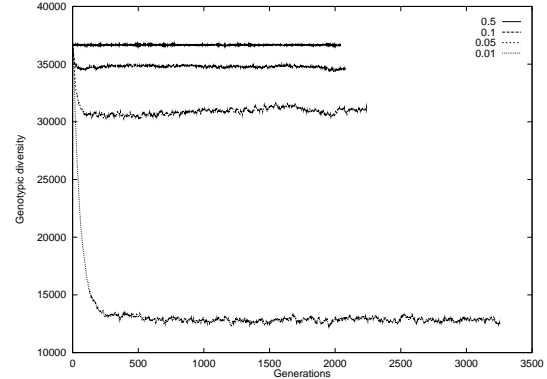


FIG. 3 - Diversité génétique en fonction du taux de mutation. Pour des taux de mutation élevés, la phase d'uniformisation disparaît. (Les expériences ont été réalisées sur  $f_6$ .)

### 3.2 D'autres expériences

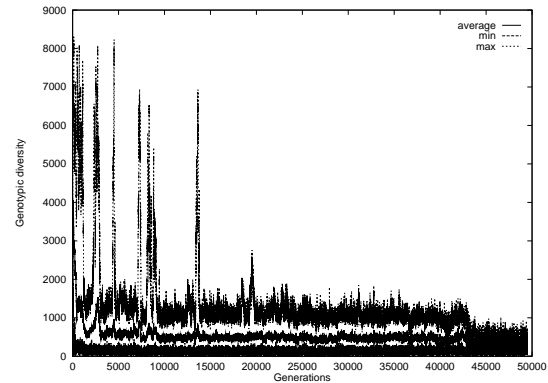


FIG. 4 - Évolution de  $\Delta_H$  lors de la résolution de  $f_6$  en partant de populations initiales uniformes. Les paramètres de la simulation demeurent identiques aux expériences précédentes. On constate des re-diversifications durant les 20000 premières générations.

Nous avons utilisé quelques techniques implantées dans EGG pour déterminer la robustesse de l'uniformisation. Nous avons effectué des expériences avec les

techniques suivantes :

**élitisme** EGG implante deux versions de l'élitisme :

- élitisme standard dans lequel le meilleur individu est conservé d'une génération sur l'autre ;
- l'élitisme habituel accompagné de la conservation du complémentaire binaire du meilleur individu. Cette méthode a été proposée par J. Grefenstette pour éviter de tomber dans le piège des fonctions décevantes [15] ;

**auto-adaptation des taux de mutation**

Cette technique est héritée des stratégies évolutives et introduite dans le domaine des AG par T. Bäck [1].

Cette technique consiste à associer à chaque individu son propre taux de mutation. Il n'y a donc plus de taux de mutation global. Ce taux de mutation est utilisé pour muter les gènes de l'individu. Par ailleurs, le taux de mutation est lui-même muté en fonction de sa propre valeur ;

**population initiale biaisée** La population est alors initialisée avec un seul type d'individu.

Toutes ces expériences ont donné les mêmes résultats, le comportement décrit précédemment. Nous avons également réalisé des expériences en faisant varier systématiquement la cardinalité de la population ou l'écart entre les générations. Aucune variation n'a été observée.

Dans le cas d'une population initialement uniforme, celle-ci subit tout d'abord une phase de diversification qui est suivie d'une phase d'uniformisation et d'un état stable (cf. figure 4). La phase de diversification est assez courte (50 générations environ) mais la phase d'uniformisation est beaucoup plus longue que dans le cas d'une population non biaisée. Des re-diversifications tardives sont observées jusque vers la 20000<sup>ème</sup> génération. Après stabilisation, le bruit de fond est alors du même ordre de grandeur que celui obtenu à partir d'une population initiale aléatoire. La diversité continue à varier autour d'une valeur moyenne beaucoup plus fortement que dans le cas d'une population initiale non uniforme.

## 4 L'uniformisation

Dans cette section, nous étudions le bruit de fond ainsi que la phase d'uniformisation que nous relierons à la notion de pression de sélection. Quelques expériences complémentaires sont décrites. Elles utilisent des fonctions très simples, spécialement conçues pour confirmer nos hypothèses, et lever quelques problèmes.

### 4.1 Bruit de fond et pression de sélection

Lorsque la phase stable est atteinte, une partie du génotype des chromosomes est fixé, le même pour tous.

Celle-ci constitue un schéma  $\sigma$  qui caractérise la région de l'espace de recherche vers laquelle l'AG converge. Les bits non fixés de  $\sigma$  n'ont pas d'importance sur le taux d'adaptation du chromosome et peuvent continuer à varier : toutes les instances de  $\sigma$  ont le même taux d'adaptation, donc les mêmes chances de survie. Par ailleurs, l'ensemble  $\mathcal{S}$  constitué des instances de  $\sigma$ ,  $\mathcal{S} = \{\chi \in \sigma\}$  est clos par crossover, c'est-à-dire :

$$\forall(\chi_i, \chi_j) \in \mathcal{S} \times \mathcal{S}, X_{\text{over}}(\chi_i, \chi_j) \in \mathcal{S} \times \mathcal{S}$$

En effet, si  $\mathcal{S}$  n'était pas clos par crossover durant la phase stable, cette phase ne serait pas stable.

Par ailleurs, durant la phase stable, les chances d'engendrer un individu ayant un meilleur taux d'adaptation sont réduites. Sinon, la mutation aurait agi avant que la phase stable ne soit engagée et aurait produit de meilleurs individus. Donc,  $\mathcal{S}$  est stable par crossover et mutation. Le nombre de gènes pouvant varier sans conséquence sur le taux d'adaptation des individus sera nommé  $\beta_{lim}$ .

$\beta_{lim}$  est donc clairement lié au niveau de bruit présent à l'état stable. Soit  $\beta$  le nombre de bits qui peuvent changer parmi les individus d'une population (c'est-à-dire, le nombre de bits non fixés de  $\sigma$ ),  $\Delta_H$  de cette population est :

$$\Delta_H = \begin{cases} \frac{\nu(\nu-1)\beta}{4} & \text{if } 2^\beta > \nu \\ (1 - 2^{1-\beta})\nu^2 & \text{if } 2^\beta < \nu \end{cases}$$

$\beta_{lim}$  est influencé par deux facteurs :

- la minceur du pic où est localisé l'optimum ;
- le codage des chromosomes ou, plus précisément, le nombre d'individus de même taux d'adaptation que la région de l'espace de recherche phénotypique peut contenir

### 4.2 Expériences

Afin de confirmer expérimentalement les énoncés précédents, nous avons réalisé des expériences supplémentaires avec des fonctions objectives spécialement conçues pour cela. Nous utilisons :

- une fonction "trou" dont la topologie est très simple : un plateau creusé d'un trou profond. La fonction doit être minimisée. Nous faisons évoluer le diamètre du trou pour faire varier le nombre de chromosomes pouvant tomber dedans. Le but est de voir si le bruit de fond peut être mis sur le compte du crossover agissant sur un sous-ensemble d'individus. Étant donnée la taille du trou, on peut déterminer l'ensemble  $\mathcal{S}$  clos par crossover des individus pouvant y tomber. Les résultats sont indiqués dans la table 1. Il existe une nette corrélation entre les prédictions théoriques et les valeurs observées. Le nombre d'individus différents dans la population uniformisée est inférieur au nombre théorique, confirmant bien notre hypothèse. Le rapport entre les valeurs théoriques et pratiques n'est pas encore expliqué ;

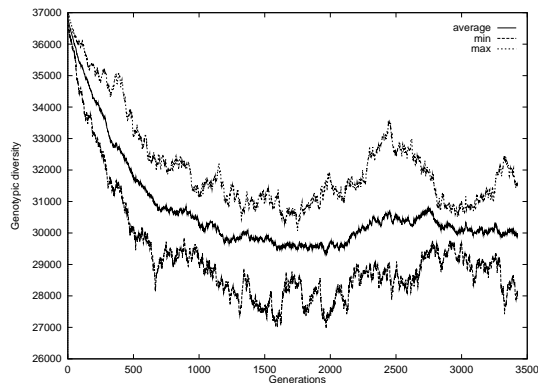


FIG. 5 - Évolution de  $\Delta_H$  sur la fonction plate en utilisant une stratégie de sélection roulette wheel. La diversité de la population demeure très élevée.

- une fonction “plate” dont la topologie est encore plus simple: un plateau. Donc, tous les individus ont le même taux d’adaptation. Le but est d’observer le comportement des AG lorsqu’aucune information concernant la localisation de l’optimum dans  $\mathcal{E}$  n’est disponible dans l’environnement. La fonction “plate” peut être vue comme un cas extrême de fonction “trou” où le diamètre du trou est infini. Aussi, les résultats concernant cette fonction se trouvent dans la table 1, dans la colonne  $\infty$ .

Remarquons immédiatement que même dans le cas de la fonction plate, l’AG converge vers un point. Ceci peut être ennuyeux dans la pratique car cela signifie que l’AG “trouve toujours un optimum”, même quand il n’y en a pas. Notons que le niveau de bruit élevé peut indiquer que l’AG n’a pas convergé vers une région précise de  $\mathcal{E}$ .

Un autre point concerne l’effet de la pression de sélection sur l’uniformisation. La figure 6 montre la variation du bruit de fond à l’état stable en fonction du rang minimum  $\eta_{min}$ . Quand la pression de sélection est élevée (c’est-à-dire quand  $\eta_{min} \rightarrow 0.0$ ), le niveau de bruit, donc la diversité, est faible, et réciproquement. De nouveau, la fonction plate donne une borne supérieure du bruit de fond pour les fonctions trou, et pour toute autre fonction d’ailleurs, étant donnée sa “platitude”. La fonction plate converge rapidement à cause de la pression de sélection due à la méthode de sélection utilisée (ranking) qui ordonne les individus. Ainsi, certains individus sont arbitrairement avantagés, bien que n’étant pas meilleur que les autres, ce qui provoque une dérive génétique. L’utilisation d’une stratégie roulette wheel tempère largement cette dérive. La population conserve alors une diversité très élevée (cf. figure 5);

- des fonctions “aléatoires” n’ayant, a priori, pas de profil bien défini. Le taux d’adaptation est cal-

culé aléatoirement quand un individu est évalué et fournit une valeur dans l’intervalle  $[0, 1[$ . Avec ce genre de fonctions objectifs, l’AG ne peut recueillir aucune information valide sur l’emplacement de l’optimum dans  $\mathcal{E}$ . D’une manière aussi étonnante que pour la fonction plate, nous observons que la population converge rapidement autour d’un individu ayant un taux d’adaptation élevé.

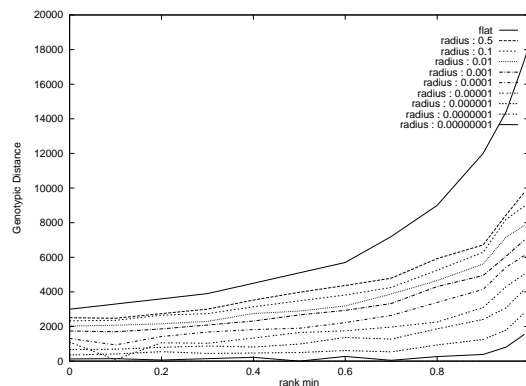


FIG. 6 - Variation de la distance génotypique vs.  $\eta_{min}$ . L’impact de la variation de  $\eta_{min}$  est indiqué ici sur la fonction “trou” et sur la fonction “plate”. Quand  $\eta_{min}$  croît, c’est-à-dire, quand la pression de sélection diminue, la diversité de la population augmente, et réciproquement. En utilisant notre définition de la pression de sélection,  $\rho = \eta_{max} - \eta_{min}$ ,  $\rho = 2$  pour  $\eta_{min} = 0.0$  et  $\rho = 0.0$  pour  $\eta_{min} = 1.0$ , avec une décroissance linéaire de  $\rho$  entre ces bornes.

## 5 Discussion

Dans cet article, nous avons décrit nos travaux en cours concernant l’étude du comportement des AG et l’uniformisation de leur population. Nous voulons ainsi obtenir des résultats pratiques concernant ce comportement, plutôt que des résultats théoriques difficilement applicables. Le but est de gagner un savoir-faire concernant l’utilisation des AG comme des heuristiques.

Etudiant l’évolution de la population, nous avons tout d’abord montré que les phases du processus (reproduction, application des opérateurs, sélection) tendent à diminuer sa diversité. Seule l’opérateur de mutation peut ajouter de la diversité dans la population. Un autre point concerne la très rapide uniformisation de la population. La conséquence immédiate est que la région de l’espace contenant l’optimum doit être trouvée très rapidement, sous peine de ne jamais la trouver (du moins, pas dans un temps raisonnable). L’uniformisation a été liée à la notion de pression de



Rayon du trou	$10^{-8}$	$10^{-7}$	$10^{-6}$	$10^{-5}$	$10^{-4}$	$10^{-3}$	$10^{-2}$	$10^{-1}$	$\infty$
Nombre d'individus pouvant tomber dans le trou	3	24	234	2334	23327	233248	2332450	23324470	$2^{30}$
Bruit de fond $N_l^{th}$	827	2291	4820	6853	8887	10921	12954	14994	18375
$\beta_{lim} = \log_2 N_l^{th}$	1.35	3.74	7.87	11.19	14.51	17.83	21.15	24.48	30
Bruit de fond $N_l^{exp}$	462	1164	2020	2646	3593	4371	4982	5816	8100
$\beta = \log_2 N_l^{exp}$	0.75	1.90	3.30	4.32	5.87	7.14	8.13	9.50	12.98
Ratio $\frac{\beta_{lim}}{\beta}$	1.80	1.97	2.38	2.59	2.47	2.50	2.60	2.58	2.31

TAB. 1 - *Bruit de fond théorique (les deux premières lignes) et expérimental (les deux lignes suivantes) pour les fonctions "trou". La colonne la plus à droite étiquetée  $\infty$  contient les données concernant la fonction "plate" qui peut être considérée comme une fonction "trou" dont le diamètre est infini. Dans ces expériences, les valeurs des chromosomes varient entre  $-5.12$  et  $5.12$ ,  $\lambda = 30$ ,  $\eta_{min} = 0.75$  et  $\nu = 50$ .*

sélection. Le bruit de fond rémanent caractérisant la diversité résiduelle de la population, une fois la convergence réalisée, a été lié au nombre de gènes pouvant varier et donnant des individus ayant le même taux d'adaptation.

La phase d'exploration peut être améliorée soit en la rendant plus rapide à trouver la bonne région de  $\mathcal{E}$ , soit en l'allongeant pour que l'espace soit fouillé de manière plus exhaustive. La pression de sélection a un rôle important à jouer car elle permet d'allonger la durée de l'exploration. Cependant, une valeur faible de la pression de sélection entraîne une faible distinction entre bons et mauvais individus. Nos travaux se poursuivent selon les directions suivantes :

**codage des chromosomes** c'est-à-dire le codage de l'espace phénotypique par l'espace génotypique. On sait qu'un codage adapté peut transformer toute optimisation de fonction en un problème aussi simple que celui de la maximisation du nombre de 1 dans une chaîne de bits [28]. Cependant, le calcul de ce codage est un problème NP-complet. Pour optimiser une fonction numérique, on s'en tient généralement à encoder la valeur de chacune des variables dans les chromosomes, la question étant l'utilisation ou non d'un codage de Gray. Peu de travaux étudient des codages différents. Nous avons réalisé quelques expériences où le codage est réalisé classiquement suivi d'un mélange des gènes dans les chromosomes. En effet, en utilisant un codage binaire, de Gray ou non, les bits composant les chromosomes ont un poids. Ce mélange a pour but de mixer bits de poids forts et bits de poids faibles. Les résultats sont surprenants, montrant une diversité lors de la phase stable bien supérieure à celle observée avec le codage simple utilisé dans le reste de cet article.

**fonction objectif** Des études concernant la définition des fonctions objectifs est nécessaire. Pour l'instant, la résolution d'un problème numérique est toujours réalisée en utilisant la fonction numérique elle-même comme fonction objectif. Une telle transcription étant moins simple pour les problèmes symboliques, quelques travaux ont été effectués concernant l'écriture de la fonction ob-

jectif dans le cas du TSP [31] ou du problème de satisfiabilité d'expressions booléennes [6] et ont donné de meilleurs résultats ;

**pression de sélection** Une meilleure utilisation de la pression de sélection est nécessaire. Ainsi, nous pensons que la pression de sélection doit s'adapter au cours de l'évolution de l'AG pour éviter des convergences prématurées.

## Références

- [1] Thomas Bäck. Self-adaptation in genetic algorithms. In [25], pages 263–271, 1991.
- [2] James E. Baker. Adaptive selection methods for genetic algorithms. In [12], pages 101–111.
- [3] Richard K. Belew and Lashon B. Booker, editors. *Proc. of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA, USA, 1991.
- [4] Lawrence Davis, editor. *Genetic Algorithms and Simulated Annealing*. Research Notes in Artificial Intelligence. Morgan Kaufmann, San Mateo, CA, USA, 1987.
- [5] Thomas E. Davis and Jose C. Principe. A simulated annealing like convergence theory for the simple genetic algorithm. In [3], pages 174–183, 1991.
- [6] Kenneth A. De Jong and William Spears. Using genetic algorithms to solve NP-complete problems. In [23], pages 124–132, 1989.
- [7] A. E. Eiben, E. H. L. Aarts, and K. M. V. Hee. Global convergence of genetic algorithms: a markov chain analysis. In [24], pages 1–12, October 1991.
- [8] Stephanie Forrest and Melanie Mitchell. Relative building-block fitness and the building-block hypothesis. In [30], pages 109–126, 1992.
- [9] David Goldberg. Simple genetic algorithms and the minimal deceptive problem. In [4], pages 74–88. 1987.
- [10] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [11] David Goldberg, Kalyanmoy Deb, and James H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6(4):333–362, 1992.

- [12] J. J. Grefenstette, editor. *Proc. of the First International Conference on Genetic Algorithms*, Pittsburgh, 1985. Lawrence Erlbaum Associates: Hillsdale, New-Jersey.
- [13] John Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):122–128, January 1986.
- [14] John J. Grefenstette. *A User's guide to Genesis Version 5.0*, October 1990. The *genesis* package is available via anonymous ftp on ftp.aic.nrl.navy.mil:pub/galist/src/ga/genesis.tar.Z.
- [15] John J. Grefenstette. Deception considered harmful. In [30], pages 75–91, 1993.
- [16] John H. Holland. *Adaptation in Natural and Artificial Systems*. Michigan Press University, Ann Arbor, MI, 1975.
- [17] Allen E. Nix and Michael D. Vose. Modeling genetic algorithms with markov chains. *Annals of Mathematics and AI*, 5:79–88, 1992.
- [18] Charles C. Peck and Atam P. Dhawan. A review and critique of genetic algorithm theories. Technical Report TR-153/6/93/ECE, Department of Electrical and Computer Engineering, University of Cincinnati, Cincinnati, OH 45221, USA, June 1993.
- [19] Philippe Preux. *Description of Special Features of Enhanced Grefenstette's Genesis*. Laboratoire d'Informatique Fondamentale de Lille, Université de Lille I, 59655 Villeneuve d'Ascq Cedex, France, April 1994. (working paper).
- [20] Y. Rabinovich and A. Wigderson. An analysis of a simple genetic algorithm. In [3], pages 215–221, 1991.
- [21] Gregory J. E. Rawlins, editor. *Workshop on the Foundations of Genetic Algorithm and Classifier*, Bloomington, IN, USA, 1991. Morgan Kaufmann, San Mateo, CA, USA.
- [22] J. David Schaffer, Richard A. Caruana, Larry J. Eshelman, and Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. [23], pages 51–60.
- [23] J.D. Schaffer, editor. *Proc. of the Third International Conference on Genetic Algorithms*, Bloomington, IN, USA, 1989. Morgan Kaufmann, San Mateo, CA, USA.
- [24] H-P. Schwefel and R. Männer, editors. *Proc. of the First Parallel Problem Solving in Nature*. Springer-Verlag, Berlin, 1991.
- [25] Francesco Varela and Paul Bourguine, editors. *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. MIT Press, Cambridge, MA, USA, December 1992. ISBN: 0-262-72019-1.
- [26] Michael D. Vose. Modeling simple genetic algorithms. In [30], pages 63–73, 1993.
- [27] Michael D. Vose and Gunar E. Liepins. Punctuated equilibria in genetic search. *Complex Systems*, 5:31–44, 1991.
- [28] Michael D. Vose and Gunar E. Liepins. Schema disruption. In [3], pages 237–242, July 1991.
- [29] Darrell Whitley. Fundamental principles of deception in genetic search. In [21], pages 221–241, 1991.
- [30] Darrell Whitley, editor. *Proc. of the Workshop on Foundations of Genetic Algorithms*, Vail, CO, USA, 1993. Morgan Kaufmann, San Mateo, CA, USA.
- [31] Darrell Whitley, Timothy Starkweather, and D'Ann Fuquay. Scheduling problems and traveling salesman: The genetic edge recombination operator. In [23], pages 133–140, 1989.