

Template Rapport Scientifique

Noé Bourgeois

Résumé—Nous avons implémenté un simulateur numérique de locomotion de créature virtuelle. Ce simulateur est composé d'un environnement, une créature s'y déplaçant en fonction de son génome, et un algorithme génétique permettant de modifier ce génome pour améliorer le comportement de la créature. Nous avons ensuite utilisé ce simulateur pour étudier l'impact de différents paramètres sur la vitesse d'évolution de populations de créature. Une population de départ générée aléatoirement ayant évolué au long d'une simulation, nous conservons ses individus et les remplaçons comme population de départ dans la simulation suivante.

I. INTRODUCTION

VIE et évolution sont deux concepts intimement liés. Cette force étrange et fascinante s'est répandue jusqu'ici via différentes méthodes de reproduction bien connues, le croisement génétique étant la plus aboutie et la plus adaptée à un monde aussi diversifié que la Terre. Le génome d'un individu est ainsi la matérialisation de l'information vitale transmise par ses ancêtres jusqu'à lui. Cet individu n'a, au départ, aucun autre moyen de transmettre de cette information que la reproduction. L'être vivant développe alors des sens et communique en principe. D'importantes adaptations survivalistes intra-générationnelles peuvent alors être effectuées. Une information dont un individu pouvant la communiquer de son vivant n'existe pas est perdue.

L'humanité se caractérise par sa capacité à transmettre une information indépendamment de ces deux canaux. Depuis Les murs des grottes jusqu'aux rayons cosmiques, la portée dans le temps et l'espace de la transmission volontaire d'information humaine est sans commune mesure avec celle de l'information biologique.

Elle a cependant peu de chance de s'étendre au-delà des limites du système solaire.

L'humanité devient alors la génitrice d'une nouvelle forme de vie, sa descendante héritant de tout son savoir, la vie artificielle.

Le concept de machine auto-répliquante a été introduit par John von Neumann en 1940 [?].

La nécessité ultime de la vie artificielle ayant été établie, il

replacants champs des possibles limité par ce que nous savons possible ne pas reinventer la roue nature pas de roue genetique ameliorée

A. Pourquoi un simulateur de marche ?

1) *La marche*: Comme tout nouveau-né, elle doit apprendre, notamment, à se déplacer.

Superviseur : Mathieu Defrance

Von Neumann, conscient des ressources minières terrestres limitées, a présenté le minage extraterrestre à grande échelle via vaisseau spatial auto-répliquants comme solution optimale.

Dans le cas de corps célestes trop massifs et dont l'atmosphère est trop peu dense, la rentabilité énergétique d'un objet de minage volant chargé de minerai devient à surface irrégulière médecine chaise roulante prothèse L'industrie du divertissement, en particulier celle du jeu vidéo, utilise déjà des simulateurs de marche pour créer des personnages plus réalistes de manière procédurale. La Science-Fiction est pleine de robot intelligents pouvant marcher ayant différentes fonctions. La proposition probablement la plus proche de nous dans le temps est celle présentée dans *"Westworld"*, où des parcs d'attraction peuplés de robots humanoïdes sont proposés aux touristes. Ces simulateurs grandeur nature pourraient être utilisés pour rentabiliser l'investissement dans la recherche et grandement accélérer son développement par l'introduction dans un environnement contrôlé de perturbations similaires à celles que les robots pourront rencontrer dans la réalité.

2) Le simulateur:

B. Pourquoi un algorithme génétique ?

Jhon Holland et son équipe, en 1975, [?]

L'algorithme génétique est un algorithme très puissant pour résoudre des problèmes d'optimisation combinatoire complexe. Sa nature stochastique lui permet de trouver des solutions souvent inattendues. Il est donc particulièrement adapté à la phase de recherche d'un projet.

II. ETAT DE L'ART

III. MÉTHODOLOGIE

A. Les hypothèses de base de notre approche

1) *Marche*: Tout d'abord, il est important de rappeler que la marche est un mouvement cyclique, c'est à dire qu'il se répète à intervalles réguliers. Il est donc possible de décrire un cycle de marche comme une séquence d'états, chacun étant une description de la position et de l'orientation du robot à un instant donné. Un pas est une chute contrôlée en avant, suivie d'une récupération de l'équilibre par la pose d'un membre au sol. La présence de plus de deux membres semble donc importante en début et fin de marche seulement. Elle n'est jamais indispensable si l'on se réfère à bon nombre d'animaux bipèdes et pourtant très stables et rapides. Une contrainte importante du projet étant la puissance de calcul de nos machines, et la notre nous permettant d'estimer que modifier des forces sur 2 pattes sera 2 fois plus rapide que sur 4, la marche bipède semble être un bon compromis.

2) *Paramètres:*

3) *Heuristique:* La composante de base la plus évidente pour l'agent est de se déplacer vite, nous récompensons donc la distance parcourue divisée par le temps mis pour la parcourir. La Cependant, il est important de noter que la vitesse n'est pas le seul paramètre important. En effet, un robot qui se déplace à une vitesse

4) *Algorithme génétique:*

B. Les fondements mathématiques

Genetic Algorithm

Require: Population size N

, Mutation rate p_m

, Crossover rate p_c

, Maximum number of generations G_{\max}

Ensure: Optimal solution

Initialize population P with N individuals

$g \leftarrow 0$

while $g < G_{\max}$ **do**

Evaluate fitness of each individual in P

Select parents for reproduction

Create empty offspring population Q

while $|Q| < N$ **do**

$p_1, p_2 \leftarrow \text{SelectParents}(P)$

$o_1, o_2 \leftarrow \text{Crossover}(p_1, p_2, p_c)$

$o_1 \leftarrow \text{Mutate}(o_1, p_m)$

$o_2 \leftarrow \text{Mutate}(o_2, p_m)$

Add o_1 and o_2 to Q

end while

$P \leftarrow Q$

$g \leftarrow g + 1$

end while

return Best individual in P

C. La méthode proposée

1) *Fitness Function:* Un élément clé d'un bon apprentissage avec algorithme génétique est la définition de la fitness function. C'est la formule qui va évaluer l'aptitude de chaque individu et leur attribuer plus ou moins de points selon les critères choisis. Notre fitness function est la suivante. Elle représente la valeur de la fitness pour un individu à un instant i .

Une créature est évaluée sur 5 critères :

- La distance entre le tronc et le sol alive_bonus
 - <0 Si le tronc touche le sol
 - >0 Sinon
- Les forces exercées : limite des mouvements trop violents pouvant provoquer la chute de la créature.
- La vitesse : distance parcourue divisée par le temps
- La hauteur maximale atteinte : maintient de la verticalité pour éviter une posture menant à la chute

Fitness = alive_bonus + (

$$H_{i-1} - H_i) * 0.1 + \frac{D_i - D_{i-1}}{time} + \text{forces} (1)$$

Ce calcul est effectué à chaque génération de la simulation sur chaque individu.

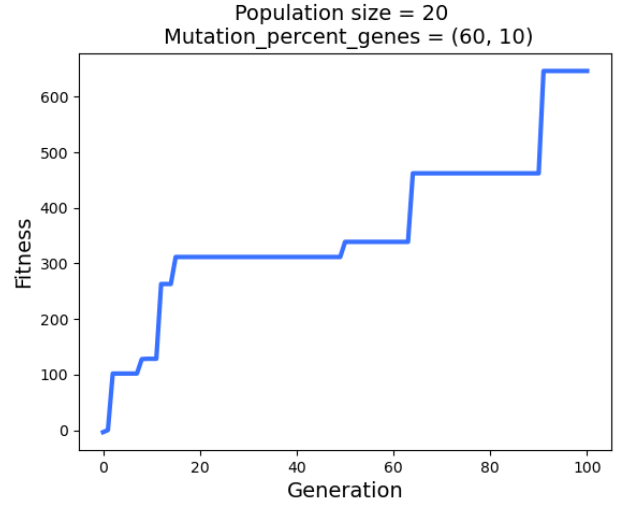


FIGURE 1. Example Image

La méthode que nous présentons ici a été développée sur base de choix d'implémentation grandement influencés par le contexte du projet. Celui-ci était séparé en 3 parties :

- 1. Recherche
- 2. Développement
- 3. Présentation d'une version vulgarisée du projet au Printemps des Sciences

Cette troisième partie rendait primordiale l'obtention d'un résultat fonctionnel et visuellement attrayant. Nous avons donc décidé d'utiliser la librairie PyGAD [?] pour nous libérer de l'écriture de l'algorithme lui-même et nous concentrer sur l'optimisations des paramètres.

D. Les jeux de données utilisés

E. Les instructions nécessaires pour pouvoir reproduire les expériences (par exemple pseudo-code)

Notre simulateur est disponible à l'adresse suivante : <https://github.com/nobourge/INFO-F308—Projets-d-informatique-3-transdisciplinaire—202223>

IV. RÉSULTATS

TABLE I
PARAMÈTRE PAR DÉFAUT DES SIMULATIONS

Parameter	Value
gene space	low : -1, high : 1, step : 0.1
init range low	-1
init range high	1
random mutation min val	-1
random mutation max val	1
initial population	None
population size	100
num generations	100
num parents mating	4
parallel processing	None
parent selection type	tournament
keep elitism	5
crossover type	uniform
mutation type	adaptive
mutation percent genes	(60, 10)

V. CONCLUSION

Notre simulateur permet de trouver en temps raisonnable des solutions permettant à des créatures de se maintenir debout sans chuter et se déplacer le temps de la simulation. Cependant, la qualité des solutions trouvées est très variable et la posture des créatures est toujours sous-optimale. Nous avons pu constater qu'il est très difficile de trouver les bons paramètres pour que l'algorithme converge vers une solution optimale. De plus, il est très difficile de trouver une fonction de fitness qui permette de bien évaluer les solutions.

L'algorithme génétique nous semble donc adapté à la pose de larges fondations.

1) *Limites*: Les limites de notre simulateur sont les suivantes :

- La fonction de fitness est très difficile à définir
- Les paramètres de l'algorithme génétique sont très difficiles à définir
- L'algorithme génétique est très lent

L'environnement physique est le plus simple possible : un sol plat. La moindre irrégularité introduite pourrait faire chuter la créature. Les forces à appliquer sur les articulations étant inscrites dans le génome, et la créature étant dépourvue de sens, aucune adaptation intra-générationnelle n'est possible.

A. Optimisations futures

B. Perspectives de recherche future

1) *Paramètres*: L'étendue des paramètres de PyGAD est très large et au vu de nos résultats obtenus après les tests présentés, une solution optimale est probablement réalisable en temps raisonnable via la bonne combinaison de paramètres. Pour la trouver, nous avons commencé la création d'une méta heuristique disponible très prochainement, l'idée est d'utiliser du scrapping pour récupérer les résultats de chaque utilisateur de PyGAD et de les comparer afin de trouver les meilleurs paramètres puis d'itérer sur cette base pour affiner les paramètres dont les valeurs sont des intervalles.

Un dictionnaire de génomes à éviter, tels qu'une suite de zéros entraînant l'absence d'action, et/ou assurément trop loin d'une solution viable

2) *Fonction de fitness*: **correction** d'un mauvais comportement, en plus de l'attribution de score, nous pourrions implémenter une fonction de correction pour identifier la région du génome commandant une action non désirée et la remplacer par des valeurs plus adaptées. La force de l'algorithme génétique étant sa souplesse, il est possible que des conditions plus laxistes permettent d'éviter de rester coincé dans un maximum local. L'objectif de hauteur du centre de masse est actuellement un maximum en un point et peut entraîner un déséquilibre limitant les possibilités de mouvement. Cet objectif pourrait être remplacé par un intervalle de hauteur global allouant plus de liberté à la créature.

La fonction de fitness pourrait elle-même être modifiée automatiquement en fonction des comportements observés

en liant les poids de chaque composante aux actions pour lesquelles elles entrent en jeu.

3) *Réseau neuronal*: Ces perspectives de développement de la fonction de fitness gén érationnelle mènent rapidement aux prémices d'un réseau neuronal. Lié à des capteurs, il permettrait à la créature de réagir à son environnement et de se déplacer de manière autonome

4) *Environnement*:

5) *Adaptation intra-générationnelle*:

6) *Parallélisation*:

7) *Autres domaines*: Le goulot d'étranglement de l'algorithme génétique est la somme des opérations sur le génome des individus, leur croisement et mutation. Dans un autre registre que le nôtre, la physique quantique ouvre le champ des possibilités d'action en permettant la superposition et l'intrication d'états. Ces propriétés pourraient être utilisées pour simuler l'équivalent de multiples créatures en "parallèle" et ainsi accélérer la convergence de l'algorithme génétique. [?]

RÉFÉRENCES

- [1] , John von Neumann *Theory of Self -Reproducing Automata*, 1966.
- [2] , Olivia Borgue and Andreas M . Hein *Near-Term Self-replicating Probes - A Concept Design*, 2005.
- [3] , John Holland *Adaptation in Natural and artificial Systems*, 1975.
- [4] , Gabriel Cormier *Systèmes Intelligents*, Université de Moncton, 2019.
- [5] , Karl Sims *Evolving Virtual Creatures*, 1994.
- [6] , Graham, Lee; Oppacher, Franz. *Speciation Through Selection and Drift*. Proceedings of The Eleventh IASTED International Conference on Artificial Intelligence and Soft Computing. ACTA Press.
- [7] , Josh C Bongard *The legion system : A novel approach to evolving heterogeneity for collective problem solving*.
- [8] , J. Bongard et H. Lipson "Simulation de la locomotion par algorithme génétique".
- [9] <https://pygad.readthedocs.io/en/latest/>
- [10] , Bart Rylander, Terence Soule, James Foster and Jim Alves-Foss *Quantum Genetic Algorithms*, 2000

ANNEXE A

CONSIGNES

Document

Le rapport doit être rédigé en L^AT_EX en utilisant ce template. La longueur du rapport ne devra pas, en tout cas, dépasser les 6 pages. Ce rapport doit être *self-contained*, c-à-d il doit pouvoir être lu et compris sans avoir besoin de se documenter ailleurs.