



Projet IA : Star Wars

Tatouine en danger

Ilies **GOURRI**
1^{ERE} ANNÉE CYCLE INGÉNIEUR

MAHER REBAI – MARS 2022

Réalisation d'un algorithme génétique

L'objectif de ce projet est de prédire la position d'un satellite qui suit une orbite de Lissajous, on a donc les coordonnées suivantes à un temps t donné grâce aux équations suivantes :

$$x(t) = p1 \times \sin(p2 \times t + p3)$$

$$y(t) = p4 \times \sin(p5 \times t + p6)$$

Pour entrainer notre algorithme on utilise une liste de valeur de x et y à un temps donné, puis grâce à un type d'algorithme évolutif : l'algorithme génétique.

PLAN D'ACTION

Le programme est constitué des principales fonction : fitness, mutation, sélection et croisement.

1. Quelle est la taille de l'espace de recherche (utiliser une notation scientifique) ?

L'espace de recherche pour chaque paramètre est $[-100; 100]$. L'espace de recherche pour les 6 paramètres sera donc le produit cartésien des intervalles que peut prendre chaque p : $[-100; 100]^6$.

On a avec l'ordinateur, une possibilité d'avoir 15 chiffres après la virgule, donc pour les 6 paramètres, la taille de l'espace de recherche sera de : $(200 \cdot 10^{15})^6 = 6.4 \cdot 10^{30}$

2. Quelle est votre fonction fitness ?

Individu

J'ai défini un individu par une liste de ses 6 paramètres, générés au hasard, compris entre -100 et 100. J'ai utilisé la fonction `uniform` de la bibliothèque `random` car il donne un chiffre de type `float`.

```
individu=[r.uniform(-100, 100),r.uniform(-100, 100),r.uniform(-100, 100),r.uniform(-100, 100),r.uniform(-100, 100),r.uniform(-100, 100)]
```

Fitness

Étant donné que nous pouvons directement calculer la valeur de x et y grâce à l'équation de l'orbite de Lissajous, il nous suffit de calculer ces valeurs et de calculer la différence avec les valeurs exactes à un temps donné. Je range ensuite ces 2 résultats dans un tuple trié selon le score.

```
def fitness(population,t,valExactX, valExactY):  
    resultat = []  
    for i in range(len(population)):  
        indiv = population[i]  
        score = 0  
        for j in range(len(t)):  
            x = indiv[0]*m.sin(indiv[1]*t[j]+indiv[2])  
            y = indiv[3]*m.sin(indiv[4]*t[j]+indiv[5])  
            score += (valExactX[j]-x)**2 + (valExactY[j]-y)**2  
        resultat.append((m.sqrt(score),population[i]))  
    return sorted(resultat)
```

PLAN D'ACTION

3. Décrivez les opérateurs mis en œuvre (mutation, croisement) ?

Mutation

Pour un individu on ajoute + ou - 1 à un paramètre pris au hasard. J'ai rajouté une condition if pour bien vérifier que les valeurs reste dans l'intervalle initiale -100 et 100.

```
def mutation(population):  
  
    pop = copy.deepcopy(population)  
  
    for i in range(len(pop)):  
  
        a = r.randint(0,5)  
        b = r.uniform(-1,1)  
  
        if population[i][a]<100 and population[i][a]>-100:  
  
            population[i][a]+=b  
  
    return pop
```

Croisement

Ici on sélectionne les 3 premiers paramètres (ceux de x) d'un individu 1 et les 3 derniers paramètres (ceux de y) d'un individu 2. On crée un nouveau individu issu de l'association de ces valeurs.

```
def croisement(population):  
  
    l=[]  
    #pop = copy.deepcopy(population)  
    for i in range(len(population)):  
  
        indiv1 = population[r.randint(0, len(population)-1)]  
        indiv2 = population[r.randint(0, len(population)-1)]  
        NEWindiv = []  
  
        for j in range(3):  
            NEWindiv.append(indiv1[j])  
        for t in range(3,6):  
            NEWindiv.append(indiv2[t])  
  
        l.append(NEWindiv)  
  
    return l
```

4. Décrivez votre processus de sélection.

La fonction fitness retourne un score et une population triée selon celui-ci. La fonction selection permet d'isoler la population sans le score, c'est lors de l'appel de cette fonction que l'on sélectionne les meilleurs de cette liste avec k.

```
def selection(fitPopulation,k=1):  
    l=[]  
    for i in range(k):  
        l.append(fitPopulation[i][1])  
    return l
```

5. Quel est la taille de votre population, combien de générations sont nécessaires avant de converger vers une solution stable?

La taille de ma population est de 10000 individu et je converge au bout de 100 générations environ vers une solution qui me satisfait. Je détaille à la suite de ces questions mon critère de satisfaction.

6. Combien de temps votre programme prend en moyenne (sur plusieurs runs) ?

Mon programme prend en moyenne 1min30 pour s'exécuter avec 10000 générations en paramètre. J'ai réalisé beaucoup de runs durant ces dernières semaines et ils avaient tous un résultat similaire.

7. Si vous avez testé différentes solutions qui ont moins bien fonctionnées, décrivez-les et discutez- les.

J'ai actuellement une population de 10000 individus créer, une sélection de 5%, une mutation 40% et un croisement de 40% et enfin on ajoute 15% d'individus d'une nouvelle population.

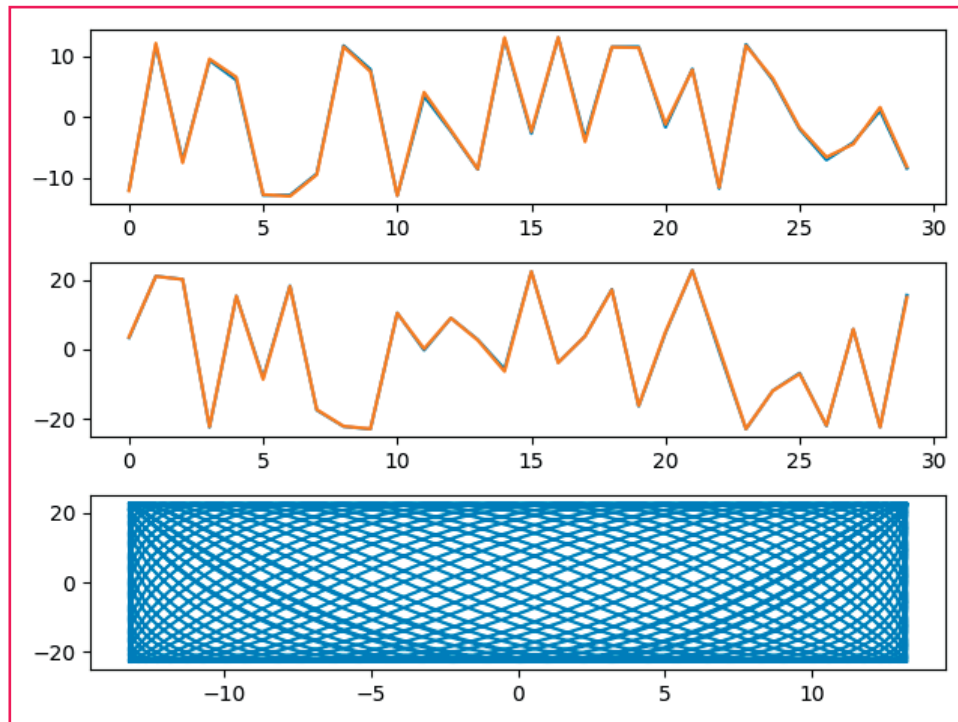
Au début, après une lecture de quelques documentations sur le sujet j'avais choisi une proportion de : 10% pour sélection, 80% croisement, 5% mutation et 5% nouvelle population. Mais j'ai réalisé que les proportions optimales varient selon le problème et que seul les tests et les modification après observation allaient m'aider à trouver les meilleurs résultats.

Vérification des points

Mon critère de satisfaction est : une distance euclidienne avec la position réel de 1.5

J'ai créer un autre code qui me permet de voir de manière plus concrète si mes valeurs sont bonnes. Sur la photo ci joint on peut voir sur le premier graphique les valeurs exactes que prend x en bleue et les valeurs que j'ai trouvé en orange par dessus, on voit bien que les valeurs sont proches. Même chose avec y juste en dessous.

Enfin j'ai affiché la trajectoire qui correspond bien à une orbite de Lissajous.



Conclusion

Avec plus de temps, j'aurais pu améliorer quelques aspects comme le temps d'exécution ou le nombre de générations nécessaire pour avoir un résultat satisfaisant.

Je suis malgré tout fier de mon travail, j'ai réussi à faire tout ce qui a été demandé, avec certains éléments qui me paraissaient complexes dans un premier temps, et qui se sont révélés comme un succès. Je ressors de ce projet avec un sentiment d'accomplissement et de nouvelles compétences en programmation.