

# **Pràctica II:**

# **Algoritmes Evolutius**

## **Annex**

**Professor:** Arnau Metaute Carrillo

**Estudiants:**

- Carla Abellana Segarra

1. Executeu els dos codis i apunteu, en format de taula, si s'arriba a una resposta, el fitness final al que arriba i les generacions que triga en arribar. Simuleu com a mínim 500 iteracions del món de Box2D i apunteu la mitja de 10 execucions de codi. Per escrit, destaqueu totes aquelles característiques o comportaments excèntrics.

A continuació es mostra una taula amb els resultats obtinguts després de 10 execucions del codi de l'algoritme genètic i més endavant del CMA-ES. A cada execució s'han simulat 1000 passos de Box2D, i s'ha registrat el fitness final i el nombre de generacions requerides per assolir-lo.

- Mitjana de generacions: 22.3
  - Mitjana de fitness: 14.7

Conclusions: Com es pot veure, 8 / 10 execucions s'ha assolit el màxim fitness, és a dir, que cap pilota toqués la zona vermella.

L'algoritme genètic implementat ha demostrat una capacitat elevada per resoldre el problema de forma eficient.

No obstant això, podem notar un comportament irregular i per tal de acabar de perfeccionar, es pot intentar ajustar paràmetres com el de la mutació, així evitem la pèrdua de solucions bones.

## CMA-ES

1r cop: millor fitness: 13 Generació: 30  Generació 26: Han caigut 3 boles Generació 27: Han caigut 2 boles Generació 28: Han caigut 3 boles Generació 29: Han caigut 3 boles Generació 30: Han caigut 2 boles  CMA-ES generació: 30 Pilotes en la zona vermelha: 2 Millor fitness: 13	2n cop: millor fitness: 13 Generació: 30  Generació 27: Han caigut 2 boles Generació 28: Han caigut 2 boles Generació 29: Han caigut 1 boles Generació 30: Han caigut 2 boles  CMA-ES generació: 30 Pilotes en la zona vermelha: 2 Millor fitness: 13	3r cop: millor fitness: 15 Generació: 22  Generació 19: Han caigut 2 boles Generació 20: Han caigut 1 boles Generació 21: Han caigut 1 boles Generació 22: Han caigut 0 boles solució optima!  CMA-ES generació: 22 Pilotes en la zona vermelha: 0 Millor fitness: 15	4t cop: millor fitness: 9 Generació: 30  Generació 25: Han caigut 11 boles Generació 26: Han caigut 9 boles Generació 27: Han caigut 9 boles Generació 28: Han caigut 9 boles Generació 29: Han caigut 8 boles Generació 30: Han caigut 6 boles  CMA-ES generació: 30 Pilotes en la zona vermelha: 6 Millor fitness: 9	5è cop: millor fitness: 15 Generació: 28  Generació 25: Han caigut 1 boles Generació 26: Han caigut 1 boles Generació 27: Han caigut 1 boles Generació 28: Han caigut 0 boles Generació 29: Han caigut 6 boles  CMA-ES generació: 28 Pilotes en la zona vermelha: 0 Millor fitness: 15
6è cop: millor fitness: 14 Generació: 30  Generació 26: Han caigut 3 boles Generació 27: Han caigut 1 boles Generació 28: Han caigut 2 boles Generació 29: Han caigut 2 boles Generació 30: Han caigut 1 boles  CMA-ES generació: 30 Pilotes en la zona vermelha: 1 Millor fitness: 14	7è cop: millor fitness: 12 Generació: 30  Generació 27: Han caigut 5 boles Generació 28: Han caigut 4 boles Generació 29: Han caigut 4 boles Generació 30: Han caigut 3 boles  CMA-ES generació: 30 Pilotes en la zona vermelha: 3 Millor fitness: 12	8è cop: millor fitness: 10 Generació: 30  Generació 27: Han caigut 4 boles Generació 28: Han caigut 7 boles Generació 29: Han caigut 5 boles Generació 30: Han caigut 5 boles  CMA-ES generació: 30 Pilotes en la zona vermelha: 5 Millor fitness: 10	9è cop: millor fitness: 15 Generació: 28  Generació 25: Han caigut 1 boles Generació 26: Han caigut 2 boles Generació 27: Han caigut 1 boles Generació 28: Han caigut 0 boles solució optima!  CMA-ES generació: 28 Pilotes en la zona vermelha: 0 Millor fitness: 15	Desena vegada: millor fitness: 15 Generació: 29  Generació 25: Han caigut 2 boles Generació 26: Han caigut 1 boles Generació 27: Han caigut 1 boles Generació 28: Han caigut 1 boles Generació 29: Han caigut 0 boles solució optima!  CMA-ES generació: 29 Pilotes en la zona vermelha: 0 Millor fitness: 15

- Mitjana de generacions: 26.75
- Mitjana de fitness: 13.1

Conclusions: Els resultats obtinguts mostren un comportament força irregular en comparació amb l'algoritme genètic. Concretament, només 4 de les 10 execucions han aconseguit la solució òptima, la qual cosa representa una taxa d'èxit del 40%, clarament la meitat que l'assolit per l'algoritme genètic.

Mentre executava el codi m'he trobat un parell de cops amb un error en la condició d'aturada de l'algoritme, on es considera que s'ha assolit el valor òptim quan en realitat el càlcul del fitness no reflecteix la posició real de les boles.

2. Quin mètode s'ha escollit a la hora de crear la mating pool?  
Mencioneu mínim una alternativa i destaqueu les diferències, tant a nivell general com en aquest cas concret, i que us han fet decantar per al mètode utilitzat finalment.

El mètode seleccionat ha sigut la **selecció elitista + selecció per torneig**.

El que tracta aquest mètode i per la raó que el vaig escollir es perquè primer s'inclouen els 5 millors individus de cada generació, això tracta l'elitisme. La resta de la població es completa seleccionant guanyadors d'un torneig entre 3 individus escollits aleatoriament de la població actual.

El que podria ser una alternativa d'aquest mètode es la selecció per ruleta, on cada individu té una probabilitat de ser seleccionat proporcional al seu fitness. És a dir, els individus amb un valor de fitness més alt tenen més probabilitat de ser seleccionats.

La raó per la qual he escollit aquest mètode per la mating pool és perquè evita que un sol individu domini massa aviat, preservant la diversitat genètica entre els altres.

### 3. Quin mètode de crossover s'ha escollit? Dueu a terme la mateixa comparació.

El crossover que he utilitzat ha sigut el **one-point crossover**. El que fa és que es selecciona un punt aleatori entre els gens de l'individu i es creuen les parts de dos progenitors per crear dos descendents

```
def mate(parent1, parent2):
    crossover_point = random.randint(1, NUM_RECTANGLES - 1)
    child1_angles = parent1.angles[:crossover_point] + parent2.angles[crossover_point:]
    child2_angles = parent2.angles[:crossover_point] + parent1.angles[crossover_point:]
    return Individual(angles=child1_angles), Individual(angles=child2_angles)
```

Com una alternativa, podria haver utilitzat el *crossover uniforme*, on cada gen s'intercanvia de manera independent amb una probabilitat del 50%. Aquest mètode trenca més fàcilment patrons útils, cosa que pot influir en la convergència.

Finalment, he optat per one point crossover ja que aquesta estratègia assegura que cada fill manté una part coherent dels angles del primer progenitor i l'altra del segon. Això, significa que certes configuracions de rectangles que han demostrat ser útils es poden preservar parcialment, mentre que es combinen amb nous patrons d'altres individus.

4. Com afecta el valor de mutació al resultat final? Creeu una taula indicant un mínim de 5 valors diferents i apunteu les mateixes dades que l'exercici 1. Perquè es succeeixen aquestes diferències entre els valors?

**TAXA MUTACIÓ = 0.65**

FITNESS AG	GENERACIONS
15	22
14	23
15	22
15	18
15	16
15	18
15	17
15	18
15	18
15	20

**TAXA MUTACIÓ = 0.02**

FITNESS AG	GENERACIONS
13	22
15	25
13	22
14	30
13	30
14	30
14	30
13	30
14	30
14	29

**TAXA MUTACIÓ = 0.20**

FITNESS AG	GENERACIONS
15	25
15	22
15	22
14	30
15	17
14	30
15	20
15	23
14	30
15	23

**TAXA MUTACIÓ = 0.05**

FITNESS AG	GENERACIONS
15	23
15	25
15	23
14	30
14	30
15	24
15	23
15	26
14	30
15	23

**TAXA MUTACIÓ = 0. 0.000002**

FITNESS AG	GENERACIONS
13	30
13	30
15	30
14	30
13	30
13	30
15	29
13	30
14	30
13	30

He realitzat les comparacions amb l'algoritme genètic, amb valors bastant contrastats entre ells, i els resultats que he obtingut han sigut els següents:

- Valors molt alts (0.65): El sistema aplica moltes mutacions, generant una gran varietat de solucions a cada generació. Això pot destruir estructures bones que s'havien format, fent que el comportament tragi més a estabilitzar-se.
- Valors mitjans (0.20): Es veu el millor rendiment. La mutació encara introduceix diversitat, però no massa com per desestabilitzar el sistema. Aquí l'algoritme té capacitat d'explorar noves solucions i, alhora, mantenir les bones.
- Valors molt baixos: Gairebé no hi ha canvis entre generacions. Això impedeix explorar nous espais de solució, per tant, l'algoritme es queda estancat en una solució baixa.

5. En el CMA-ES trobem el mètode de mutació? Perquè es deu aquest fet i quines millores porta sobre els Algoritmes Genètics?

En el cas del CMA-ES, no s'utilitza una mutació explícita com en els algorismes genètics clàssics. En lloc d'això, el procés de generació de noves solucions es basa en una distribució normal multivariant que s'ajusta automàticament a mesura que l'algoritme evoluciona. Aquest ajust es fa mitjançant una matriu de covariància que s'adapta segons els resultats obtinguts en iteracions anteriors.

La gran avantatge del CMA-ES és que no fa servir mutacions a l'atzar, sinó que "aprèn" de les direccions de cerca més prometedores. Si detecta que les millors solucions tendeixen a seguir un cert patró, adapta la seva estratègia per seguir-lo. Si, pel contrari, es detecta una direcció que porta a resultats dolents, el sistema evita seguir per aquest camí.

En definitiva, el CMA-ES ajusta la seva manera de cercar noves solucions d'una manera intel·ligent, tenint en compte tant els encerts com els errors del passat.

6. Quin mètode és comunament utilitzat durant l'execució d'un CMA-ES per a evitar els mínims locals? Quina lògica té darrere?

Durant l'execució d'un CMA-ES, un dels mètodes més comuns per evitar quedar atrapat en mínims locals és l'ús de la **matriu de covariància adaptativa**. Aquest mecanisme permet que la distribució de cerca s'estiri i es desplaci cap a les direccions que han produït millors resultats en el passat.

La lògica darrere d'aquest sistema és aprendre del comportament històric de les solucions, adaptant la forma i l'orientació de la "campana" de la distribució normal que genera noves solucions. Així, si el problema presenta una vall estreta o una forma allargada, el CMA-ES pot adaptar-se per continuar explorant en la direcció correcta i no quedar-se bloquejat.

7. És important la matriu de covariàncies inicial? Creeu una taula indicant un mínim de 3 valors diferents i apunteu les mateixes dades que l'exercici 1. Es produueixen canvis respecte l'original? Perquè?

He variat a la variable *initial\_sigma* els valors; 0.2, 1, 3

```
def cmaes_algorithm():
    initial_solution = np.random.uniform(-np.pi, np.pi, NUM_RECTANGLES)
    initial_sigma = 1.0

    opts = cma.CMAOptions()
    opts.set("bounds", [-np.pi, np.pi])
    opts.set("popsize", POPULATION_SIZE)
    opts.set("verbose", -1)
```

La taula queda de la següent manera:

Valor $\sigma$	Mitja Fitness CMA	Mitja Generacions CMA
0.2	0	30
1	14.6	23.5
3	12.4	30

Segons els diferents valors de sigma podem conoure la següent informació:

- $\sigma = 0.2$ : Aquest valor molt petit fa que les noves solucions generades estiguin molt a prop les unes de les altres. Això s'ha traduït en un comportament molt estable: en totes les 10 execucions, el CMA-ES ha aconseguit un fitness de 0. No obstant això, mai ha convergit abans de 30 iteracions, fet que demostra una evolució molt lenta.
- $\sigma = 1.0$ : Aquest valor es com el tenia posat. Segons la mitja i les generacions obtingudes indica un bon equilibri entre exploració i explotació.
- $\sigma = 3.0$ : Amb aquest valor elevat, el CMA-ES explora molt més a l'atzar. S'observa que ha trigat les 30 iteracions completes. I els resultats no han sigut òptims, això indica que, tot i explorar molt, no ha estat capaç de refinjar tant com altres valors.

Clarament es produueixen canvis significatius respecte l'original, tant en la qualitat del fitness com en la velocitat de convergència.

Això passa perquè la matriu de covariàncies inicial controla l'amplada amb que es generen noves solucions. I si variem aquest valor tant a molt petit com a molt gran, veurem canvis significatius.

8. Quin és el nombre de població ideal? Repetiu les execucions amb 50 i 200 individus, recolliu les dades en una taula com l'exercici 1 i comenteu els avantatges i inconvenients que té crear més o menys individus del recomanat.

### 50 individus

FITNESS AG	FITNESS CMA-ES
15	11
14	12
15	10
13	11
14	9
15	10
14	11
15	12
13	10
15	11

- **Mitjana AG:** 14.3
- **Mitjana CMA-ES:** 10.7

### 200 individus

FITNESS AG	FITNESS CMA-ES
15	13
14	14
15	15
14	12
15	14
14	15
15	13
15	15
14	14
15	13

- **Mitjana AG:** 14.7
- **Mitjana CMA-ES:** 13.8

Un cop veient els resultats, en una població petita, com la de 50 individus, el principal avantatge és la rapidesa de càlcul, el qual va molt bé quan disposes de pocs recursos computacionals. Tanmateix, aquest tipus de població pot presentar desavantatges importants com que tendeix a convergir ràpidament

cap a una solució que pot no ser òptima, i sovint pateix una manca de variabilitat genètica, cosa que redueix la capacitat d'explorar noves solucions.

En canvi, una població gran, com la de 200 individus, ofereix una diversitat molt més gran dins la població, la qual cosa permet una exploració més àmplia de l'espai de solucions. Tot i això, el cost computacional augmenta significativament i això pot ser una desavantatge.

9. Són les característiques de l'individu òptimes? Executeu el codi de nou on cada individu estigui format per 15 i 25 rectangles, recolliu les dades en una taula com l'exercici 1 i comenteu i discutiu els canvis de comportament dels individus.

### 15 rectangles

FITNESS AG	FITNESS CMA-ES
12	11
12	11
11	10
10	11
11	9
12	10
13	11
12	11
11	10
11	11

- Mitjana AG:** 11.5
- Mitjana CMA-ES:** 10.5

### 25 rectangles

FITNESS AG	FITNESS CMA-ES
15	15
14	15
15	15
15	15
15	14
15	15
15	15
15	15
15	14
15	15

- Mitjana AG:** 14.9

- **Mitjana CMA-ES:** 14.8

Després de fer el mateix procediment que en l'exercici 1 amb aquesta variació dels números de rectangles, podem veure que l'augment o disminució del nombre de rectangles impacta clarament en el rendiment dels algoritmes evolutius.

Quan el nombre de rectangles és 15, els individus tenen menys flexibilitat geomètrica per fer l'estructura de manera òptima, per tant, obtenim un fitness més baix i no aconseguim un fitness de 15.

En canvi, augmentar el nombre de rectangles a 25 proporciona més graus de llibertat i capacitat d'adaptació als individus. L'AG, amb 25 rectangles, també millora respecte a la versió amb 15, però triga més generacions a convergir degut al major espai de cerca.