الغاية من الجلسة:

تطبيق خوارزمية النحل من أجل إيجاد أفضل قيمة للتابع x^2

مقدمة:

في خوارزمية النحل يتم محاكاة عمل النحل في البحث عن مصدر الطعام يتم استخدام ثلاث أنواع من النحل

النحل الموظف والمراقب والمستكشف لكل منهم مهمة محددة فالنحل الموظف يقوم بالبحث عن مصادر الطعام والنحل المراقب يقوم بمراقبة عمل أفضل النحلات الموظفة والمستكشفة تقوم بالبحث بشكل عشوائي عن مصادر طعام جديدة

في هذه الجلسة سنقوم بتطبيق عدة عمليات وهي:

- توليد نحلات ضمن مجال محدد
- تطبيق تابع select لاختيار أفضل نحلة بالاعتماد على تابع fitness
- إنشاء تابع لتمثيل خوارزمية النحل

```
n_bees = 30
n_iterations = 100
Upper_band = 10
Lower_band=-10
Best_bees = 10
scout_bees = 5

def fitness_function(x):
return x ** 2
def generate_random_bee(Upper_band,Lower_band):
x = random.uniform(Upper_band,Lower_band)
fitness = fitness_function(x)

return {'x': x, 'fitness': fitness}
```

تابع يقوم باختيار أفضل النحلات الموظفة

```
def select_best_bees(bees,Best_bees):

bees_sorted = sorted(bees, key=lambda bee: bee['fitness'])
return bees_sorted[:Best_bees]
```

تابع الانتقال الى مصدر طعام

```
def move_bee(bee, best_bees,Upper_band,Lower_Band):
best_bee = random.choice(best_bees)
step_size = random.uniform(0.5, 1.5)
new_x = best_bee['x'] + step_size * (bee['x'] - best_bee['x'])
new_x = max(Lower_band, min(new_x, Upper_band))
return {'x': new_x, 'fitness': fitness_function(new_x)}

def bee_algorithm():
bees = [generate_random_bee(Upper_band,Lower_band) for _ in range(n_bees)]
for _ in range(n_iterations):
best_bees = select_best_bees(bees)
for bee in bees:
if random.random() < 0.5:
bee = move_bee(bee, best_bees,Upper_band,Lower_band)
bees += [generate_random_bee(Upper_band,Lower_band) for _ in
range(scout_bees)]
bees = select_best_bees(bees,30)
best_bees = select_best_bees(bees)
return best_bees[0]
best_solution = bee_algorithm()
```

print(f"Best solution: x = {best_solution['x']},Fitness value =
{best_solution['fitness']}")

Best solution: x = -0.641148115219135,Fitness value = 0.41107090564904913

I=[1,2]

I+=[2,4]