ANA670 Applied Optimization

Final Exam

First Name	**Randall**
Last Name	**Crawford**
ID#	**901012194**
Email Address	**rcc_0149@yahoo.com**

Table of Contents

How to submit your Assignment

After filling all the parts in this file, please follow the following steps.

1) Add your name and ID to the first page.
2) Save the file as a PDF file
3) Rename the file as

   YOUR ***First*** *Name* - *YOUR* ***Last*** *Name* - *ID* – ANA670 - ***FinalExam****.pdf*

   Example: John - Smith - ID – ANA670 - FinalExam.pdf

4) Upload the file and submit it (only using BrightSpace)

P1 [60 points]

Describe the steps carried out in an iteration of the Genetic Algorithm in your own words. Include the purpose and effect of the Selection, Crossover and Mutation steps. Why would we use it instead of, say, Gradient Descent? (hint: it is not always possible to use analytical algorithms – why?) (75-150 words)

Solution

A Genetic Algorithm (GA) iteration follows a structured process inspired by biological evolution. It begins with encoding the optimization function into binary or real-valued arrays as chromosomes. A fitness function (e.g., F = A - f(x)) is defined to evaluate individuals. The process starts by initializing a population, followed by assessing each individual's fitness.

A new population is then created through *Selection*, where fitter individuals (based on fitness-proportion reproduction) are chosen to reproduce, ensuring better traits are passed on. This enhances the overall quality of the population. *Crossover*, with a high probability ($p_c \approx 0.7$-1.0), swaps segments between parent chromosomes (e.g., single-point or multi-point crossover), combining genetic information from two selected individuals to create offspring, promoting diversity and exploring new solutions. *Mutation*, with a low probability ($p_m \approx 0.001$-0.05), randomly flips bits to introduce variation and avoid local optima, adding further diversity. This cycle repeats until stopping criteria are met, and then results are decoded to obtain the solution to the problem.

We would use Genetic Algorithm (GA) instead of methods like Gradient Descent (GD), because GD relies on differentiable, continuous functions, and gradient information, which isn't always available. GA excels in complex, noisy, or discrete problems where analytical gradients are impractical or nonexistent. GD can get trapped in local minima, especially in non-convex landscapes, while GA's population-based approach explores multiple solutions simultaneously, leveraging crossover and mutation to escape such traps. Additionally, GA handles multi-objective optimization and non-differentiable constraints effectively, making it more versatile despite requiring more computational resources.

P2 [70 points]

Seeking the maximum of the following Objective Function, carry out 3 iterations of the PSO algorithm, showing detailed calculations (each step)

$$f(x) = -x^2 + 2x + 11$$

within in the range $-2 \leq x \leq 2$ using the PSO method. Use 4 particles ($n = 4$) with the initial positions x1 = −1.5, x2 = 0.0, x3 = 0.5, and x4 = 1.25.

Fill in your answers below. Pay close attention to & answer the prompts in the box below.

Solution

PSO Method

PSO is an optimization technique where particles move through the search space, adjusting their positions based on their own best-known position (personal best, pBest) and the swarm's best-known position (global best, gBest). The position update for each particle is given by:

$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot r_1(pBest_i - x_i(t)) + c_2 \cdot r_2(gBest - x_i(t))$
$x_i(t + 1) = x_i(t) + v_i(t + 1)$

where:

$v_i(t)$: velocity of particle i at iteration t
$x_i(t)$: position of particle i at iteration t
w: inertia weight (assume w = 0.7)
c_1, c_2: cognitive and social learning factors (assume $c_1 = c_2 = 2$)
r_1, r_2: random numbers between 0 and 1 (vary simulation for each iteration)
$pBest_i$: personal best position of particle i
gBest: global best position across all particles

v_{max}: Use a 0.5 threshold to help keep particles from getting stuck at boundaries (e.g., x = 2) and allow controlled movement to the optimum.

Since initial velocities aren't given, assume all initial velocities for each particle equal 0.1 for simplicity.

Initial Evaluation

First, evaluate the objective function $f(x) = -x^2 + 2x + 11$ at the initial positions:

$f(x_1) = f(-1.5) = -(-1.5)^2 + 2(-1.5) + 11 = -2.25 - 3 + 11 = 5.75$
$f(x_2) = f(0.0) = -(0)^2 + 2(0) + 11 = 11$
$f(x_3) = f(0.5) = -(0.5)^2 + 2(0.5) + 11 = -0.25 + 1 + 11 = 11.75$
$f(x_4) = f(1.25) = -(1.25)^2 + 2(1.25) + 11 = -1.5625 + 2.5 + 11 = 11.9375$

Initial pBest values are the same as the initial f(x) values:

$pBest_1 = 5.75$, $pBest_2 = 11$, $pBest_3 = 11.75$, $pBest_4 = 11.9375$
gBest = 11.9375 (from particle 4, the highest value)

I) **Detailed calculations, 3 iterations:**

Iteration 1

Random numbers: $r_1 = [0.3, 0.4, 0.5, 0.6]$, $r_2 = [0.7, 0.8, 0.9, 0.2]$

Particle 1: $v_1' = 0.7(0.1) + 2 \cdot 0.3(5.75 - (-1.5)) + 2 \cdot 0.7(11.9375 - (-1.5)) = 0.07 + 4.35 + 18.8125 = 23.2325$ (clamp to 0.5)

$x_1' = -1.5 + 0.5 = -1.0$
$f(-1.0) = -(-1.0)^2 + 2(-1.0) + 11 = -1 - 2 + 11 = 8$
$pBest_1 = 8$

Particle 2: $v_2' = 0.7(0.1) + 2 \cdot 0.4(11 - 0) + 2 \cdot 0.8(11.9375 - 0) = 0.07 + 8.8 + 19.1 = 27.97$ (clamp to 0.5)

$x_2' = 0 + 0.5 = 0.5$
$f(0.5) = -(0.5)^2 + 2(0.5) + 11 = -0.25 + 1 + 11 = 11.75$
$pBest_2 = 11.75$

Particle 3: $v_3' = 0.7(0.1) + 2 \cdot 0.5(11.75 - 0.5) + 2 \cdot 0.9(11.9375 - 0.5) = 0.07 + 11.25 + 20.5875 = 31.9075$ (clamp to 0.5)

$x_3' = 0.5 + 0.5 = 1.0$
$f(1.0) = -(1.0)^2 + 2(1.0) + 11 = -1 + 2 + 11 = 12$
$pBest_3 = 12$

- **Particle 4: $v_4' = 0.7(0.1) + 2 \cdot 0.6(11.9375 - 1.25) + 2 \cdot 0.2(11.9375 - 1.25) = 0.07 + 12.825 + 4.275 = 17.7$ (clamp to 0.5)**

$x_4' = 1.25 + 0.5 = 1.75$
$f(1.75) = -(1.75)^2 + 2(1.75) + 11 = -3.0625 + 3.5 + 11 = 11.4375$
$pBest_4 = 11.4375$

$gBest = 12$ (from $x_3' = 1.0$)

Iteration 2

Random numbers: $r_1 = [0.2, 0.3, 0.4, 0.5]$, $r_2 = [0.6, 0.7, 0.8, 0.9]$

Particle 1: $v_1' = 0.7(0.5) + 2 \cdot 0.2(8 - (-1.0)) + 2 \cdot 0.6(12 - (-1.0)) = 0.35 + 3.6 + 15.6 = 19.55$ (clamp to 0.5)

$x_1' = -1.0 + 0.5 = -0.5$
$f(-0.5) = -(-0.5)^2 + 2(-0.5) + 11 = -0.25 - 1 + 11 = 10.75$
$pBest_1 = 10.75$

Particle 2: $v_2' = 0.7(0.5) + 2 \cdot 0.3(11.75 - 0.5) + 2 \cdot 0.7(12 - 0.5) = 0.35 + 6.75 + 16.1 = 23.2$ (clamp to 0.5)

$x_2' = 0.5 + 0.5 = 1.0$
$f(1.0) = -(1.0)^2 + 2(1.0) + 11 = -1 + 2 + 11 = 12$
$pBest_2 = 12$

Particle 3: $v_3' = 0.7(0.5) + 2 \cdot 0.4(12 - 1.0) + 2 \cdot 0.8(12 - 1.0) = 0.35 + 8.8 + 8.8 = 17.95$ (clamp to 0.5)

$x_3' = 1.0 + 0.5 = 1.5$
$f(1.5) = -(1.5)^2 + 2(1.5) + 11 = -2.25 + 3 + 11 = 11.75$
$pBest_3 = 11.75$

Particle 4: $v_4' = 0.7(0.5) + 2 \cdot 0.5(11.4375 - 1.75) + 2 \cdot 0.9(12 - 1.75) = 0.35 + 9.6875 + 18.45 = 28.4875$ (clamp to 0.5)

$x_4' = 1.75 + 0.5 = 2.25$ (clamp to 2)
$f(2.0) = -(2.0)^2 + 2(2.0) + 11 = -4 + 4 + 11 = 11$
$pBest_4 = 11$

gBest = 12 (from $x_2' = 1.0$)

Iteration 3

Random numbers: $r_1 = [0.1, 0.2, 0.3, 0.4]$, $r_2 = [0.5, 0.6, 0.7, 0.8]$

Particle 1: $v_1' = 0.7(0.5) + 2 \cdot 0.1(10.75 - (-0.5)) + 2 \cdot 0.5(12 - (-0.5)) = 0.35 + 2.25 + 12.5 = 15.1$ (clamp to 0.5)

$x_1' = -0.5 + 0.5 = 0.0$
$f(0.0) = -(0)^2 + 2(0) + 11 = 11$
$pBest_1 = 11$

Particle 2: $v_2' = 0.7(0.5) + 2 \cdot 0.2(12 - 1.0) + 2 \cdot 0.6(12 - 1.0) = 0.35 + 4.4 + 13.2 = 17.95$ (clamp to 0.5)

$x_2' = 1.0 + 0.5 = 1.5$
$f(1.5) = -(1.5)^2 + 2(1.5) + 11 = -2.25 + 3 + 11 = 11.75$
$pBest_2 = 11.75$

Particle 3: $v_3' = 0.7(0.5) + 2 \cdot 0.3(11.75 - 1.5) + 2 \cdot 0.7(12 - 1.5) = 0.35 + 6.15 + 14.7 = 21.2$ (clamp to 0.5)

$x_3' = 1.5 + 0.5 = 2.0$
$f(2.0) = -(2.0)^2 + 2(2.0) + 11 = -4 + 4 + 11 = 11$
$pBest_3 = 11$

Particle 4: $v_4' = 0.7(0.5) + 2 \cdot 0.4(11 - 2) + 2 \cdot 0.8(12 - 2) = 0.35 + 7.2 + 16 = 23.55$ (clamp to 0.5)

$x_4' = 2 + 0.5 = 2.5$ (clamp to 2)
$f(2.0) = -(2.0)^2 + 2(2.0) + 11 = -4 + 4 + 11 = 11$
$pBest_4 = 11$

gBest = 11.75 (from $x_2' = 1.5$)

II) Value of the solution found so far in the iterations (from the last iteration above)

1st Iteration: gBest = 12 (from $x_3' = 1.0$)
2nd Iteration: gBest = 12 (from $x_2' = 1.0$)
3rd Iteration: gBest = 11.75 (from $x_2' = 1.5$)

III) Did it get close to the correct solution? Why or why not, do you think?

Yes, with x = 1, f(1) = $-(1)^2 + 2(1) + 11 = -1 + 2 + 11 = 12$ appears to be the maximum. This was found on the 1st iteration, indicating a successful convergence with chosen parameters. Believe the use of the vmax = 0.5 threshold really helped the process converge, and making sure to sufficiently vary random numbers in the iterations helped cover sufficient ground to recognize the maximum. Believe my static choices for inertia weight, as well as cognitive and social learning factors were reasonably sound. Imagine the more one would get to use this method, the better understanding they would gain with parameter setting.

P3 [70 points]

About Ant Colony Optimization, explain in your own words specifically the what the following two formulas in the textbook do in Ant Colony Optimization:

I) Pheromone updating **formula 13.3** and
II) Route choice probability **formula 13.1** (which depends on it)

(See our textbook "Engineering Optimization" in Course Resources on the course site.)

Solution

I) Pheromone Updating Formula 13.3

$$\phi_{ij}^{t+1} = (1 - \gamma)\phi_{ij}^{t} + \delta\phi_{ij}^{t}$$

This formula is used to update the amount of pheromone on a path (or edge) between two points i and j in the next time step (t + 1). In Ant Colony Optimization, which mimics how ants find the shortest path to food by leaving pheromone trails, this update helps the system "remember" and adjust the desirability of different routes over time.

Explanation:

ϕ_{ij}^{t} **: This is the pheromone level on the path from i to j at the current time t.**

$(1 - \gamma)$ **: This part represents the evaporation of pheromone over time, where γ (a value between 0 and 1) is the evaporation rate. Evaporation prevents the pheromone trails from getting too strong and allows the ants to explore new paths, avoiding getting stuck on a suboptimal route.**

$\delta\phi_{ij}^{t}$ **: This is the additional pheromone deposited, which depends on the quality of the path (e.g., how short or efficient it was). The better the path, the more pheromone is added.**

When combined, this formula balances forgetting old information (evaporation) with adding new information (deposition) to guide future ant choices.

II) Route Choice Probability Formula 13.1

$$p_{ij} = \phi_{ij}{}^{\alpha} d_{ij}{}^{\beta} \,/\, \sum_{i,j=1}^{n_d} \phi_{ij}{}^{\alpha} d_{ij}{}^{\beta}$$

This formula calculates the probability that an ant will choose to move from point i to point j based on the pheromone levels and other factors like distance. It is a key part of deciding which paths ants are more likely to take, influencing the optimization process.

Explanation:

$\phi_{ij}{}^{\alpha}$ **: This is the pheromone level on the path from i to j, raised to the power α, which controls how much the ants rely on pheromone information. A higher α means they trust the pheromone trails more.**

$d_{ij}{}^{\beta}$ **: This represents a desirability factor (often the inverse of distance or some cost), raised to the power β, which controls how much ants consider this factor. A higher β makes distance (or cost) more important.**

$\sum_{i,j=1}^{n_d} \phi_{ij}{}^{\alpha} d_{ij}{}^{\beta}$ **: This is the sum of the numerator over all possible paths, normalizing the probability so it adds up to 1. n_d is the number of possible destinations.**

The result p_{ij} is the probability of choosing the path from i to j, combining both pheromone trails (past experience) and heuristic information (e.g., distance) to guide the ants toward better solutions.

How They Work Together in Ant Colony Optimization

The pheromone updating formula (13.3) adjusts the pheromone levels after each iteration based on the ants' experiences, reinforcing good paths and weakening unused ones.

The route choice probability formula (13.1) uses the updated pheromone levels (along with distance or other heuristics) to decide where ants should go next. This creates a feedback loop: good paths get more pheromone, increasing their probability of being chosen, which leads to more pheromone, and so on, helping the colony converge on an optimal solution (like the shortest path).

This process mimics how real ants find efficient routes and can be used in optimization problems to solve complex tasks, such as finding the best route in a network or scheduling.

The end