NIC

Gravitational Search Algorithm (GSA)

1. Abdul Baseer 2.Hamza Bilal
3. Daim Raza 4.Mahad Mustafa

Topics Covered in Presentation

- What is Force.
- What is Gravity
- What is Gravitational Force
- What is Gravitational Search Algorithm
- Gravitational Search Algorithm Step-by-step Explanation
- How to Initialize Population Randomly
- How to calculate velocity, Force, Mass and Acceleration
- How to Update Values for each search agent.

From where it come

Gravitational Search Algorithm is Implemented by Newton's Gravitational Law

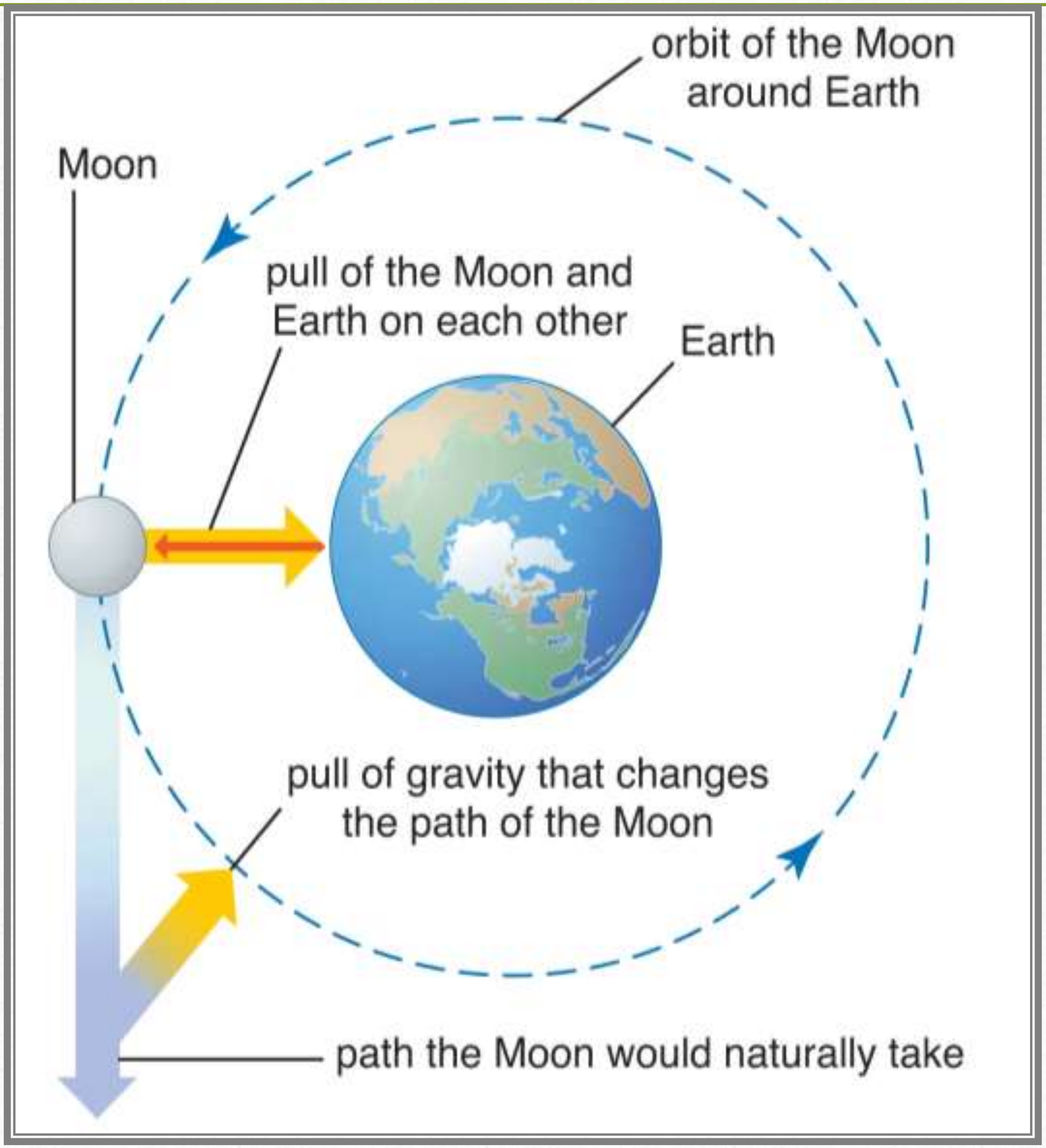
This is new Optimization Algorithm based on Law of Gravity and Mass.

What is Force

1. Force = Push or Pull of an Object.

What is GRAVITY

- Gravity is Force that pull all elements of matter together.
- On planet Earth Gravity Gives weight to physical bodies/object.
- In the outer space there is no Gravity [only small amount of gravity can be found]
- **Weight = 0**
- If there is no Gravity



What is Gravitational Force

Force Acting on the particle surface

$$F = mg$$

Here ,

m = Mass

g = Gravitational Force

G = 6.67 * 10^{-11} $NM^2 KG^{-2}$

According to Gravitational Force, Every Particle in the Universe attract every other particle having mass.

Gravitational Force is Attractive force between Two Masses M1 and M2 separated by Distance r.
Gravitational Force is Given By:

$$F = -G\left(\frac{M_1 M_2}{r^2}\right)$$

Here, G = Universal Gravitational Constant
F = Force
r = Distance between center of masses.
M_1 = Mass of Object one.
M_2 = Mass of Object two.

Gravitational Search Algorithm

- Gravitational Central Force [Attractive in Nature]
- Work Done by it is Independent of the path followed. [Independent of the presence of other bodies around].
- It is Conservative force.
- It is Weakest force in Nature. [1038 smaller than Nuclear Force].
- Force that cause apple fall downward from tree.
- Force that keep you walking on Earth instead of floating away into space.

Gravitational Search Algorithm

- In this algorithms Agents are Considered as Objects.
- Objects performance is measured by their Masses [Heavy Masses means Good Solution].
- Gravitational force is used by objects to attract each other.
- Gravitational force cause movement of objects towards Havier masses.

Exploitation Step in GSA Algorithm: When Object / Agent with Heavy Mass moves Slowly

Heavy Masses = Better Solutions


Who
What
When
Where
Why
How

Where GSA Used

Gravitational Search Algorithm is Implemented to Solve Complex Optimization Problems.

How is GSA Constructed?

By Law of Gravity.

How Mass is Determined?

Fitness Function is used for Mass Determination.

What is Agent in GSA?

- ✓ Consider Objects as Agents
- ✓ Solution for Optimization Problems
- ✓ Mass and Current Position for each agent is Known.

When Gravitational Search Algorithm will Stop?

- ✓ During Generations / Iterations when all agents/object will gather on same point

How to Update Agent Position.

- ✓ **Obtain New Positions**

Parameters OF GSA

- ✓ Mass
- ✓ Position
- ✓ Active and Passive Gravitational Masses

Mass Calculation

$$\text{Mass} = \frac{Fitness(i) - Worst(t)}{Best(t) - Worst(t)}$$

For Minimization: Consider Minimum Fitness Value **BEST** & Maximum Fitness Value **Worst**.

For Maximization: Consider Minimum Fitness Value **WORST** & Maximum Fitness Value **BEST**.

Force Calculation

The force Between ith and jth Agent

Gravitational Force is Attractive force between Two Masses M1 and M2 separated by Distance r. Gravitational Force is Given By:

$$\boxed{F_{ij}^{d}(t) = G(t)\left(\frac{M_{pi}(t) * M_{aj}(t)}{R_{ij}(t) + \varepsilon}\right)(X_j^d(t) - X_i^d(t))}$$

M_{pi} = Passive Gravitational Mass (for agent i)

M_{ai} = Active Gravitational Mass (for agent j)

R = Euclidean Distance between 2 agents.

$$\boxed{F = -G\left(\frac{M_1 M_2}{r^2}\right)}$$

Position And Velocity of Agent

Calculated as

Velocity(t+1) = rand * Velocity(t) + Acceleration (t)

Position(t+1) = Position(t) + Velocity (t+1);

Uniform Random variable. Interval [0,1]

GSA Steps

1. Initialization Phase
2. Randomly Generate Search Space of size N
3. Evaluate the Fitness Value for each Individual / Agents/Object
4. Set Counter t = 0. // Iteration Counter
5. Update Best and Worst Value in the Population
6. Calculate G, Mass, Acceleration, Force for current Population
7. Update Position and Velocity for each Object/Agent
8. Evaluate the Fitness values
9. Repeat Until Stopping Criteria met
10. Return Best Solution
11. End

STEP 1

Population Initialization Phase

N Agents Population is Given as:

X_1 = Position of Search Agents.

Here, N = 4

We have Population of 4 Search Agents. (i = 1,2,3,4)

X1, X2, X3 and X4

Domain [-100, 100]

Low = -100, **High** = 100

STEP 2

Randomly Generate Search Space of size N

Sr. No	Population Position
1	63.9447
2	81.1584
3	-74.6026
4	82.6752

STEP 3

Fitness Value Evaluation Phase

Sr. No	Fitness Evalution
1	3.920
2	6.5867
3	5.5655
4	6.8352

F(X1), F(X2), F(X3), F(X4)

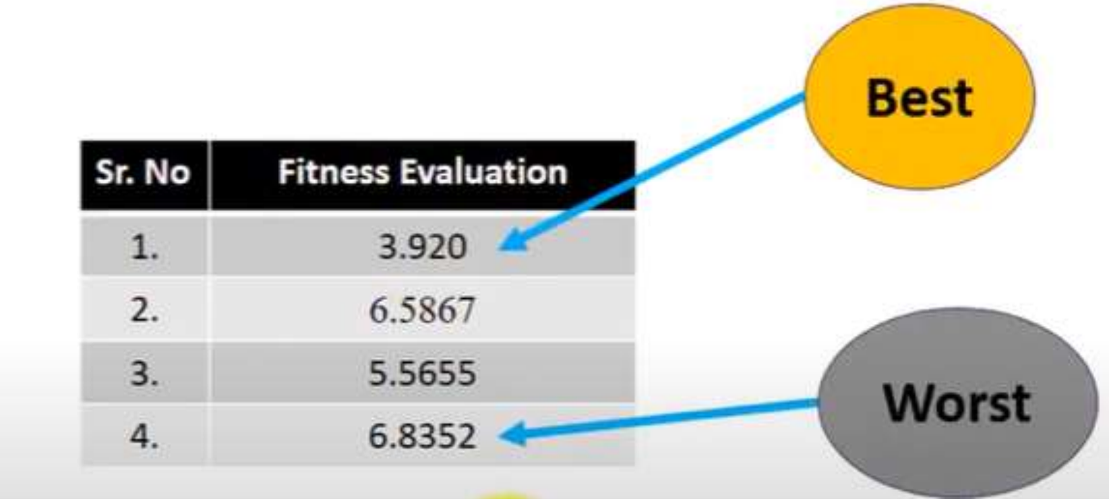
Here, N=4, (i = 1,2,3,4) [-100,100]

$$F_1(X) = \sum_{i=1}^{n} x_i^2$$

STEP 4

Set Counter $t = 0$

So, $t = 0$ // Iteration Counter



Sr. No	Fitness Evaluation
1.	3.920
2.	6.5867
3.	5.5655
4.	6.8352

Step 5

Update Best And Worst Value

<u>For Minimization:</u>

- Consider Minimum Fitness Value **BEST**
- Maximum Fitness Value **WORST**

Step 6 (i)

Gravitational constant calculation(**G**)

For G Constant Consider:

$\alpha = 20;$

Iteration = 0;

G0 = 100;

$$G = G0 * \exp\left(\frac{-\alpha * iteration}{MaximumIteration}\right)$$

Iterations (t)	G Value
t = 0	100
t = 1	98.0199
t = 2	96.0789
t = 3	94.1765

Step 6 (ii)

Mass Calculation (**Mass**)

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)},$$

Sr. No	Mass Calculation
1.	0.6576
2.	0.0561
3.	0.2864
4.	0

Sr. No	Distance Calculation
1.	18.2137
2.	156.1210
3.	157.2778
4.	157.2787

Sr. No	Force Calculation
1.	0.0016
2.	1.0623
3.	0.0014
4.	0

Step 6 (iii)
Force Calculation (Force)

$$F_{ij}^{d}(t) = G(t)\,(\frac{M_{pi}(t) * M_{aj}(t)}{R_{ij}(t) + \varepsilon}\,(X_j^{d}(t) - X_i^{d}(t))$$

Sr. No	Force Calculation
1.	0.0016
2.	1.0623
3.	0.0014
4.	0

Iterations (t)	G Value
t = 0	100
t = 1	98.0199
t = 2	96.0789
t = 3	94.1765

Step 6 (iv)

Acceleration Calculation

Sr. No	Acceleration Calculation
1.	0.1600
2.	106.2300
3.	0.1400
4.	0

Step 7

<u>Update Position and velocity</u>

Update Position and Velocity for each Object/Agent.

Velocity(t+1) = rand * Velocity(t) + Acceleration (t);

Position(t+1)= Position(t) + Velocity (t+1);

Sr. No	Velocity Calculation
1.	0.1600
2.	106.2300
3.	1.1400
4.	0

Sr. No	Acceleration Calculation
1.	0.1600
2.	106.2300
3.	0.1400
4.	0

Sr. No	Position Calculation
1.	63.1047
2.	187.3884
3.	-73.4626
4.	82.6452

Step 8

Fitness Value Evaluation Phase

Sr. No	Fitness Evaluation
1.	3.9822
2.	3.5114
3.	5.3968
4.	6.8302

Sr. No	Position Calculation
1.	63.1047
2.	187.3884
3.	-73.4626
4.	82.6452

F(X1), F(X2), F(X3), F(X4)
Here, N=4, (i = 1,2,3,4) [-100,100]

$$F_1(X) = \sum_{i=1}^{n} x_i^2$$

After One Iteration

Sr. No	Fitness Evaluation
1.	3.920
2.	6.5867
3.	5.5655
4.	6.8352

Previous

Sr. No	Fitness Evaluation
1.	3.9822
2.	3.5114
3.	5.3968
4.	6.8302

New

Step 9

Repeat Until Stopping Criteria Met.

Step 10

When Stopping Criteria met Return best Solution


Any Questions


THANK
YOU