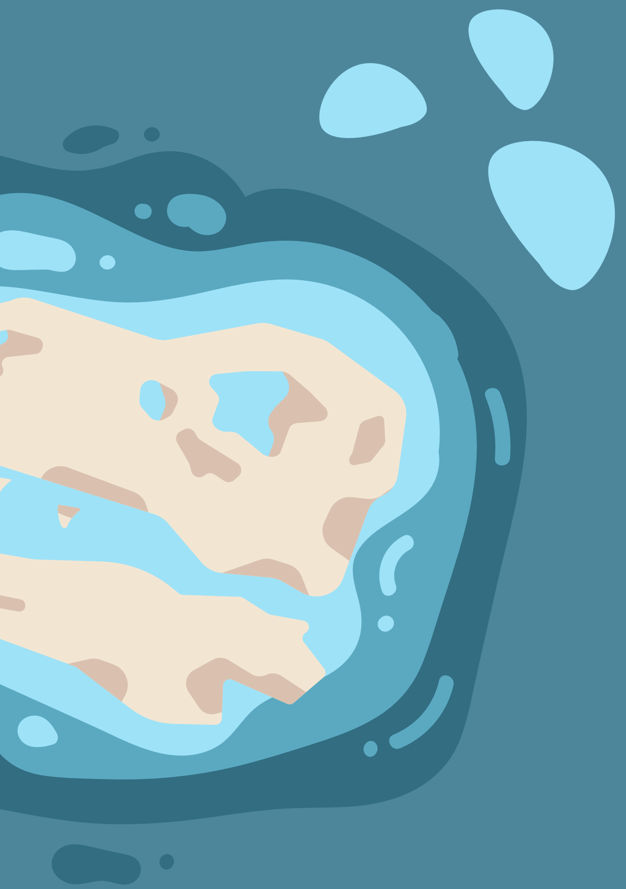
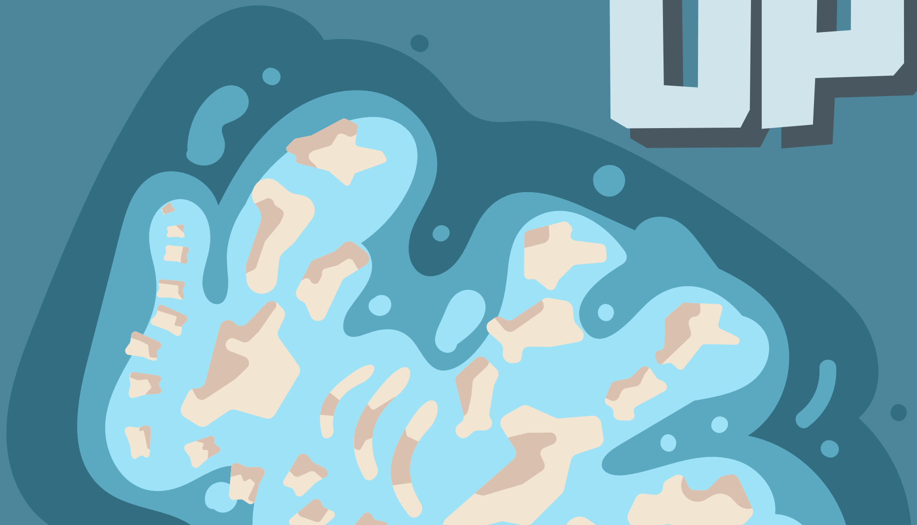
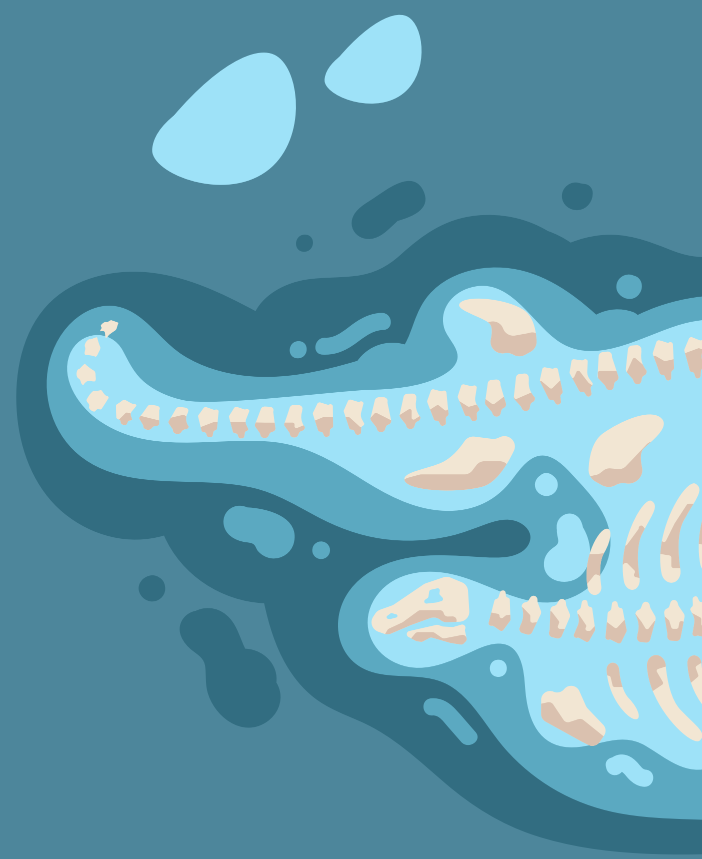
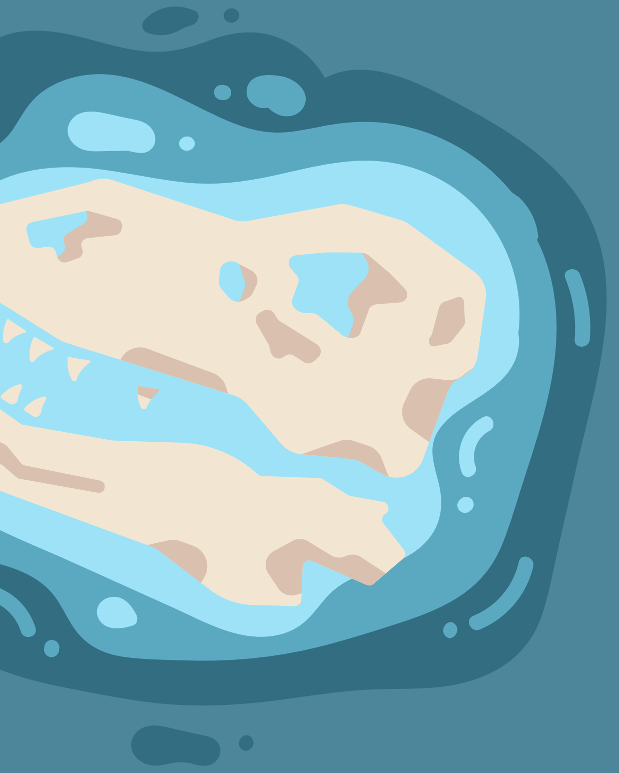
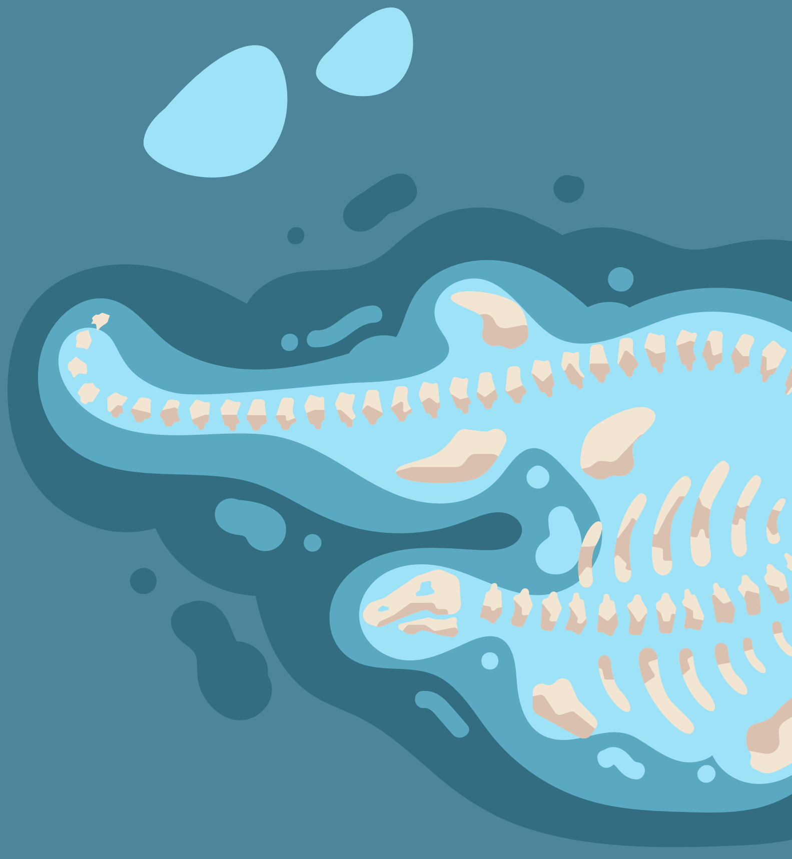
EQUILIBRIUM OPTIMIZER

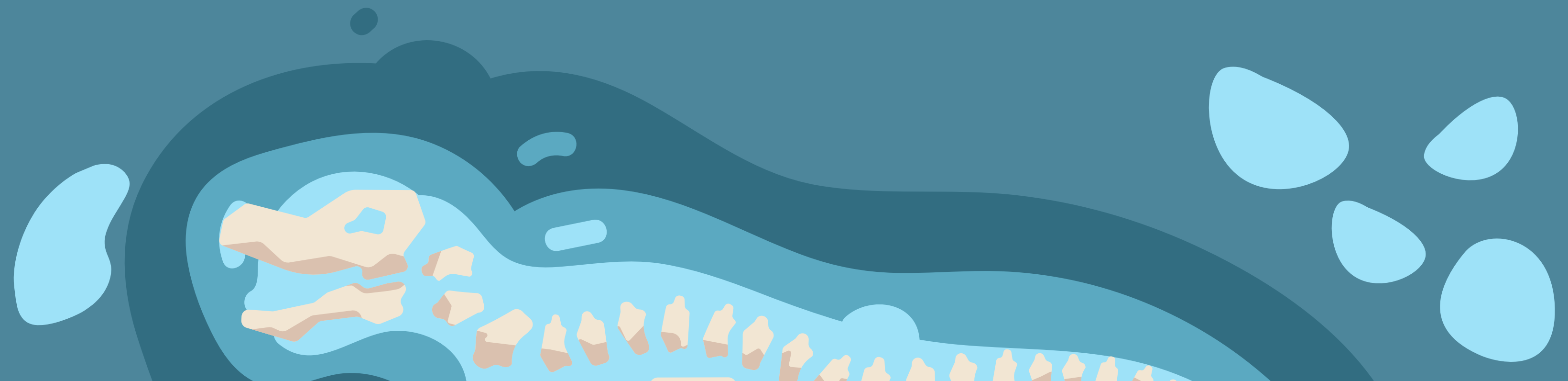






PARTICIPANTS

Yousef Mohamed Elbaz
Sameh Hussein Elsayed
Zeyad Awad
Mostafa Mahmoud
Zeyad Metwaly
Zeyad Medhat



INTRODUCTION

There are two major categories for mathematical optimization methods: deterministic and stochastic. Linear and non-linear programming are some of the most commonly used deterministic methods, characterized by using the gradient information of the problem to search the space and find the solution.

Physicists find out calculus that defined best solution that can be essential for decision making process

Optimum: Technique Of Find The Best Solution,
Optimization is the process of making a trading system more effective by adjusting the variables used for technical analysis in many different fields

EQUILIBRIUM OPTIMIZER

- Equilibrium refers to the condition of balanced which stated that the opposing forced or influences are balanced .

- Equilibrium Optimizer (EO) is a novel optimization algorithm inspired by control volume mass balance models used to estimate both dynamic and equilibrium states.

- Each particle (solution) with its concentration (position) acts as a search agent. The search agents randomly update their concentration with respect to best-so-far solutions

INISPIRATION

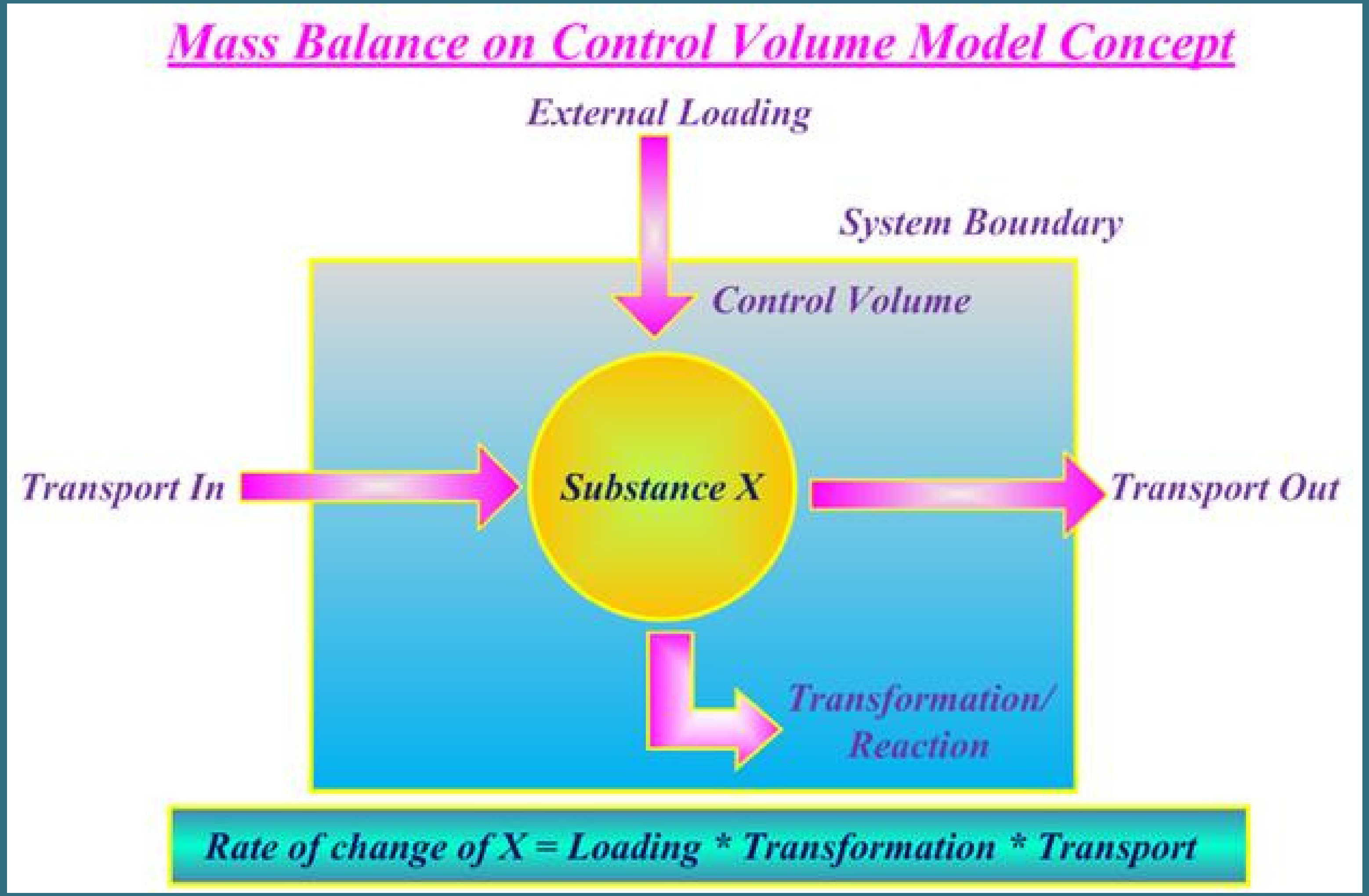


Fig 1 : Mass balance on control volume model concept

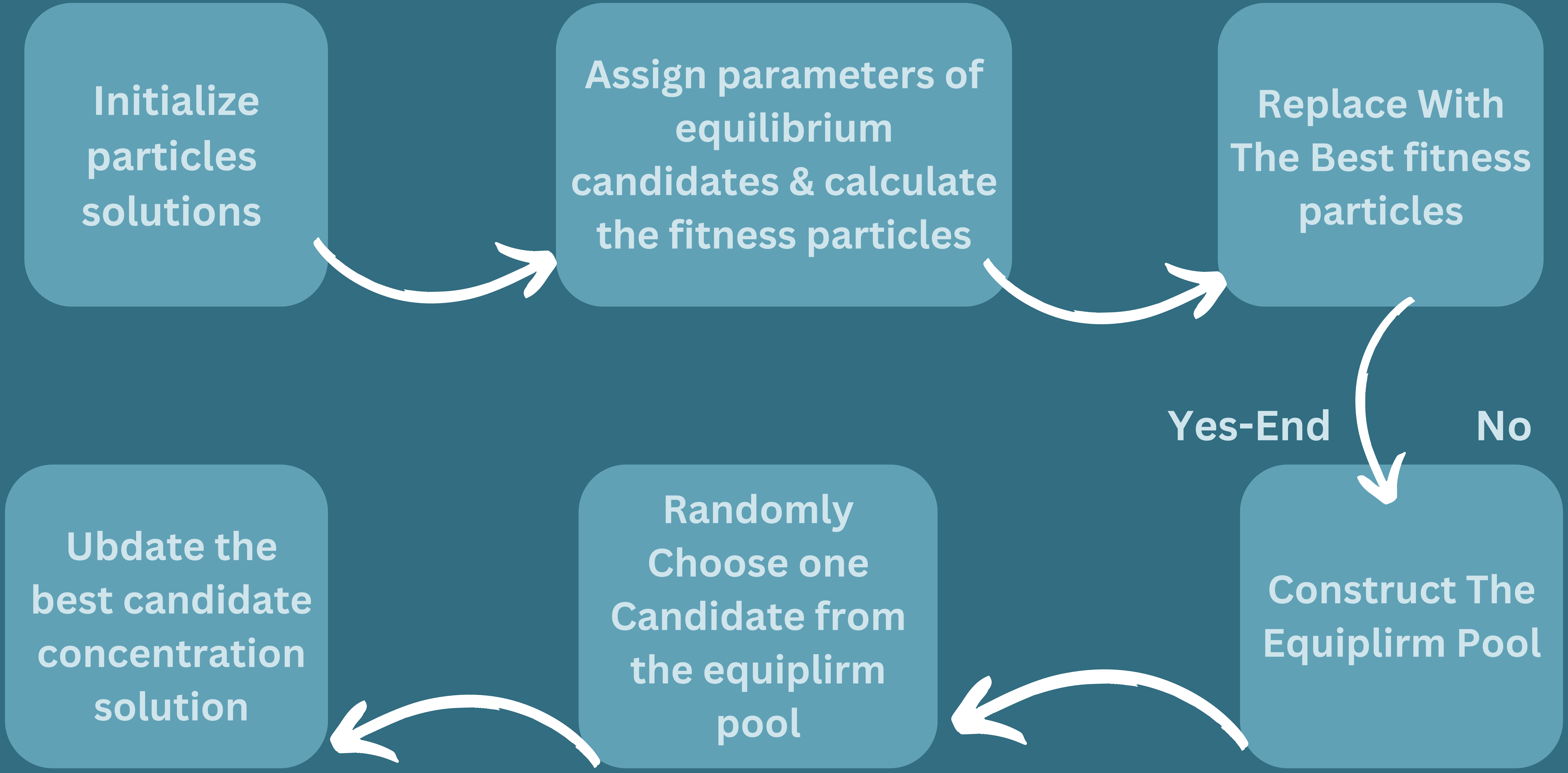
- A mass balance is only meaningful in terms of a specific region of space.
- This region is called the control volume.

$$\overrightarrow{\mathrm{Gen}} = \overrightarrow{\mathrm{E\,equ}} + (\vec{\mathrm{E}} - \overrightarrow{\mathrm{E\,equ}}) * \vec{\mathrm{F}} + \frac{\overrightarrow{\mathrm{Gen}}}{\vec{\varphi}\mathrm{Z}}(1-\vec{\mathrm{F}})$$

Fig 2 :differential equation for generic mass-balance

- A first-order ordinary differential equation expressing the generic mass-balance equation

FLOW CHART



Initialize particles solutions

Assign parameters of equilibrium candidates & calculate the fitness particles

Replace With The Best fitness particles

Yes-End

No

Construct The Equiplirm Pool

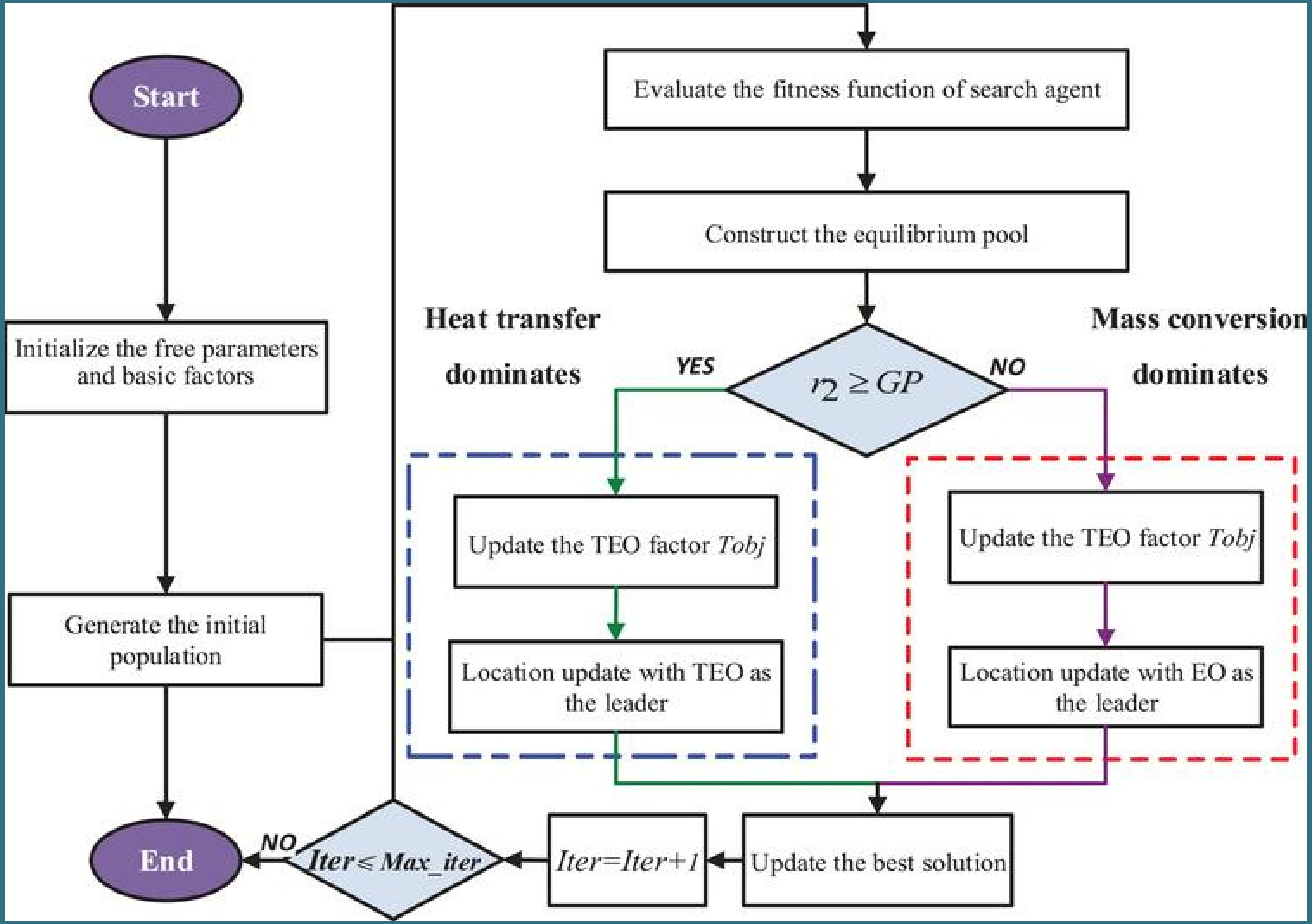
Randomly Choose one Candidate from the equiplirm pool

Ubdate the best candidate concentration solution

How EO Works ?!

- In EO, a particle is analogous to a solution and a concentration is analogous to a particle's position like Pso algorithm.
- There are three terms that presenting ubdating for particle and concentrarions : equilibrium concentration(CEQ),concentration difference(C1), generation rate (G)
- Equilibrium Concentration : defined as one of the best-so-far solutions randomly selected from a pool, called the equilibrium pool.
- concentration difference between a particle and the equilibrium state, which acts as a direct search mechanism. encourages particles to globally search the domain, acting as explorers.
- Generation Rate : which mostly plays the role of an exploiter, orsolution refiner, particularly with small steps

FLOW CHART



Start

Initialize the free parameters and basic factors

Generate the initial population

Evaluate the fitness function of search agent

Construct the equilibrium pool

Heat transfer dominates

YES

$r_2 \geq GP$

NO

Mass conversion dominates

Update the TEO factor *Tobj*

Location update with TEO as the leader

Update the TEO factor *Tobj*

Location update with EO as the leader

Update the best solution

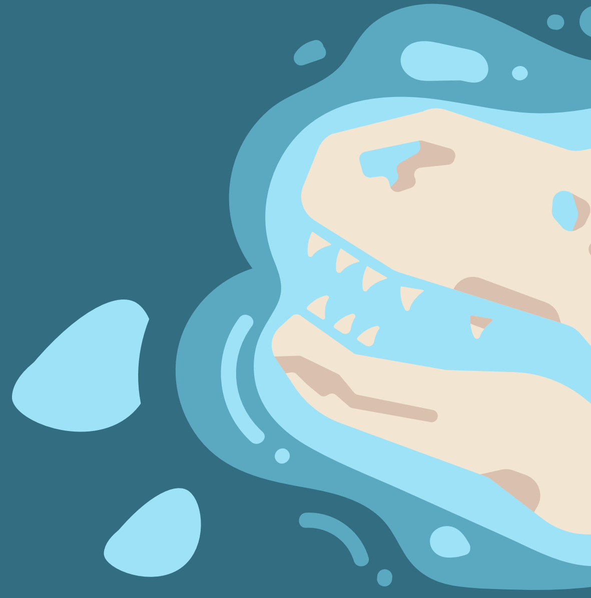
Iter=Iter+1

Iter≤Max_iter

NO

End

Fig 3 : Flow Chart For Equilibrium optimizer Algorithm



EXPLANATION PHASE

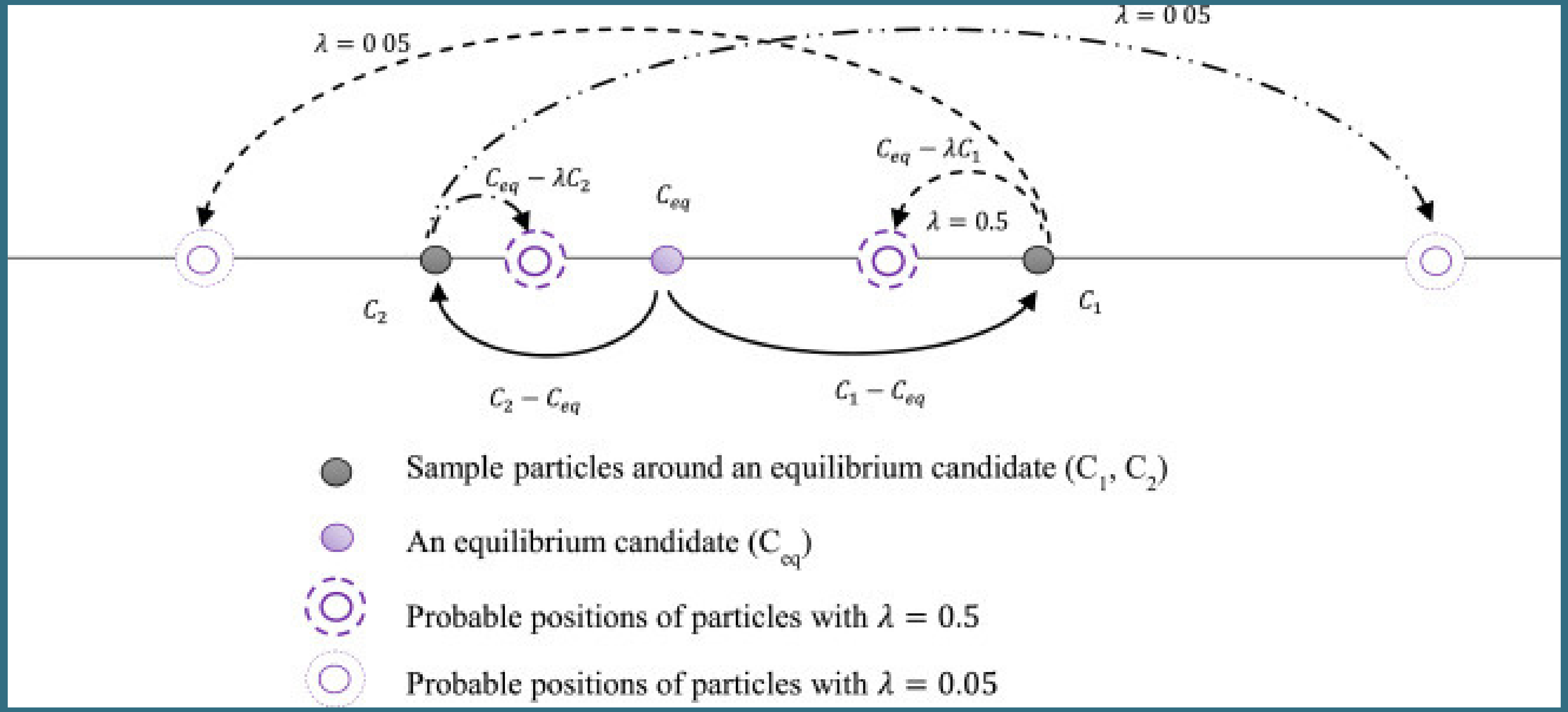
Inspiration

mass balance equation is used to describe the concentration of a nonreactive constituent in a control volume.

$$V\frac{dC}{dt} = QC_{eq} - QC + G$$

Equation (1)

C is the concentration, inside control volume (V), $V\ dC\ /dt$ is the rate of change , Q is the volumetric flow rate,Ceq represents concentration at equilibrium state ,G is the mass generation rate



$\lambda = 0.05$
$\lambda = 0.05$
$C_{eq} - \lambda C_1$
$C_{eq} - \lambda C_2$
C_{eq}
$\lambda = 0.5$
C_2
C_1
$C_2 - C_{eq}$
$C_1 - C_{eq}$
● Sample particles around an equilibrium candidate (C_1, C_2)
● An equilibrium candidate (C_{eq})
○ Probable positions of particles with $\lambda = 0.5$
○ Probable positions of particles with $\lambda = 0.05$

Fig (4) demonstrates a 1-D version exploration and exploitation.

When $V\ dC\ /dt$ reaches to zero, a steady equilibrium state is reached.

$$C = C_{eq} + (C_0 - C_{eq})\, F + \frac{G}{\lambda V}(1 - \mathrm{F})$$

Equation (3)

to and Co are the initial start time and concentration ,dependent on the integration interval in Equation (2)

$$\int_{C_0}^{C} \frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = \int_{t_0}^{t} dt$$

Equation (2)

can be used to either estimate the concentration in the control volume with a known turnover rate or to calculate the average turnover rate using a simple linear regression.

F is calculated as follows

$$F = \exp[-\lambda(\mathrm{t} - t_0)]$$

Equation (4)

An accurate definition of this term will assist having a reasonable balance between exploration and exploitation

Initialization

$$C_i^{initial} = C_{min} + rand_i(C_{max} - C_{min}) \qquad i = 1,2, \ldots\ldots n$$

Equation (5)

The initial concentrations are constructed based on the number of particles and dimensions with uniform random initialization

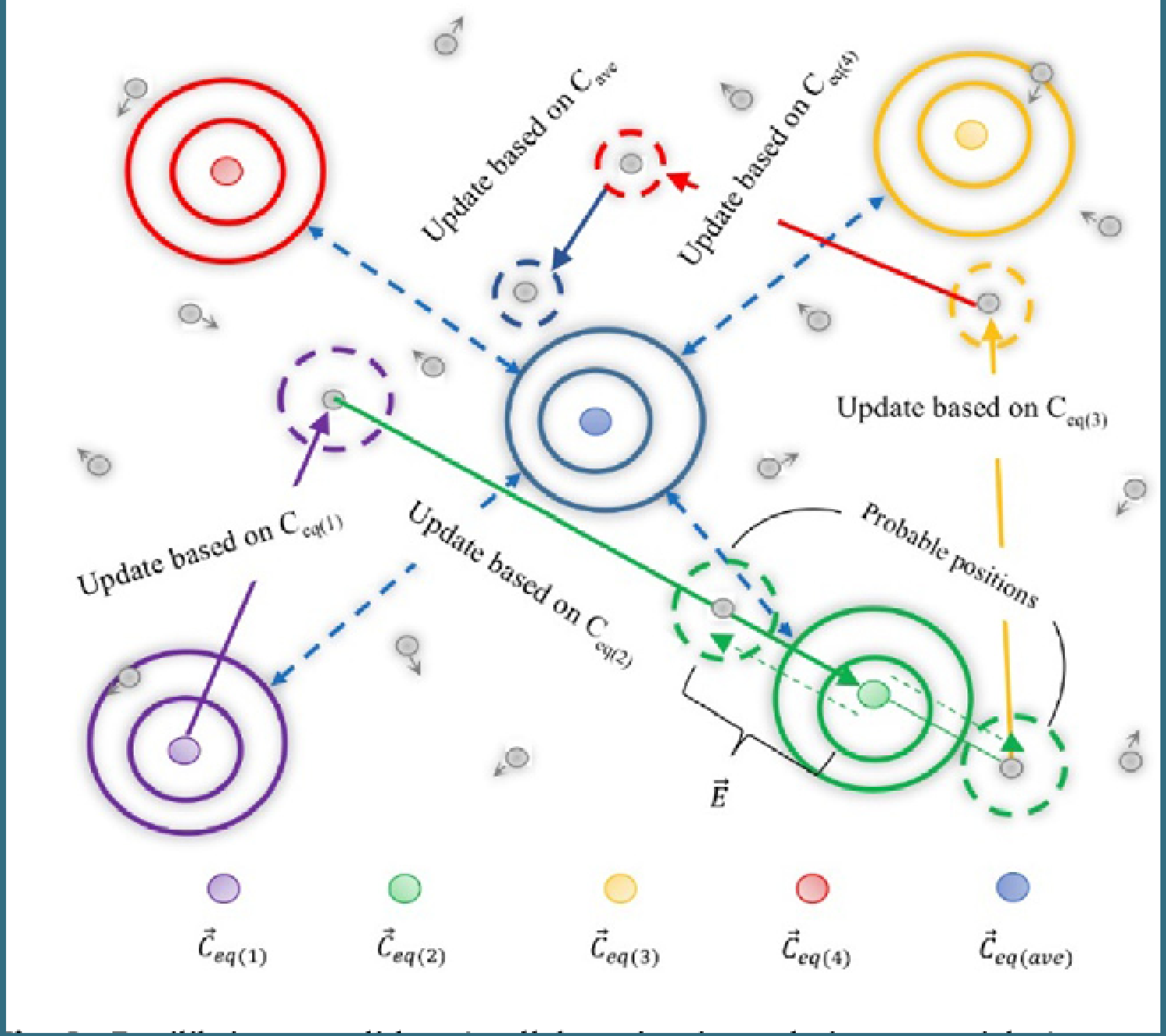

Update based on C_{ave}
Update based on $C_{eq(4)}$
Update based on $C_{eq(3)}$
Update based on $C_{eq(1)}$
Update based on $C_{eq(2)}$
Probable positions
$\vec{E}$
$\vec{C}_{eq(1)}$
$\vec{C}_{eq(2)}$
$\vec{C}_{eq(3)}$
$\vec{C}_{eq(4)}$
$\vec{C}_{eq(ave)}$

Figure 5 Equilibrium candidates' collaboration in updating a particles' concentration in 2D dimensions

Initialization

$$C_i^{initial} = C_{min} + rand_i(C_{max} - C_{min}) \qquad i = 1,2, \ldots..n$$

Equation (6)

The initial concentrations are constructed based on the number of particles and dimensions with uniform random initialization

Ci initial is the initial concentration vector of the i-th particle *Cmin* and *Cmax* denote the minimum and maximum values for the dimensions Randi is a random vector in the interval of [0,1], and n is the number of particles.

Equilibrium pool and candidates (Ceq)

The equilibrium state is the final convergence state of the algorithm, which is desired to be the global optimum.

$$\vec{C}_{eq,pool} = \{\ \vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\ \}$$

Equation (7)

Exponential term (*F*)

$$\vec{F} = e^{-\vec{\lambda}(t-t_0)}$$

Equation (8)

λ is assumed to be a random vector in the interval of [0,1].

An accurate definition of this term will assist EO in having a reasonable balance between exploration and exploitation.

function of iteration (Iter)

$$t = (1 - \frac{Iter}{Max\_iter})^{(a_2 \frac{Iter}{Max\_iter})}$$

Equation (9)

Iter and *Max_iter* present the current and the maximum number of iterations

*a*2 is a constant value used to manage exploitation ability

$$\vec{t}_0 = \frac{1}{\vec{\lambda}} \ln\left(-a_1 sign(\vec{r} - 0.5)[1 - e^{-\vec{\lambda}t}]\right) + t$$

Equation (10)

*a*1 is a constant value that controls exploration ability.

The higher the a1, the better the exploration ability and consequently the lower exploitation performance. in Equation (10)

$$\vec{t}_0 = \frac{1}{\vec{\lambda}} \ln\left(-a_1 sign(\vec{r} - 0.5)[1 - e^{-\vec{\lambda} t}]\right) + t$$ Equation (10)

Similarly, the higher the a2, the better the exploitation ability and the lower the exploration ability.in Equation (9)

$$t = \left(1 - \frac{Iter}{Max\_iter}\right)^{\left(a_2 \frac{Iter}{Max\_iter}\right)}$$ Equation (9)

a1 and a2 are equal to 2 and 1, respectively.

r is a random vector between 0 and 1

Eq. (11) shows the revised version of Eq. (8) with the substitution of Eq. (10) into Eq. (8).

$$\vec{F} = a_1 sign(\vec{r} - 0.5)\left[e^{-\vec{\lambda} t} - 1\right]$$ Equation (11)

$sign(r$ - 0.5) effects on the direction of exploration and exploitation. in Equation (10)

Generation rate (G)

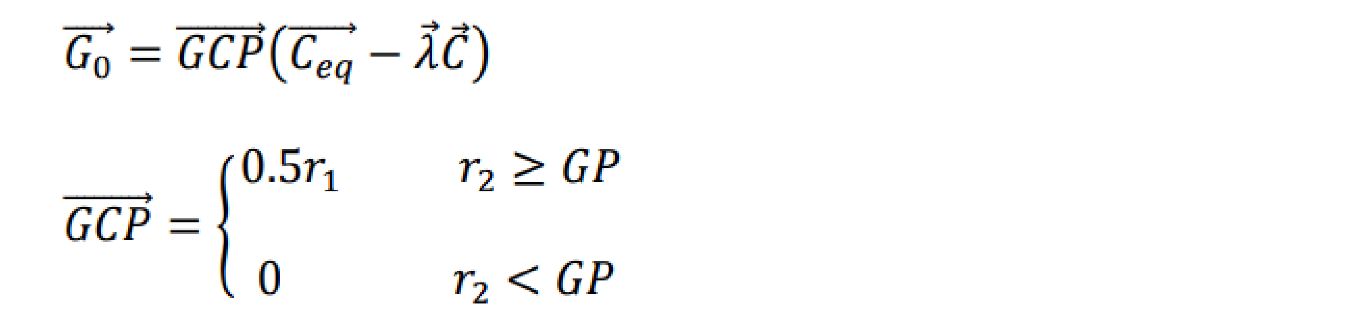
$$\vec{G} = \vec{G}_0\, e^{-\vec{k}(t - t_0)}$$ Equation (12)

The generation rate is to provide the exact solution by improving the exploitation phase.

G0 is the initial value and k indicates a decay constant.

These parameters can be tuned for other problems as needed.

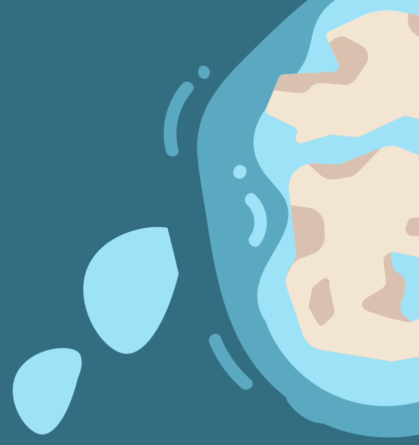
Generation rate Control Parameter (GCP)



$$\overrightarrow{G_0} = \overrightarrow{GCP}\left(\overrightarrow{C_{eq}} - \vec{\lambda}\vec{C}\right)$$

$$\overrightarrow{GCP} = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases}$$

Equation (13)



If **r2>=0.5**, " Heat Transfer"ubdate the term of r1,then ubdate location of leader.
If **r2 < 0.5** " Massive Conversion"ubdate the term of r1,then ubdate location of leader.

Particle's memory saving : Adding memory saving procedures assists each particle in keeping track of its coordinates in the space, which also informs its fitness value. This mechanism resembles the pbest concept in PSO. The fitness value of each particle in the current iteration is compared to that of the previous iteration and will be overwritten if it achieves a better fit. This mechanism aids in exploitation capability but can increase the chance of getting trapped in local minima if the method does not benefit from global exploration ability [23].

Exploration ability of EO

- $a1$ controls the exploration quantity (magnitude) of the algorithm.
- $sign(r - 0.5)$ controls the exploration direction.
- Generation probability (GP) controls the participation probability of concentration.
- Equilibrium pool: This vector consists of five particles.

Exploitation ability of EO

- $a2$ this parameter is similar to $a1$, but controls the exploitation feature.
- $sign(r - 0.5)$: controls the exploitation quality.
- Memory saving memory saving, saves a number of best-so-far particles.
- Equilibrium pool by lapse of iteration, exploration fades out and exploitation fades in.

The comparative methods for this section include three categories of optimization methods: Genetic Algorithm (GA) [7] and Particle Swarm Optimization (PSO) [8] as the most well-known and well-studied evolutionary and swarm intelligence algorithms.

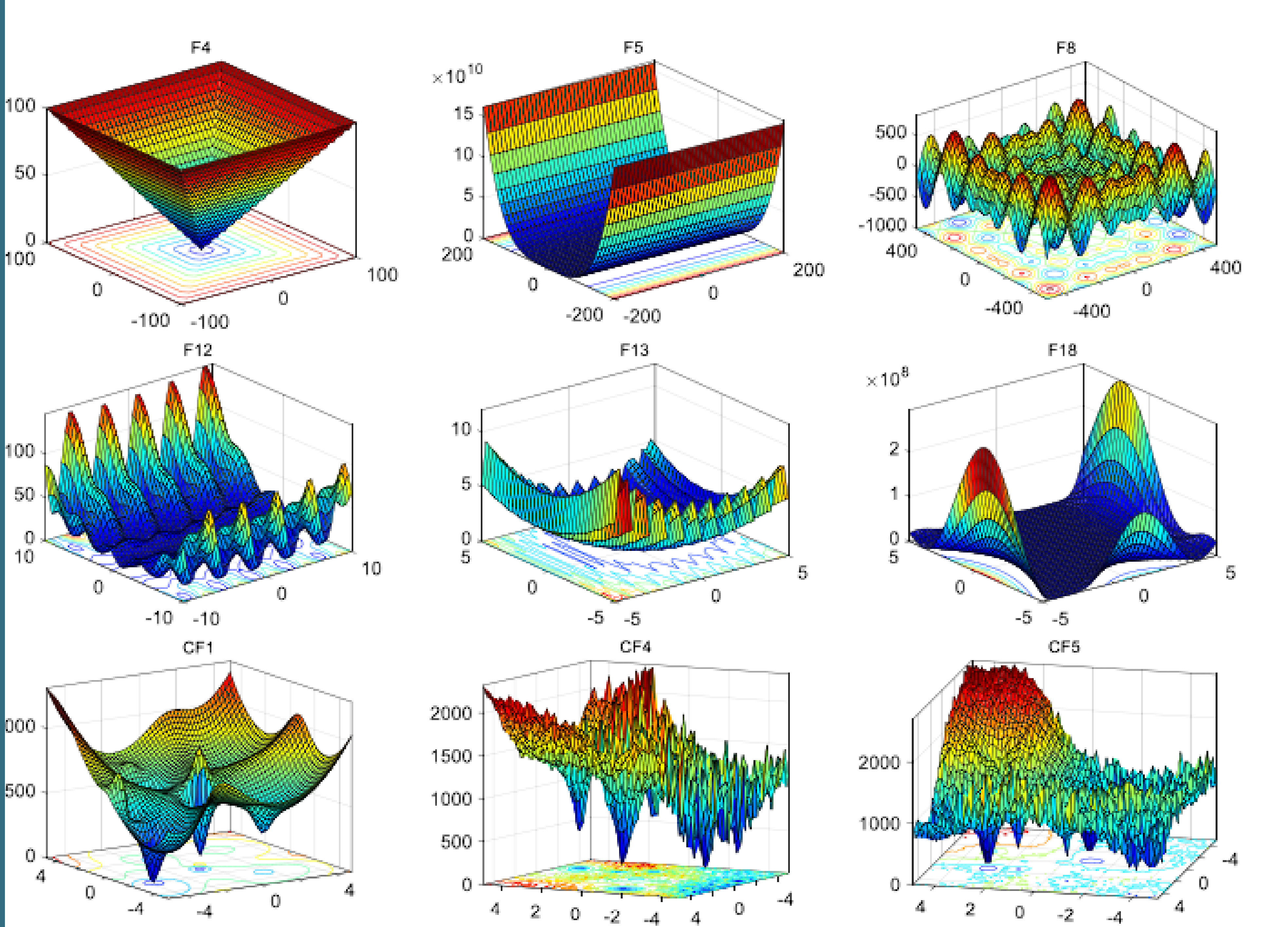
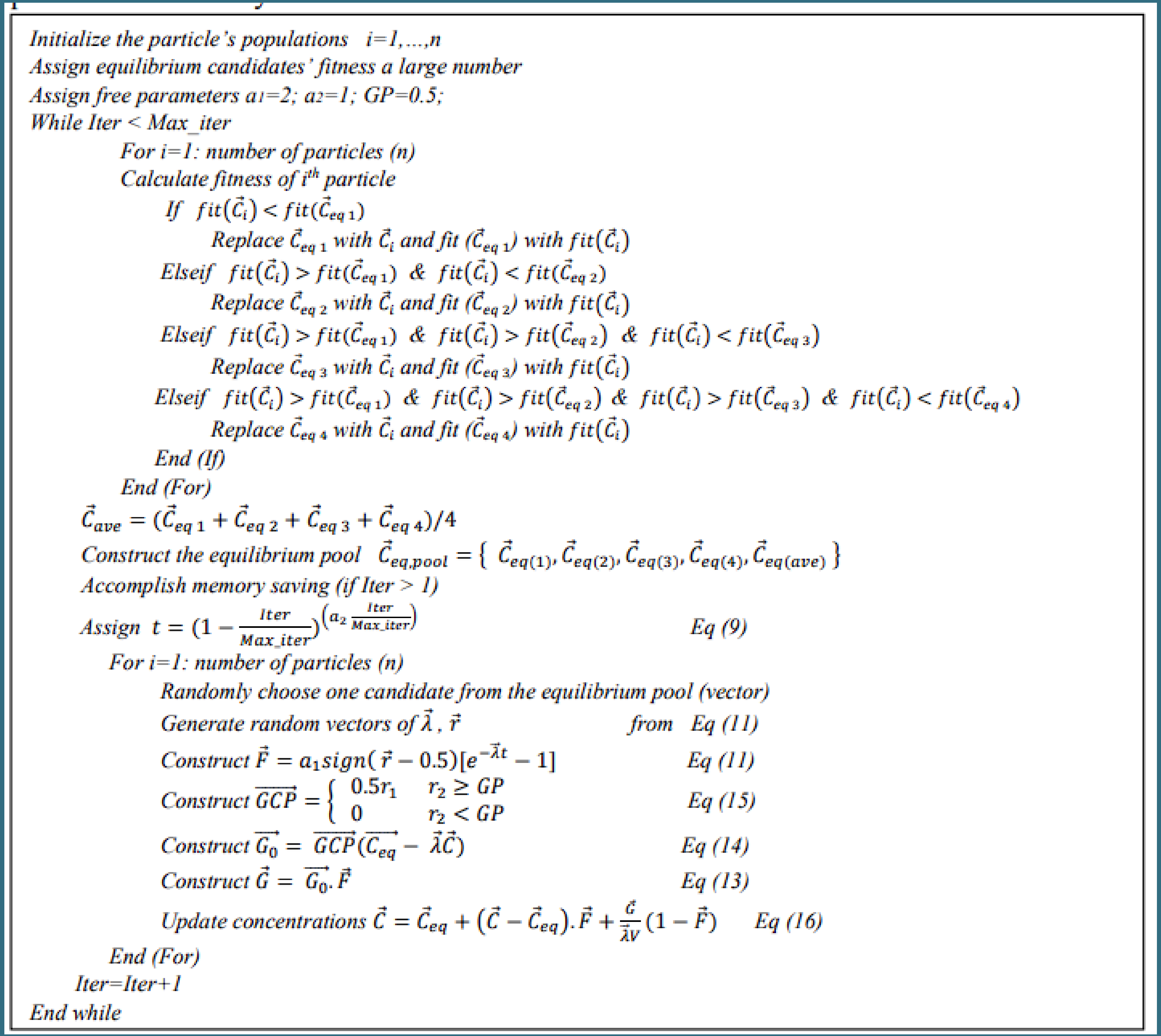


Figure 3. A two-dimensional perspective view for couple of the mathematical benchmark functions.

PSEUDOCODE OF EEQUILIBRIUM OPTIMIZER



```
Initialize the particle's populations   i=1,...,n
Assign equilibrium candidates' fitness a large number
Assign free parameters $a_1=2$; $a_2=1$; GP=0.5;
While Iter < Max_iter
        For i=1: number of particles (n)
        Calculate fitness of i^th particle
            If  $fit(\vec{C}_i) < fit(\vec{C}_{eq\,1})$
                Replace $\vec{C}_{eq\,1}$ with $\vec{C}_i$ and fit $(\vec{C}_{eq\,1})$ with $fit(\vec{C}_i)$
            Elseif  $fit(\vec{C}_i) > fit(\vec{C}_{eq\,1})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq\,2})$
                Replace $\vec{C}_{eq\,2}$ with $\vec{C}_i$ and fit $(\vec{C}_{eq\,2})$ with $fit(\vec{C}_i)$
            Elseif  $fit(\vec{C}_i) > fit(\vec{C}_{eq\,1})$ & $fit(\vec{C}_i) > fit(\vec{C}_{eq\,2})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq\,3})$
                Replace $\vec{C}_{eq\,3}$ with $\vec{C}_i$ and fit $(\vec{C}_{eq\,3})$ with $fit(\vec{C}_i)$
            Elseif  $fit(\vec{C}_i) > fit(\vec{C}_{eq\,1})$ & $fit(\vec{C}_i) > fit(\vec{C}_{eq\,2})$ & $fit(\vec{C}_i) > fit(\vec{C}_{eq\,3})$ & $fit(\vec{C}_i) < fit(\vec{C}_{eq\,4})$
                Replace $\vec{C}_{eq\,4}$ with $\vec{C}_i$ and fit $(\vec{C}_{eq\,4})$ with $fit(\vec{C}_i)$
            End (If)
        End (For)
    $\vec{C}_{ave} = (\vec{C}_{eq\,1} + \vec{C}_{eq\,2} + \vec{C}_{eq\,3} + \vec{C}_{eq\,4})/4$
    Construct the equilibrium pool  $\vec{C}_{eq,pool} = \{ \vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)} \}$
    Accomplish memory saving (if Iter > 1)
    Assign  $t = (1 - \frac{Iter}{Max\_iter})^{(a_2 \frac{Iter}{Max\_iter})}$                    Eq (9)
      For i=1: number of particles (n)
          Randomly choose one candidate from the equilibrium pool (vector)
          Generate random vectors of $\vec{\lambda}$, $\vec{r}$                    from  Eq (11)
          Construct $\vec{F} = a_1 sign(\vec{r} - 0.5)[e^{-\vec{\lambda}t} - 1]$                    Eq (11)
          Construct $\overrightarrow{GCP} = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases}$                    Eq (15)
          Construct $\overrightarrow{G_0} = \overrightarrow{GCP}(\overrightarrow{C_{eq}} - \vec{\lambda}\vec{C})$                    Eq (14)
          Construct $\vec{G} = \overrightarrow{G_0}.\vec{F}$                    Eq (13)
          Update concentrations $\vec{C} = \vec{C}_{eq} + (\vec{C} - \vec{C}_{eq}).\vec{F} + \frac{\vec{G}}{\vec{\lambda}V}(1 - \vec{F})$        Eq (16)
      End (For)
    Iter=Iter+1
End while
```

Computational complexity

Computational complexity of an optimization algorithm is presented by a function relating the running time of the algorithm to the input size of problem. For this purpose, Big-O notation is used here as a common terminology. Complexity is dependent upon the number of particles (n), the number of dimensions (d), and the number of iterations (t), and (c) is the cost of function evaluation

$$O(EO) = O(\mathit{problem\ definition}) + O(\mathit{initialization}) + O(t(\mathit{function\ evaluations})) + O(t(\mathit{Memory\ saving})) + O(t(\mathit{Concentration\ Update}))$$

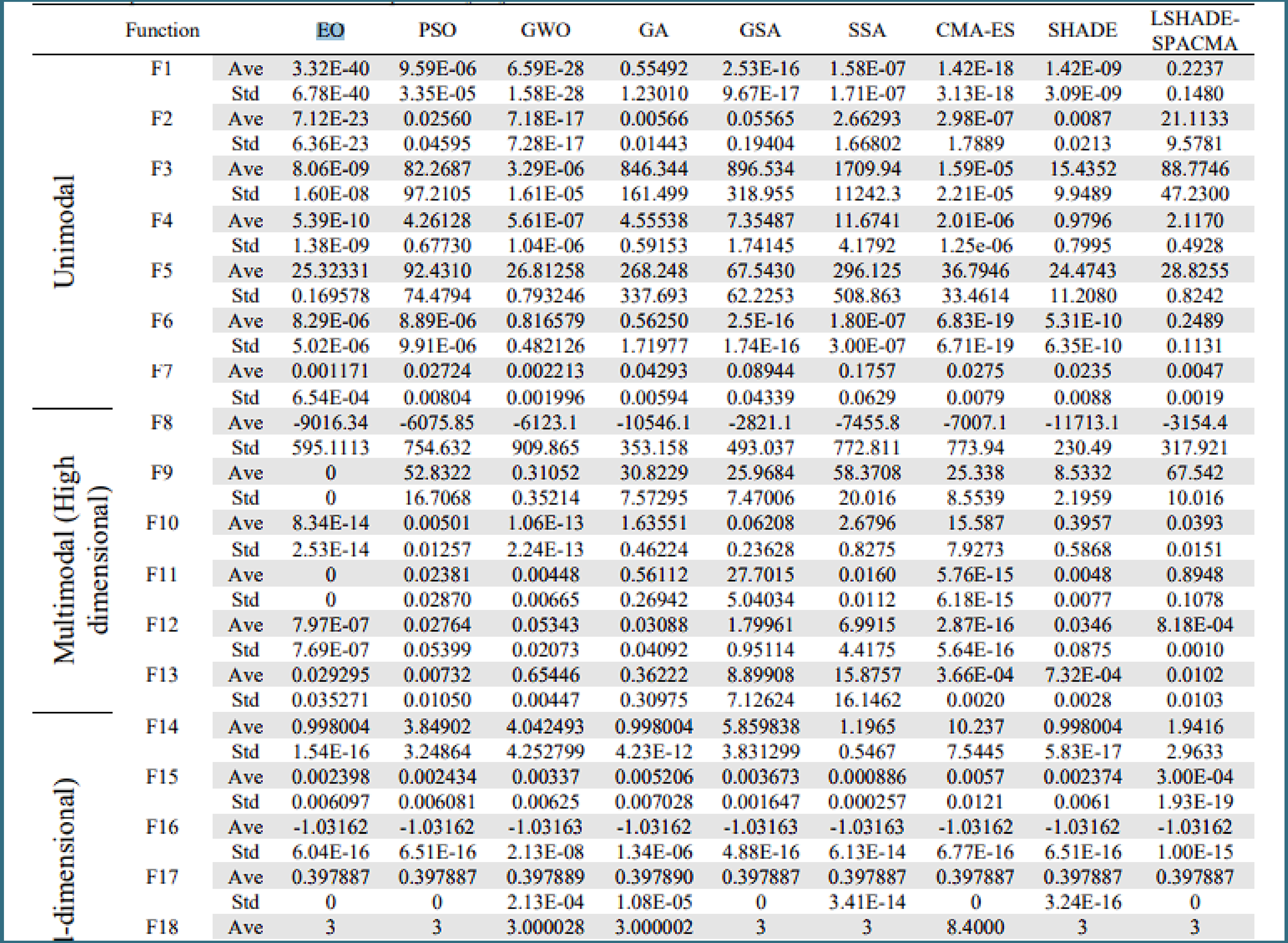
The overall computational complexity is defined as: $O(EO) = O(1 + nd + tcn + tn + tnd) \cong O(tnd + tcn)$

Results on Benchmark Functions

This section demonstrates the effectiveness of the proposed algorithm on a set of 58 benchmark test functions, including 29 commonly used unimodal, multimodal, and composition functions, as well as another 29 functions from the CEC-BC-2017 test suite [25]. This study utilizes both quantitative and qualitative validation metrics. Quantitative metrics include the average and standard deviation values for different test functions and qualitative metrics include trajectory, search, optimization, and average fitness history.

F1	Sphere	F14	Kowalik	F26	Alpine
F2	Rastrigin	F15	Shekel Foxholes	F27	Michalewicz
F3	Ackley	F16	Goldstein-Price	F28	Exponential
F4	Griewank	F17	Shekel 5	F29	Schaffer
F5	Schwefel	F18	Branin	F30	Bent Cigar
F6	P2.22	F19	Hartmann 3	F31	Bohachevsky 1
F7	Rosenbrock	F20	Shekel 7	F32	Elliptic
F8	Sehwwefel	F21	Shekel 10	F33	Drop Wave
F9	P2.21	F22	Six-Hump Camel-	F34	Cosine Mixture
F1	Quartic	F23	Back	F15	Ellipsoidal
0	Schwefel P1.2	F24	Hartmann 6	F36	Levy and
F11	Penalized 1	F25	Zakharov		Montalvo 1
F1	Penalized 2		Sum Squares		
2	Schwefel				
F1	P2.26				

BENCHMARK FUNCTIONS



	Function		EO	PSO	GWO	GA	GSA	SSA	CMA-ES	SHADE	LSHADE-SPACMA
Unimodal	F1	Ave	3.32E-40	9.59E-06	6.59E-28	0.55492	2.53E-16	1.58E-07	1.42E-18	1.42E-09	0.2237
		Std	6.78E-40	3.35E-05	1.58E-28	1.23010	9.67E-17	1.71E-07	3.13E-18	3.09E-09	0.1480
	F2	Ave	7.12E-23	0.02560	7.18E-17	0.00566	0.05565	2.66293	2.98E-07	0.0087	21.1133
		Std	6.36E-23	0.04595	7.28E-17	0.01443	0.19404	1.66802	1.7889	0.0213	9.5781
	F3	Ave	8.06E-09	82.2687	3.29E-06	846.344	896.534	1709.94	1.59E-05	15.4352	88.7746
		Std	1.60E-08	97.2105	1.61E-05	161.499	318.955	11242.3	2.21E-05	9.9489	47.2300
	F4	Ave	5.39E-10	4.26128	5.61E-07	4.55538	7.35487	11.6741	2.01E-06	0.9796	2.1170
		Std	1.38E-09	0.67730	1.04E-06	0.59153	1.74145	4.1792	1.25e-06	0.7995	0.4928
	F5	Ave	25.32331	92.4310	26.81258	268.248	67.5430	296.125	36.7946	24.4743	28.8255
		Std	0.169578	74.4794	0.793246	337.693	62.2253	508.863	33.4614	11.2080	0.8242
	F6	Ave	8.29E-06	8.89E-06	0.816579	0.56250	2.5E-16	1.80E-07	6.83E-19	5.31E-10	0.2489
		Std	5.02E-06	9.91E-06	0.482126	1.71977	1.74E-16	3.00E-07	6.71E-19	6.35E-10	0.1131
	F7	Ave	0.001171	0.02724	0.002213	0.04293	0.08944	0.1757	0.0275	0.0235	0.0047
		Std	6.54E-04	0.00804	0.001996	0.00594	0.04339	0.0629	0.0079	0.0088	0.0019
Multimodal (High dimensional)	F8	Ave	-9016.34	-6075.85	-6123.1	-10546.1	-2821.1	-7455.8	-7007.1	-11713.1	-3154.4
		Std	595.1113	754.632	909.865	353.158	493.037	772.811	773.94	230.49	317.921
	F9	Ave	0	52.8322	0.31052	30.8229	25.9684	58.3708	25.338	8.5332	67.542
		Std	0	16.7068	0.35214	7.57295	7.47006	20.016	8.5539	2.1959	10.016
	F10	Ave	8.34E-14	0.00501	1.06E-13	1.63551	0.06208	2.6796	15.587	0.3957	0.0393
		Std	2.53E-14	0.01257	2.24E-13	0.46224	0.23628	0.8275	7.9273	0.5868	0.0151
	F11	Ave	0	0.02381	0.00448	0.56112	27.7015	0.0160	5.76E-15	0.0048	0.8948
		Std	0	0.02870	0.00665	0.26942	5.04034	0.0112	6.18E-15	0.0077	0.1078
	F12	Ave	7.97E-07	0.02764	0.05343	0.03088	1.79961	6.9915	2.87E-16	0.0346	8.18E-04
		Std	7.69E-07	0.05399	0.02073	0.04092	0.95114	4.4175	5.64E-16	0.0875	0.0010
	F13	Ave	0.029295	0.00732	0.65446	0.36222	8.89908	15.8757	3.66E-04	7.32E-04	0.0102
		Std	0.035271	0.01050	0.00447	0.30975	7.12624	16.1462	0.0020	0.0028	0.0103
l-dimensional)	F14	Ave	0.998004	3.84902	4.042493	0.998004	5.859838	1.1965	10.237	0.998004	1.9416
		Std	1.54E-16	3.24864	4.252799	4.23E-12	3.831299	0.5467	7.5445	5.83E-17	2.9633
	F15	Ave	0.002398	0.002434	0.00337	0.005206	0.003673	0.000886	0.0057	0.002374	3.00E-04
		Std	0.006097	0.006081	0.00625	0.007028	0.001647	0.000257	0.0121	0.0061	1.93E-19
	F16	Ave	-1.03162	-1.03162	-1.03163	-1.03162	-1.03163	-1.03163	-1.03162	-1.03162	-1.03162
		Std	6.04E-16	6.51E-16	2.13E-08	1.34E-06	4.88E-16	6.13E-14	6.77E-16	6.51E-16	1.00E-15
	F17	Ave	0.397887	0.397887	0.397889	0.397890	0.397887	0.397887	0.397887	0.397887	0.397887
		Std	0	0	2.13E-04	1.08E-05	0	3.41E-14	0	3.24E-16	0
	F18	Ave	3	3	3.000028	3.000002	3	3	8.4000	3	3

THANK YOU!