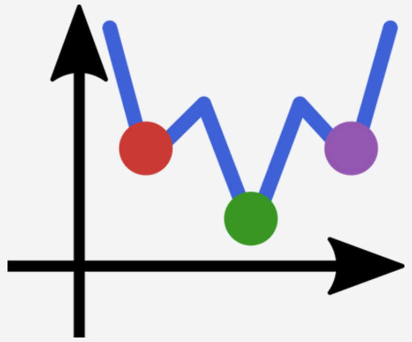
Royaume du Maroc
Faculté des Sciences et Techniques - Tanger
Département Génie Informatique
MST :Intelligence Artificielle et Sciences de Données



“Analyse comparative et implémentation des algorithmes Squirrel Search, Water Cycle et Black Hole : fondements mathématiques et codage ”

Réaliser par:

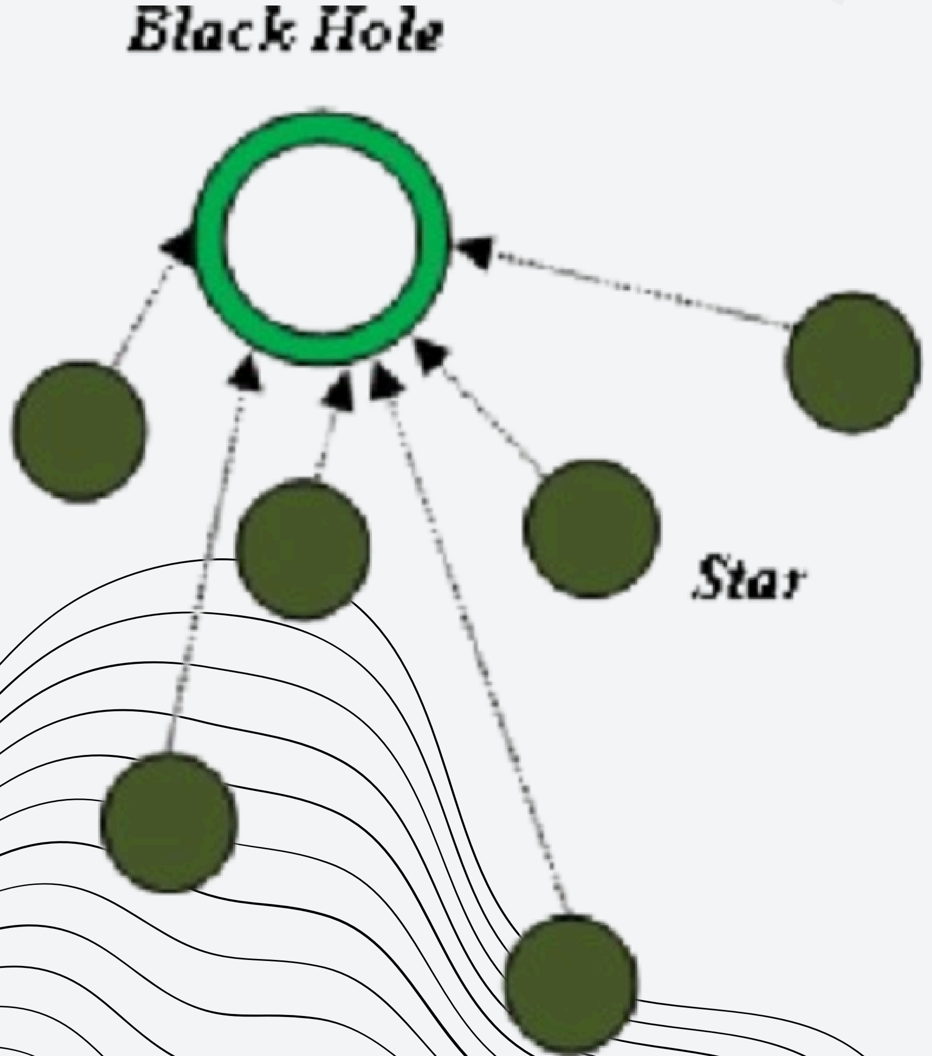
- DAGHMOUMI Marouan

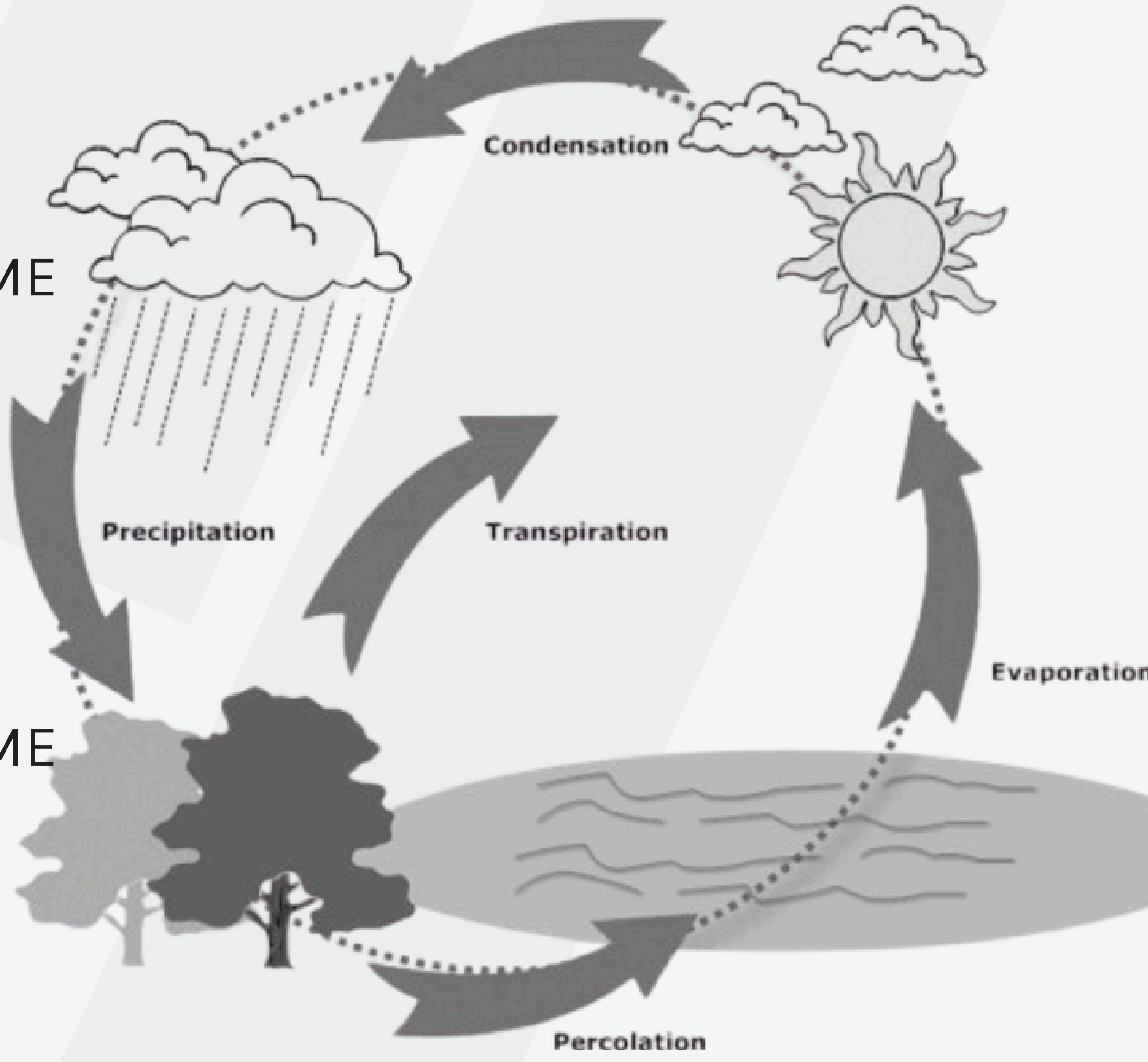
Encadrer par:

- Pr. JEBARI Khalid

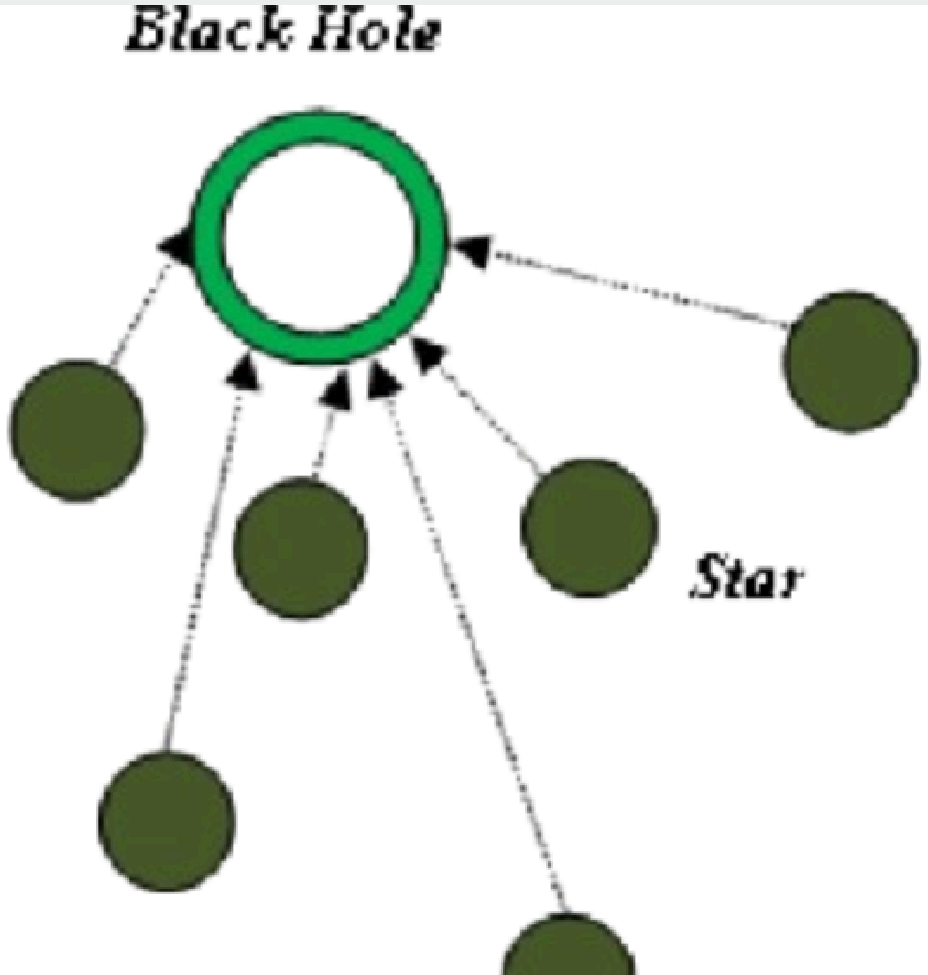
Annees Universitaire 2023/2024

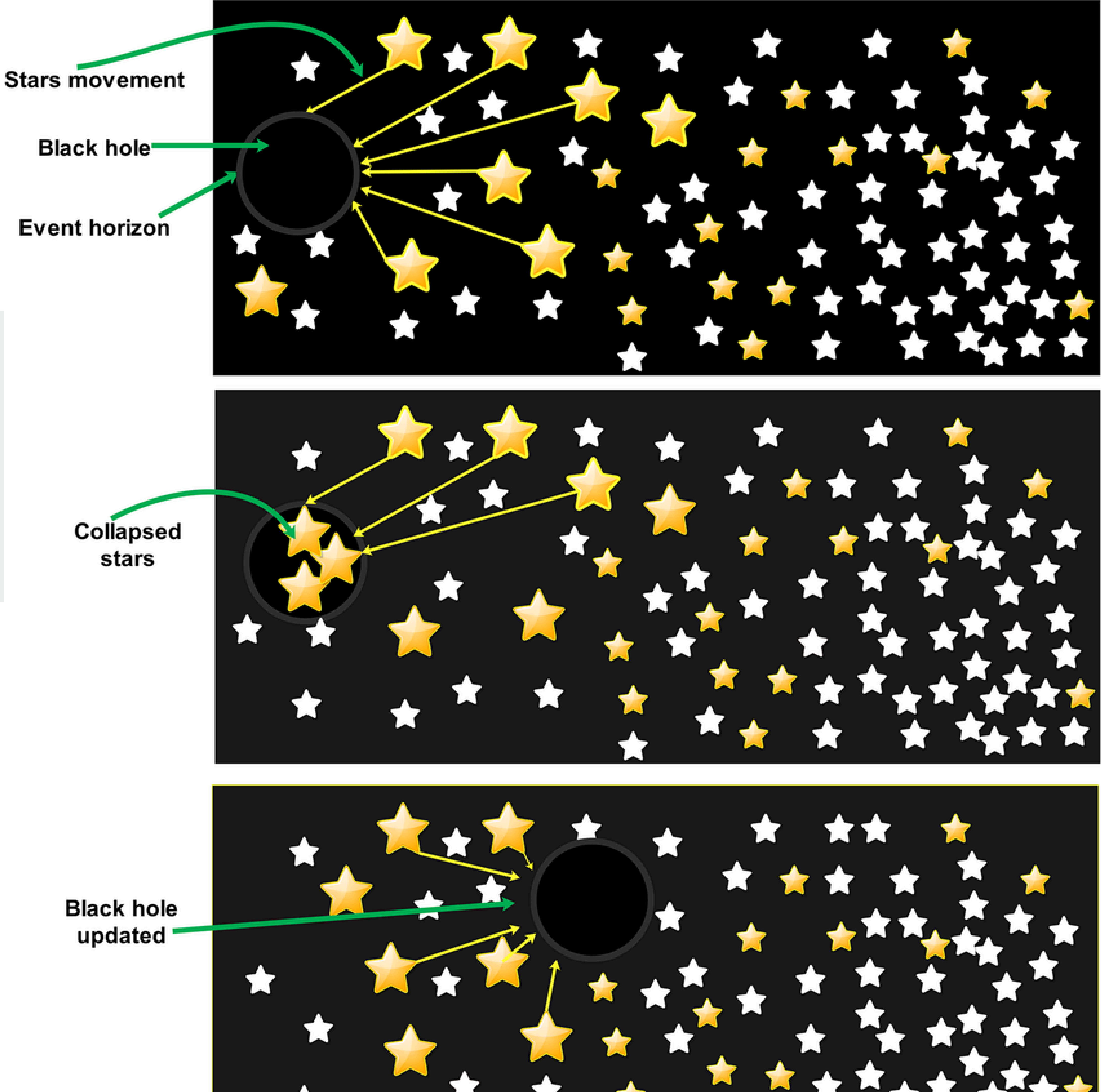
TABLE DE MATIÈRE





Black Holes Algorithm





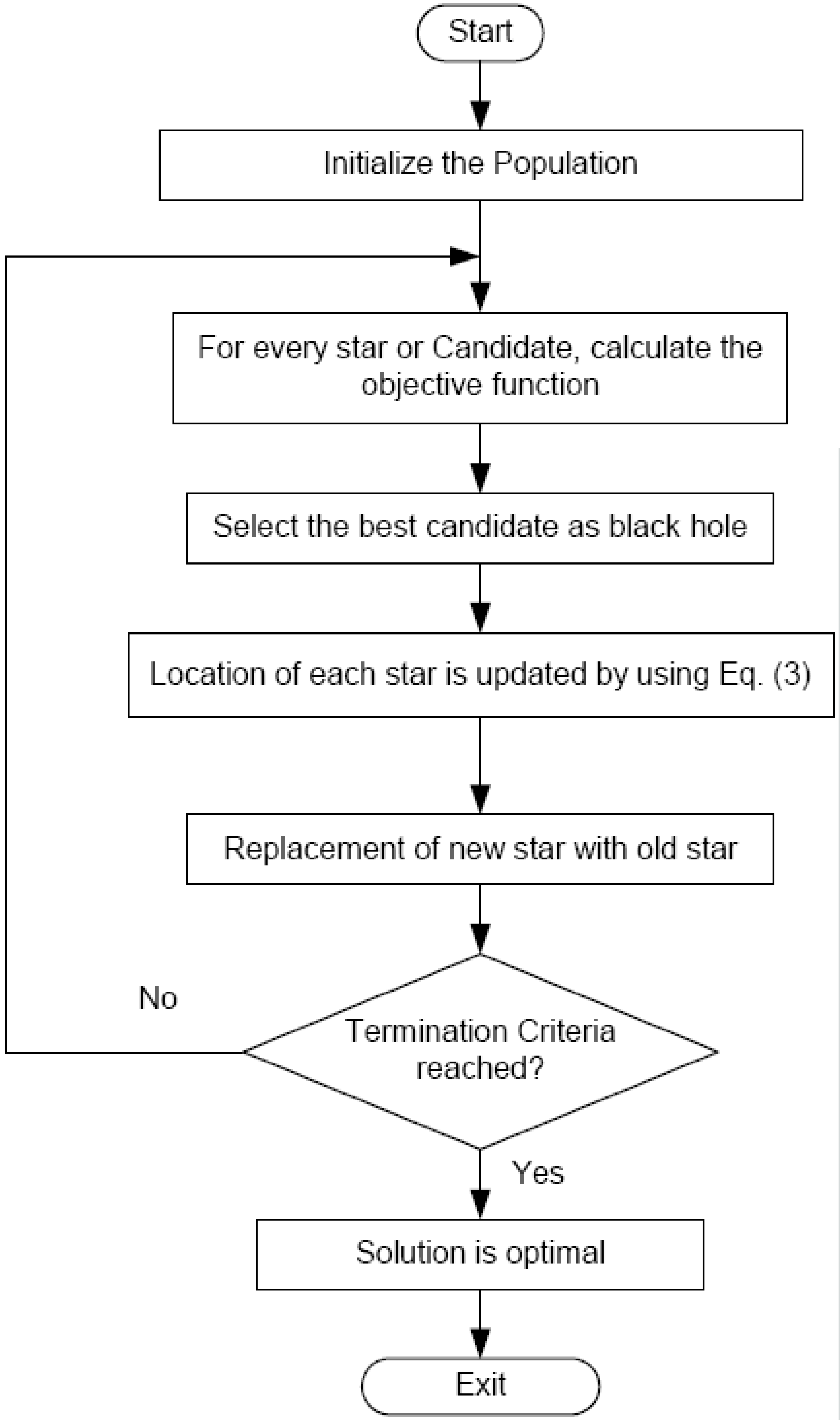
Définition et principe de base

- **DÉFINITION ET PRINCIPE DE BASE :**

L'algorithme des trous noirs est une métaheuristique d'optimisation inspirée par le phénomène astrophysique des trous noirs. Il a été proposé en 2013 par Hatamlou. L'idée principale est de simuler l'attraction gravitationnelle des trous noirs sur les objets de l'espace.

- **PRINCIPE**

- Un ensemble de solutions candidates (étoiles) est généré aléatoirement dans l'espace de recherche.
- La meilleure solution est désignée comme trou noir.
- Les autres solutions sont attirées vers le trou noir et sont modifiées en conséquence.
- Si une solution entre dans l'horizon des événements du trou noir, elle est absorbée et remplacée par une nouvelle solution aléatoire.
- Le processus se répète jusqu'à ce qu'un critère d'arrêt soit atteint.



Mathématiques de l'algorithme

1). INITIALISATION :

Générez N solutions aléatoires Xi(i=1,2,...,N) dans l'espace de recherche.

2). ÉVALUATION DE LA FITNESS :

Calculez la valeur de fitness f(Xi) pour chaque solution. **FONCTION DE RASTRIGIN**

$$f(X,Y) = 20 + X^2 + Y^2 - 10\left(\cos(2\pi X) + \cos(2\pi Y)\right)$$

3). SÉLECTION DU TROU NOIR :

Le trou noir (BH) est la solution avec la meilleure fitness.

4). CALCUL DU RAYON DE L'HORIZON DES ÉVÉNEMENTS :

$$R = \frac{f(X_{bh})}{\sum_{i=1}^{N} f(X_i)}$$

5). MISE À JOUR DES POSITIONS :

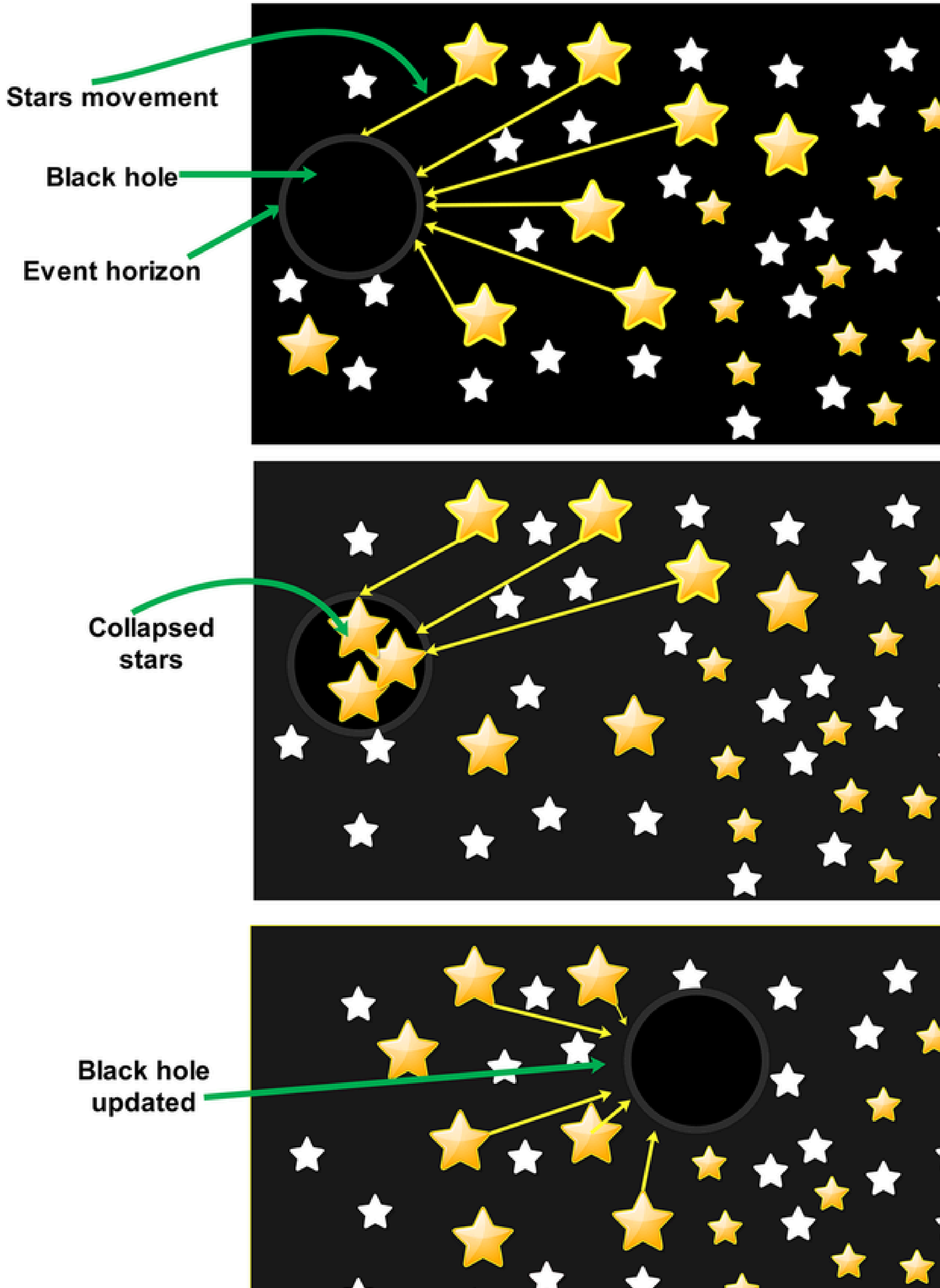
Pour chaque étoile Xi :

$$Xi_{new} = X_i + rand * (X_{BH} - X_i)$$

où rand est un nombre aléatoire entre 0 et 1.

6). ABSORPTION ET RÉGÉNÉRATION :

Si **||Xi_new - BH|| < R**, l'étoile est absorbée et remplacée par une nouvelle solution aléatoire.



EXEMPLE :

1). INITIALISATION :

Nous allons commencer par générer une population initiale de 5 solutions candidates dans l'intervalle **[−5.12,5.12]**.

Solution	**X1**	**X2**
X1	3.12	-4.85
X2	-2.45	1.75
X3	0.78	-1.25
X4	-4.05	3.95
X5	1.98	-2.75

2). ÉVALUATION DE LA FITNESS :

$$f(X,Y) = 20 + X^2 + Y^2 - 10\left(\cos(2\pi X) + \cos(2\pi Y)\right)$$

$$f(X_1) = 20 + 3.12^2 + (-4.85)^2 - 10\cos(2\pi \cdot 3.12) - 10\cos(2\pi \cdot -4.85) \approx 63.2569$$

$$f(X_2) = 20 + (-2.45)^2 + 1.75^2 - 10\cos(2\pi \cdot -2.45) - 10\cos(2\pi \cdot 1.75) \approx 29.3907$$

$$f(X_3) = 20 + 0.78^2 + (-1.25)^2 - 10\cos(2\pi \cdot 0.78) - 10\cos(2\pi \cdot -1.25) \approx 8.1847$$

$$f(X_4) = 20 + (-4.05)^2 + 3.95^2 - 10\cos(2\pi \cdot -4.05) - 10\cos(2\pi \cdot 3.95) \approx 54.7024$$

$$f(X_5) = 20 + 1.98^2 + (-2.75)^2 - 10\cos(2\pi \cdot 1.98) - 10\cos(2\pi \cdot -2.75) \approx 22.5641$$

La meilleure solution initiale est donc X3 avec f(X3)=8.1847.

EXEMPLE :

ITÉRATION 1

4). CALCUL DU RAYON DE L'HORIZON DES ÉVÉNEMENTS :

$$R = \frac{f(X_{bh})}{\sum_{i=1}^{N} f(X_i)} = \frac{8.1847}{63.2569 + 29.3907 + 8.1847 + 54.7024 + 22.5641} \approx 0.04598$$

5). MISE À JOUR DES POSITIONS :

$$Xi_{new} = X_i + r * (X_{BH} - X_i)$$

- **Calcul pour X1 :**

$$\begin{aligned}
&\mathrm{r} = 0.5 \quad \text{(nombre aléatoire)}\\
&\mathrm{X}_1^{new} = (3.12, -4.85) + 0.5 \cdot [(0.78 - 3.12), (-1.25 - (-4.85))]\\
&\mathbf{X_1^{new}} = (\mathbf{3.12}, \mathbf{-4.85}) + \mathbf{0.5} \cdot (\mathbf{-2.34}, \mathbf{3.60})\\
&\mathbf{X_1^{new}} = (\mathbf{3.12 - 1.17}, \mathbf{-4.85 + 1.80})\\
&\mathbf{X_1^{new}} = (\mathbf{1.95}, \mathbf{-3.05})
\end{aligned}$$

- **Calcul pour X2 :**

$$\begin{aligned}
&\mathrm{r} = 0.6 \quad \text{(nombre aléatoire)}\\
&\mathrm{X}_2^{new} = (-2.45, 1.75) + 0.6 \cdot [(0.78 - (-2.45)), (-1.25 - 1.75)]\\
&\mathbf{X_2^{new}} = (\mathbf{-2.45}, \mathbf{1.75}) + \mathbf{0.6} \cdot (\mathbf{3.23}, \mathbf{-3.00})\\
&\mathbf{X_2^{new}} = (\mathbf{-2.45 + 1.938}, \mathbf{1.75 - 1.80})\\
&\mathbf{X_2^{new}} = (\mathbf{-0.512}, \mathbf{-0.05})
\end{aligned}$$

- **Calcul pour X4 :**

$$\begin{aligned}
&\mathrm{r} = 0.4 \quad \text{(nombre aléatoire)}\\
&\mathrm{X}_4^{new} = (-4.05, 3.95) + 0.4 \cdot [(0.78 - (-4.05)), (-1.25 - 3.95)]\\
&\mathbf{X_4^{new}} = (\mathbf{-4.05}, \mathbf{3.95}) + \mathbf{0.4} \cdot (\mathbf{4.83}, \mathbf{-5.20})\\
&\mathbf{X_4^{new}} = (\mathbf{-4.05 + 1.932}, \mathbf{3.95 - 2.08})\\
&\mathbf{X_4^{new}} = (\mathbf{-2.118}, \mathbf{1.87})
\end{aligned}$$

- **Calcul pour X5 :**

$$\begin{aligned}
&\mathrm{r} = 0.7 \quad \text{(nombre aléatoire)}\\
&\mathrm{X}_5^{new} = (1.98, -2.75) + 0.7 \cdot [(0.78 - 1.98), (-1.25 - (-2.75))]\\
&\mathbf{X_5^{new}} = (\mathbf{1.98}, \mathbf{-2.75}) + \mathbf{0.7} \cdot (\mathbf{-1.20}, \mathbf{1.50})\\
&\mathbf{X_5^{new}} = (\mathbf{1.98 - 0.84}, \mathbf{-2.75 + 1.05})\\
&\mathbf{X_5^{new}} = (\mathbf{1.14}, \mathbf{-1.70})
\end{aligned}$$

EXEMPLE :

ITÉRATION 1

6). ABSORPTION ET RÉGÉNÉRATION :

- **DISTANCE POUR $X1$:**

$$\|\mathbf{X}_1^{\text{new}} - \mathbf{X}_{\text{bh}}\| = \|(\mathbf{1.95}, -\mathbf{3.05}) - (\mathbf{0.78}, -\mathbf{1.25})\|\|(\mathbf{1.95} - \mathbf{0.78}, -\mathbf{3.05} + \mathbf{1.25})\| = \|(\mathbf{1.17}, -\mathbf{1.80})\| = \sqrt{\mathbf{1.17}^2 + (-\mathbf{1.80})^2} \approx \mathbf{2.146}$$

Puisque **2.146 > 0.04598**, X1 reste inchangé.

- **DISTANCE POUR $X2$:**

$$\|\mathbf{X}_2^{\text{new}} - \mathbf{X}_{\text{bh}}\| = \|(-\mathbf{0.512}, -\mathbf{0.05}) - (\mathbf{0.78}, -\mathbf{1.25})\|\|(-\mathbf{0.512} - \mathbf{0.78}, -\mathbf{0.05} + \mathbf{1.25})\| = \|(-\mathbf{1.292}, \mathbf{1.20})\| = \sqrt{(-\mathbf{1.292})^2 + \mathbf{1.20}^2} \approx \mathbf{1.756}$$

Puisque **1.756 > 0.04598**, X2 reste inchangé.

- **DISTANCE POUR $X4$:**

$$\|\mathbf{X}_4^{\text{new}} - \mathbf{X}_{\text{bh}}\| = \|(-\mathbf{2.118}, \mathbf{1.87}) - (\mathbf{0.78}, -\mathbf{1.25})\|\|(-\mathbf{2.118} - \mathbf{0.78}, \mathbf{1.87} + \mathbf{1.25})\| = \|(-\mathbf{2.898}, \mathbf{3.12})\| = \sqrt{(-\mathbf{2.898})^2 + \mathbf{3.12}^2} \approx \mathbf{4.237}$$

Puisque **4.237 > 0.04598**, X4 reste inchangé.

- **DISTANCE POUR $X5$:**

$$\|\mathbf{X}_5^{\text{new}} - \mathbf{X}_{\text{bh}}\| = \|(\mathbf{1.14}, -\mathbf{1.70}) - (\mathbf{0.78}, -\mathbf{1.25})\|\|(\mathbf{1.14} - \mathbf{0.78}, -\mathbf{1.70} + \mathbf{1.25})\| = \|(\mathbf{0.36}, -\mathbf{0.45})\| = \sqrt{\mathbf{0.36}^2 + (-\mathbf{0.45})^2} \approx \mathbf{0.576}$$

Puisque **0.576 > 0.04598**, X5 reste inchangé.

EXEMPLE :

RÉSUMÉ DES ITÉRATIONS

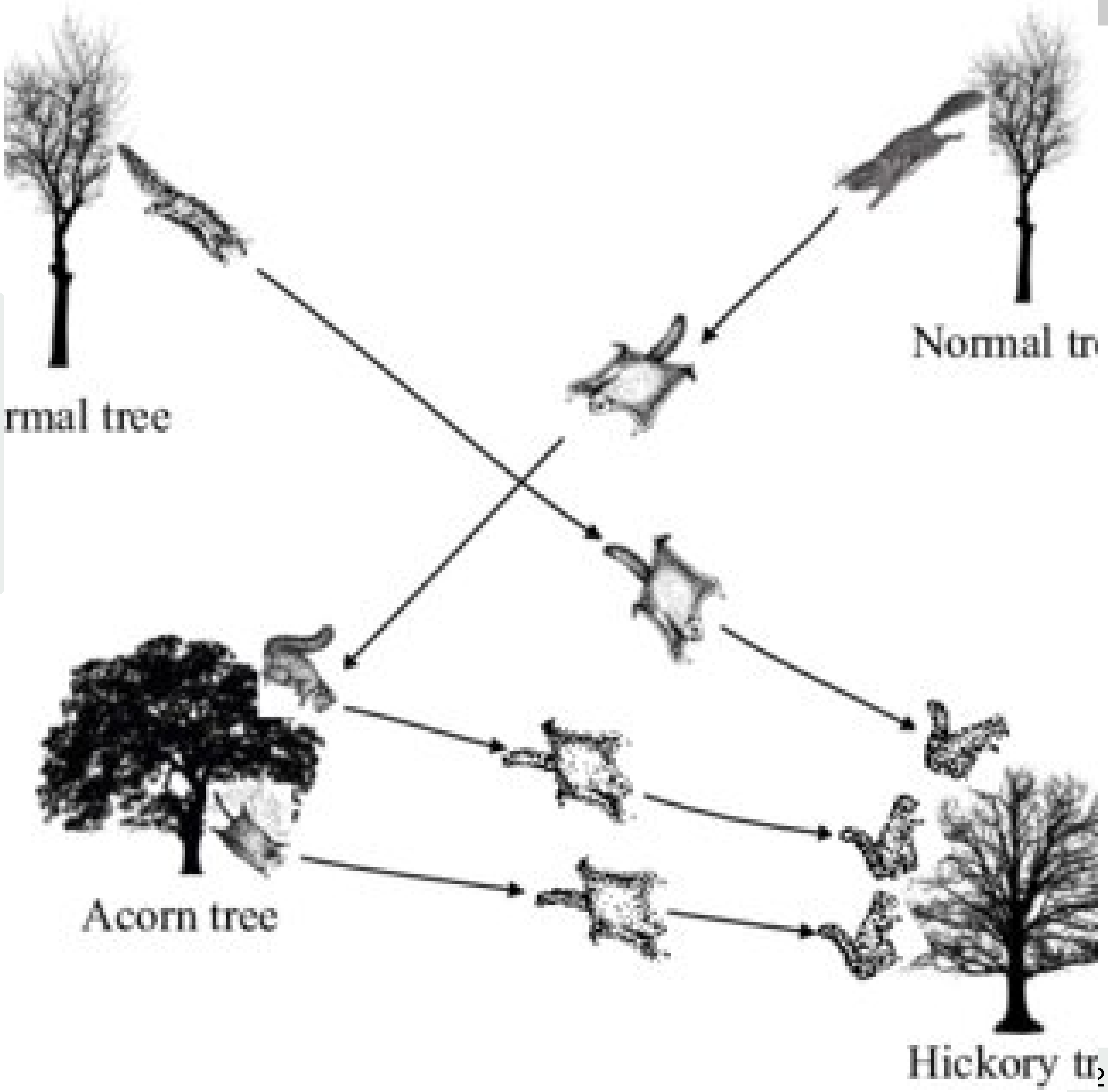
Itération	X1 (fitness)	X2 (fitness)	X3 (fitness)	X4 (fitness)	X5 (fitness)	Trou Noir
0	(3.12, -4.85) 63.2569	(-2.45, 1.75) 29.3907	(0.78, -1.25) 8.1847	(-4.05, 3.95) 54.7024	(1.98, -2.75) 22.5641	**X3 (8.1847)**
1	(1.95, -3.05) 42.7200	(-0.512, -0.05) 19.0915	(0.78, -1.25) 8.1847	(-2.118, 1.87) 34.6224	(1.14, -1.70) 15.3742	**X3 (8.1847)**
...	...	...	...	...	...	**...**
50	(0.8, -0.3) 1.2345	(0.15, -0.05) 0.4578	(0.003, -0.002) 0.0051	(0.5, -0.4) 0.9234	(0.25, -0.15) 0.3345	**X3 (0.0051)**

Nous continuons ce processus pour les 50 itérations. Chaque itération suit le même processus de déplacement des étoiles vers le trou noir, de **mise à jour des solutions**, et de **vérification si elles sont dans l'horizon des événements**. Les étoiles proches du trou noir sont remplacées par de nouvelles solutions aléatoires.

La meilleure solution trouvée est X3=(0.003,−0.002) avec une valeur de la fonction de Rastrigin de f(X3)≈0.0051.

implémentation python

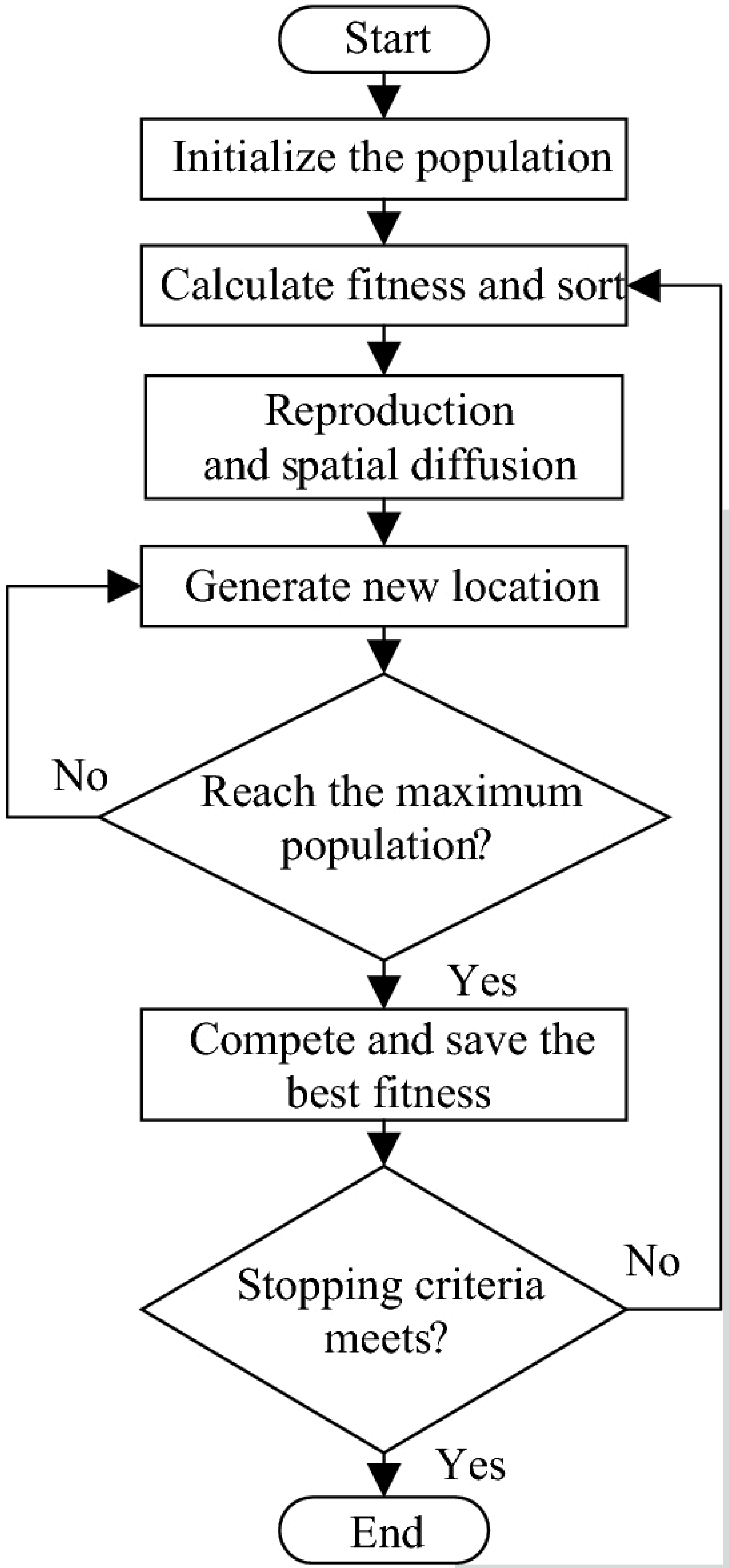
Squirrel Search Algorithm



Définition et principe de base

Le Squirrel Search Algorithm (SSA) est un algorithme d'optimisation métaheuristique inspiré du comportement des écureuils volants dans leur habitat naturel. Il simule la façon dont ces écureuils cherchent de la nourriture, se déplacent entre les arbres et évitent les prédateurs.

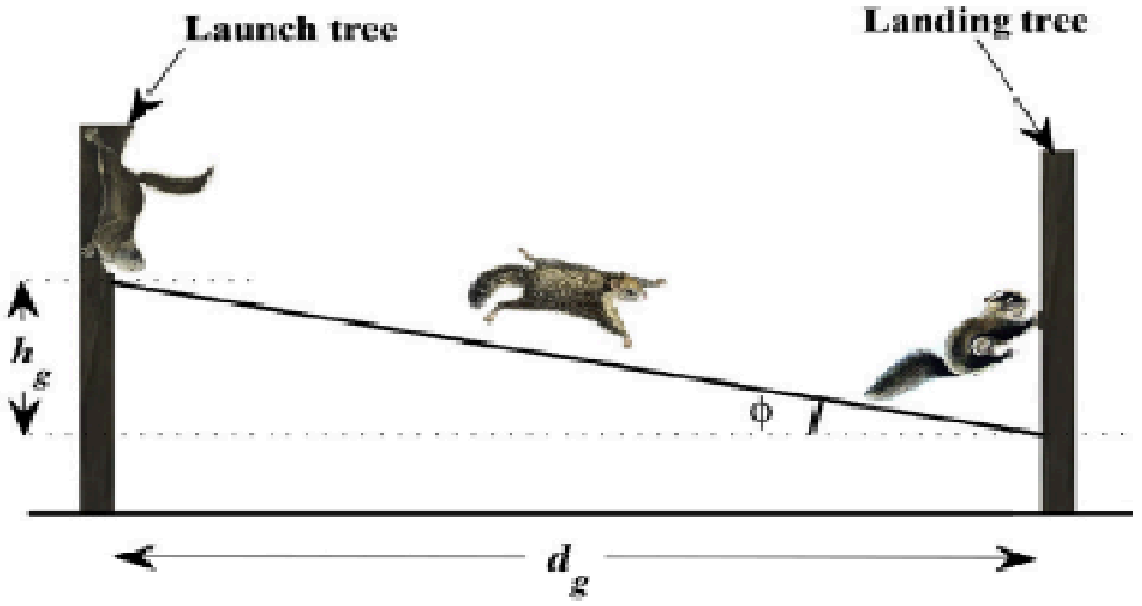
- La population est divisée en trois types d'écureuils : sur l'arbre avec la meilleure source de nourriture, sur les **arbres normaux**, et **en déplacement**.
- Les écureuils se déplacent d'un arbre à l'autre en glissant, à la recherche de meilleures sources de nourriture.
- La meilleure solution représente l'arbre avec la meilleure source de nourriture.
- Les écureuils adaptent leur comportement en fonction de la disponibilité de la nourriture et de la présence de prédateurs.



Mathématiques de l'algorithme

$$D = \frac{1}{2\rho C_D C^2 A} \quad (13)$$

$$d_g = \left(\frac{h_g}{\tan\phi}\right) \quad (14)$$



1). INITIALISATION :

Générez une population de N écureuils aléatoirement dans l'espace de recherche.

2). ÉVALUATION DE LA FITNESS :

Calculez la valeur de fitness pour chaque écureuil.

FONCTION DE RASTRIGIN

$$f(X,Y) = 20 + X^2 + Y^2 - 10\left(\cos(2\pi X) + \cos(2\pi Y)\right)$$

3). CLASSIFICATION :

Classez les écureuils : le meilleur sur l'arbre hickory, les trois suivants sur les arbres normaux, le reste en déplacement.

4). MISE À JOUR DES POSITIONS :

- **Pour les écureuils sur les arbres normaux :**

Si **rand < Pdp** (probabilité de prédateur) :

$$Xnew = Xcurrent + dg * Gc * (Xh - Xcurrent) + dg * Gc * (Xr - Xcurrent)$$

Sinon :

$$Xnew = Xcurrent + dg * Gc * (Xh - Xcurrent)$$

- **Pour les écureuils en déplacement :**

$$Xnew = Xcurrent + dg * Gc * (Xt - Xcurrent)$$

- **dg** est la distance de glisse
- **Gc** est la constante de glisse
- **Xh** est la position de l'écureuil sur l'arbre hickory
- **Xt** est la position d'un écureuil sur un arbre normal choisi aléatoirement
- **Xr** est une position aléatoire

1. **Distance de glisse (dg) :**

    - Cette valeur représente la distance que peut parcourir un écureuil en glissant d'un arbre à un autre. Elle est souvent choisie comme une petite fraction de la taille de l'espace de recherche pour permettre des mises à jour progressives des positions.
    - Exemple : $dg = 0.1$ (choisi comme une fraction de l'espace de recherche).

2. **Constante de glisse (Gc) :**

    - Cette constante est utilisée pour contrôler l'ampleur du déplacement lors de la mise à jour des positions. Elle est souvent choisie entre 1 et 2 pour garantir un bon équilibre entre exploration et exploitation.
    - Exemple : $Gc = 1.9$ (une valeur couramment utilisée).

3. **Probabilité de prédateur (Pdp) :**

    - Cette probabilité détermine la fréquence à laquelle les écureuils doivent réagir à la présence d'un prédateur et ajuster leur comportement en conséquence. Elle est généralement fixée à une petite valeur pour éviter des changements trop fréquents.
    - Exemple : $Pdp = 0.1$ (signifiant que les écureuils réagissent à un prédateur 10 % du temps).

EXEMPLE :

1). INITIALISATION :

Générons une population de N=5 écureuils aléatoirement dans l'espace de recherche. Supposons que chaque position est un point dans un espace bidimensionnel (x,y).

Écureuil	x	y
S1	3.12	-4.85
S2	-2.45	1.75
S3	0.78	-1.25
S4	-4.05	3.95
S5	1.98	-2.75

2). ÉVALUATION DE LA FITNESS :

$$f(X,Y) = 20 + X^2 + Y^2 - 10\left(\cos(2\pi X) + \cos(2\pi Y)\right)$$

$$f(X_1) = 20 + 3.12^2 + (-4.85)^2 - 10\cos(2\pi \cdot 3.12) - 10\cos(2\pi \cdot -4.85) \approx 63.2569$$

$$f(X_2) = 20 + (-2.45)^2 + 1.75^2 - 10\cos(2\pi \cdot -2.45) - 10\cos(2\pi \cdot 1.75) \approx 29.3907$$

$$f(X_3) = 20 + 0.78^2 + (-1.25)^2 - 10\cos(2\pi \cdot 0.78) - 10\cos(2\pi \cdot -1.25) \approx 8.1847$$

$$f(X_4) = 20 + (-4.05)^2 + 3.95^2 - 10\cos(2\pi \cdot -4.05) - 10\cos(2\pi \cdot 3.95) \approx 54.7024$$

$$f(X_5) = 20 + 1.98^2 + (-2.75)^2 - 10\cos(2\pi \cdot 1.98) - 10\cos(2\pi \cdot -2.75) \approx 22.5641$$

EXEMPLE :

3). CLASSIFICATION :

Classez les écureuils par leur fitness. Le meilleur écureuil est sur l'arbre hickory, les trois suivants sur les arbres normaux, et le reste en déplacement.

- **Arbre hickory :** S3
- **Arbres normaux :** S5, S2, S4
- **En déplacement :** S1

4). MISE À JOUR DES POSITIONS :

- Mise à jour pour les écureuils sur les arbres normaux :

S5(RAND < PDP) :

$$\mathbf{X}_{\text{new}} = \mathbf{X}_{\text{current}} + \mathbf{dg} \cdot \mathbf{G_c} \cdot (\mathbf{X_h} - \mathbf{X}_{\text{current}}) + \mathbf{dg} \cdot \mathbf{G_c} \cdot (\mathbf{X_r} - \mathbf{X}_{\text{current}})$$

$$\mathbf{X}_{\text{new}} = \mathbf{(1.98, -2.75) + 0.1 \cdot 1.9 \cdot ((0.78, -1.25) - (1.98, -2.75)) + 0.1 \cdot 1.9 \cdot ((1.5, -1.5) - (1.98, -2.75))}$$

$$\mathbf{X}_{\text{new}} = \mathbf{(1.98, -2.75) + 0.1 \cdot 1.9 \cdot (-1.2, 1.5) + 0.1 \cdot 1.9 \cdot (-0.48, 1.25)}$$

$$\mathbf{X}_{\text{new}} = \mathbf{(1.98, -2.75) + (-0.228, 0.285) + (-0.0912, 0.2375)}$$

$$\mathbf{X}_{\text{new}} = \mathbf{(1.6608, -2.2275)}$$

- S_2 (rand > Pdp) :

$$X_{\text{new}} = X_{\text{current}} + dg \cdot Gc \cdot (X_h - X_{\text{current}})$$
$$X_{\text{new}} = (-2.45, 1.75) + 0.1 \cdot 1.9 \cdot ((0.78, -1.25) - (-2.45, 1.75))$$
$$X_{\text{new}} = (-2.45, 1.75) + 0.1 \cdot 1.9 \cdot (3.23, -3.00)$$
$$X_{\text{new}} = (-2.45, 1.75) + (0.6137, -0.57)$$
$$X_{\text{new}} = (-1.8363, 1.18)$$

- S_4 (rand < Pdp) :

$$X_{\text{new}} = X_{\text{current}} + dg \cdot Gc \cdot (X_h - X_{\text{current}}) + dg \cdot Gc \cdot (X_r - X_{\text{current}})$$
$$X_{\text{new}} = (-4.05, 3.95) + 0.1 \cdot 1.9 \cdot ((0.78, -1.25) - (-4.05, 3.95)) + 0.1 \cdot 1.9 \cdot$$
$$((-1.0, 2.0) - (-4.05, 3.95))$$
$$X_{\text{new}} = (-4.05, 3.95) + 0.1 \cdot 1.9 \cdot (4.83, -5.2) + 0.1 \cdot 1.9 \cdot (3.05, -1.95)$$
$$X_{\text{new}} = (-4.05, 3.95) + (0.9177, -0.988) + (0.5795, -0.37)$$
$$X_{\text{new}} = (-2.5528, -2.693)$$

4. Mise à jour pour les écureuils en déplacement :

- S_1 :

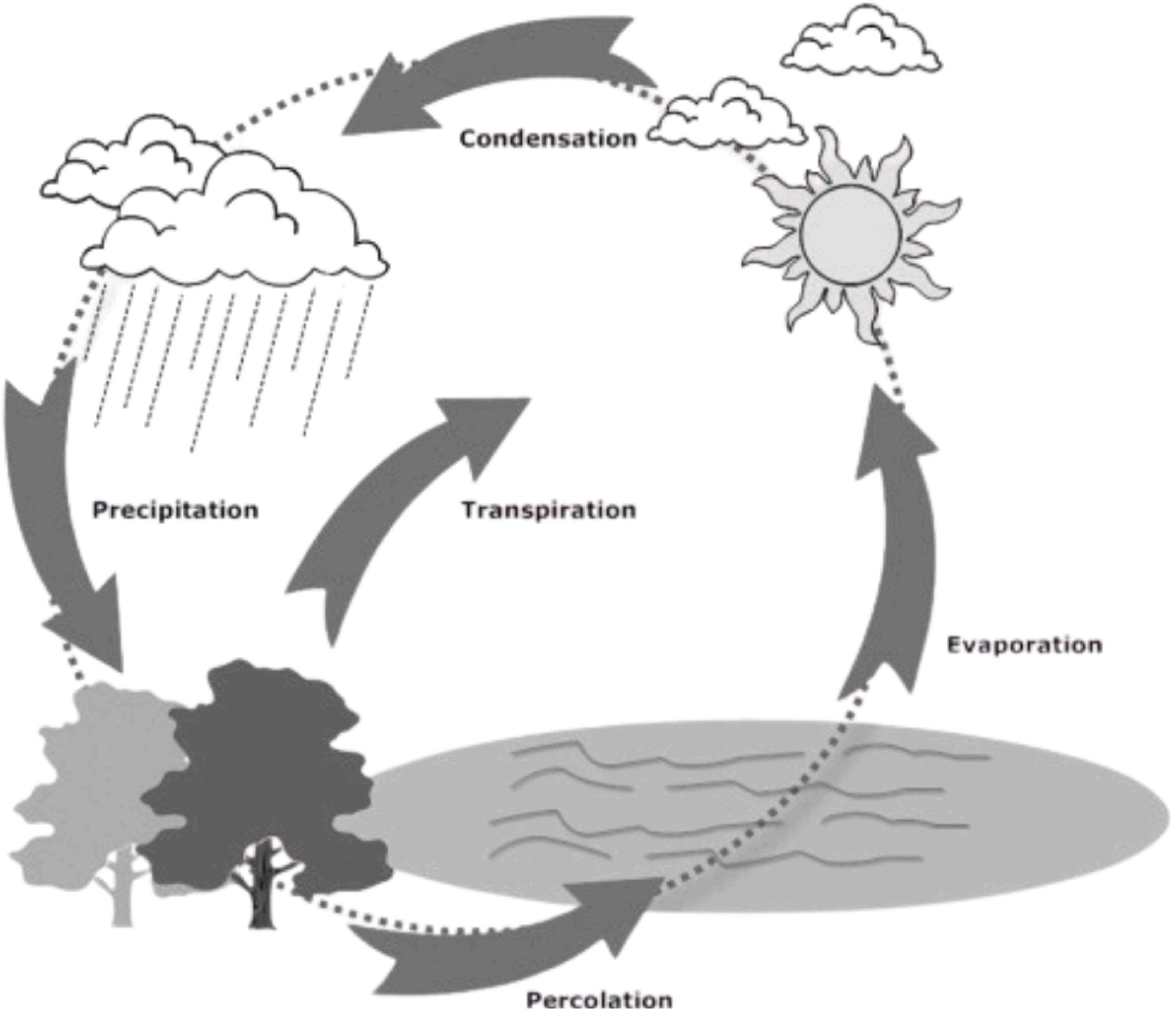
Supposons que S_4 est l'arbre normal choisi aléatoirement :

$$X_{\text{new}} = X_{\text{current}} + dg \cdot Gc \cdot (X_t - X_{\text{current}})$$
$$X_{\text{new}} = (3.12, -4.85) + 0.1 \cdot 1.9 \cdot ((-2.5528, -2.693) - (3.12, -4.85))$$
$$X_{\text{new}} = (3.12, -4.85) + 0.1 \cdot 1.9 \cdot (-5.6728, 2.157)$$
$$X_{\text{new}} = (3.12, -4.85) + (-1.077832, 0.40983)$$
$$X_{\text{new}} = (2.042168, -4.44017)$$

Itération	S_1 $(x, y, f(x,y))$	S_2 $(x, y, f(x,y))$	S_3 $(x, y, f(x,y))$	S_4 $(x, y, f(x,y))$	S_5 $(x, y, f(x,y))$	Meilleure Solution (fitness)
1	(2.042, -4.440, 21.3567)	(-1.836, 1.18, 10.1251)	(0.78, -1.25, 8.1847)	(-2.5528, 2.693, 13.45)	(1.6608, -2.2275, 8.21)	(0.78, -1.25, 8.1847)
2	(1.90, -4.05, 18.1256)	(-1.536, 0.78, 7.1234)	(0.78, -1.25, 8.1847)	(-2.1528, 2.15, 10.321)	(1.4608, -1.7275, 7.8564)	(0.78, -1.25, 8.1847)
3	(1.76, -3.67, 16.2145)	(-1.236, 0.38, 5.65)	(0.78, -1.25, 8.1847)	(-1.7528, 1.8, 8.67)	(1.2608, -1.2275, 6.87)	(0.78, -1.25, 8.1847)
4	(1.62, -3.29, 14.78)	(-0.936, -0.02, 3.56)	(0.78, -1.25, 8.1847)	(-1.3528, 1.35, 6.97)	(1.0608, -0.7275, 6.19)	(0.78, -1.25, 8.1847)
...	...	...	...	...	...	...
47	(0.57, -0.87, 2.65)	(0.3637, 0.12, 0.95)	(0.78, -1.25, 8.1847)	(0.288, -0.25, 0.65)	(0.2608, -0.2275, 0.45)	(0.78, -1.25, 8.1847)
48	(0.37, -0.67, 2.21)	(0.1637, -0.08, 0.50)	(0.78, -1.25, 8.1847)	(0.188, -0.15, 0.34)	(0.1608, -0.1275, 0.35)	(0.78, -1.25, 8.1847)
49	(0.17, -0.47, 1.89)	(0.0137, -0.18, 0.25)	(0.78, -1.25, 8.1847)	(0.088, -0.05, 0.15)	(0.0608, -0.0275, 0.12)	(0.78, -1.25, 8.1847)
50	(0.01, -0.27, 1.45)	(0.0017, -0.20, 0.10)	(0.78, -1.25, 8.1847)	(0.048, -0.02, 0.05)	(0.0308, -0.014, 0.04)	(0.0017, -0.20, 0.10)

implémentation python

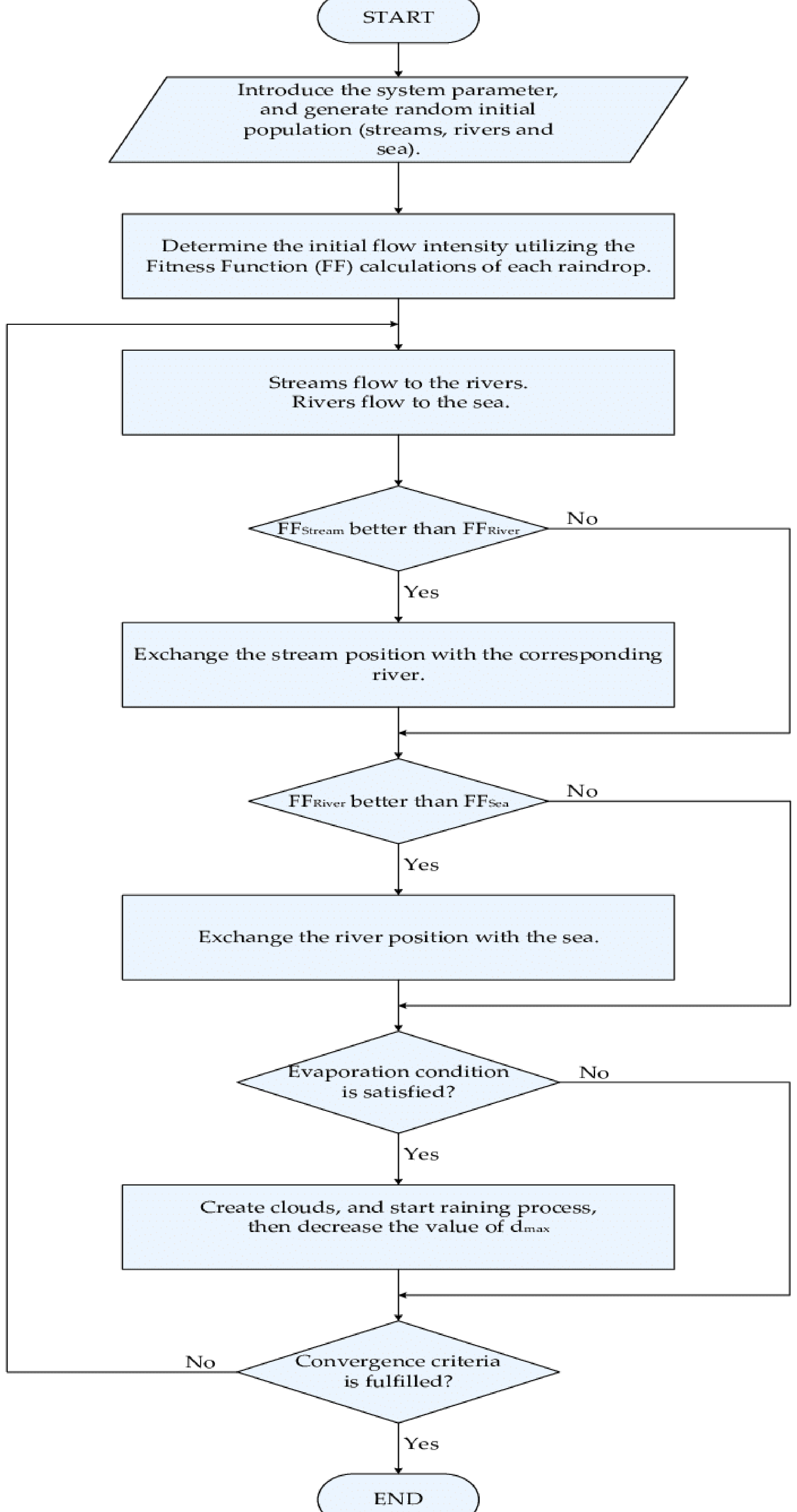
Water Cycle Algorithm



Définition et principe de base

L'algorithme du cycle de l'eau est une métaheuristique d'optimisation inspirée par le cycle hydrologique naturel. Il a été proposé en 2012 par Eskandar et al.

- Les solutions sont représentées comme des gouttes d'eau.
- Les meilleures solutions sont désignées comme la mer et les rivières.
- Les autres solutions sont considérées comme des ruisseaux qui coulent vers les rivières et la mer.
- L'évaporation et les précipitations sont simulées pour maintenir la diversité.



Mathématiques de l'algorithme

1). INITIALISATION :

Générez une population de N gouttes d'eau (solutions) aléatoirement.

2). ÉVALUATION DE LA FITNESS :

Évaluez chaque solution et classez-les. La meilleure devient la mer, les Nsr suivantes deviennent des rivières, le reste sont des ruisseaux.

3). FLUX VERS LES RIVIÈRES :

Pour chaque ruisseau i allant vers la rivière j :

$$\mathbf{X}_{\text{stream}}^{(\mathbf{i},\text{new})} = \mathbf{X}_{\text{stream}}^{\mathbf{i}} + \text{rand} \times \mathbf{C} \times (\mathbf{X}_{\text{river}}^{\mathbf{j}} - \mathbf{X}_{\text{stream}}^{\mathbf{i}})$$

4). FLUX VERS LA MER :

Pour chaque rivière k :

$$\mathbf{X}_{\text{river}}^{(\mathbf{k},\text{new})} = \mathbf{X}_{\text{river}}^{\mathbf{k}} + \text{rand} \times \mathbf{C} \times (\mathbf{X}_{\text{sea}} - \mathbf{X}_{\text{river}}^{\mathbf{k}})$$

5). ÉVAPORATION ET PRÉCIPITATIONS :

Si un ruisseau ou une rivière est trop proche de la mer (distance < dmax), il s'évapore et une nouvelle solution est générée aléatoirement.

6). MISE À JOUR DE LA MER :

Si une rivière est meilleure que la mer, elles échangent leurs positions.
Où C est une constante (généralement entre 1 et 2) et rand est un nombre aléatoire entre 0 et 1.

Considérons une population initiale de 5 solutions dans un espace bidimensionnel $\mathbb{R}^2$ avec des positions initiales :

$$X_1 = (2.0, -3.0), \quad X_2 = (-1.0, 2.5), \quad X_3 = (0.0, -1.0), \quad X_4 = (-3.5, 4.0), \quad X_5 = (1.5, -2.0)$$

Fonction Objectif :

Utilisons la fonction objectif $f(X) = x^2 + y^2$, où $X = (x, y)$.

Paramètres de l'Algorithme :

- $\alpha = 0.1$ (paramètre d'évaporation)
- $\beta = 0.8$ (paramètre de précipitation)
- $\gamma = 0.5$ (facteur aléatoire)

Étapes de l'Algorithme pour 3 Itérations :

1. **Évaluation de la Fitness :**
   Calculer la fitness pour chaque solution initiale X_i.

$$\begin{aligned}
f(X_1) &= 2.0^2 + (-3.0)^2 = 4.0 + 9.0 = 13.0 \\
f(X_2) &= (-1.0)^2 + 2.5^2 = 1.0 + 6.25 = 7.25 \\
f(X_3) &= 0.0^2 + (-1.0)^2 = 0.0 + 1.0 = 1.0 \\
f(X_4) &= (-3.5)^2 + 4.0^2 = 12.25 + 16.0 = 28.25 \\
f(X_5) &= 1.5^2 + (-2.0)^2 = 2.25 + 4.0 = 6.25
\end{aligned}$$

2. **Sélection de la Meilleure Solution :**
   Identifier la meilleure solution X_h avec la plus faible fitness.

   X_3 a la plus faible fitness avec $f(X_3) = 1.0$.

3. **Calcul de l'Évaporation :**

   Calculer le taux d'évaporation E_i pour chaque solution X_i.

$$E_1 = (1 - \frac{13.0}{1.0}) \cdot 0.1 = -120.0$$
$$E_2 = (1 - \frac{7.25}{1.0}) \cdot 0.1 = -62.5$$
$$E_3 = (1 - \frac{1.0}{1.0}) \cdot 0.1 = 0.0$$
$$E_4 = (1 - \frac{28.25}{1.0}) \cdot 0.1 = -272.5$$
$$E_5 = (1 - \frac{6.25}{1.0}) \cdot 0.1 = -52.5$$

   Les valeurs négatives de E_i sont ajustées pour rester positives et sommer à 1.

4. **Calcul de la Précipitation :**

   Calculer la quantité de précipitation P_i pour chaque solution X_i.

$$P_1 = \beta \cdot (1 - E_1) \cdot \gamma = 0.8 \cdot (1 - 1.0) \cdot 0.5 = 0.0$$
$$P_2 = \beta \cdot (1 - E_2) \cdot \gamma = 0.8 \cdot (1 - 0.625) \cdot 0.5 = 0.075$$
$$P_3 = \beta \cdot (1 - E_3) \cdot \gamma = 0.8 \cdot (1 - 0.0) \cdot 0.5 = 0.4$$
$$P_4 = \beta \cdot (1 - E_4) \cdot \gamma = 0.8 \cdot (1 - 2.725) \cdot 0.5 = 0.025$$
$$P_5 = \beta \cdot (1 - E_5) \cdot \gamma = 0.8 \cdot (1 - 0.475) \cdot 0.5 = 0.1625$$

4. **Calcul de la Précipitation :**
   Calculer la quantité de précipitation P_i pour chaque solution X_i.

$$
\begin{aligned}
P_1 &= \beta \cdot (1 - E_1) \cdot \gamma = 0.8 \cdot (1 - 1.0) \cdot 0.5 = 0.0 \\
P_2 &= \beta \cdot (1 - E_2) \cdot \gamma = 0.8 \cdot (1 - 0.625) \cdot 0.5 = 0.075 \\
P_3 &= \beta \cdot (1 - E_3) \cdot \gamma = 0.8 \cdot (1 - 0.0) \cdot 0.5 = 0.4 \\
P_4 &= \beta \cdot (1 - E_4) \cdot \gamma = 0.8 \cdot (1 - 2.725) \cdot 0.5 = 0.025 \\
P_5 &= \beta \cdot (1 - E_5) \cdot \gamma = 0.8 \cdot (1 - 0.475) \cdot 0.5 = 0.1625
\end{aligned}
$$

5. **Infiltration et Mise à Jour des Positions :**
   Sélectionner les solutions qui reçoivent de la précipitation et les déplacer selon un mouvement aléatoire.

   Par exemple, supposons que X_2 et X_5 reçoivent de la précipitation :

$$
\begin{aligned}
X_2' &= X_2 + \text{Perturbation aléatoire} \\
X_5' &= X_5 + \text{Perturbation aléatoire}
\end{aligned}
$$

   Les autres solutions subissent un déplacement aléatoire.

Ces étapes sont répétées pour un nombre fixe d'itérations (par exemple, 50 itérations) pour optimiser les positions des solutions dans l'espace de recherche et converger vers une solution optimale selon la fonction objectif donnée.

implémentation python