0.0.1 Re-framed Monarch Butterfly Algorithm (Re-framed MBO)

- **Objectives**:

Objective Problem	
$f(\mathbf{x}_i)$	fitness of $\mathbf{x}_i$.
n	the dimension of the search space.
$[lb_{\mathbf{x}}, ub_{\mathbf{x}}]$	the interval of objective variable $\mathbf{x}$, in our cases, it is defined in the IOHprofiler, $[lb_{\mathbf{x}}, ub_{\mathbf{x}}] = [-5, +5]$.
Objective Solution	
$\mathbf{x}_i$	it can be imagined as one individual in Swarm-Intelligence Algorithms, $\mathbf{x}_i \in R^n$.

- **Parameters**:

T	the maximum iterations, the budget in our cases, in our case, it is defined in IOHprofiler.
M	population size, $M = 50$.
$\mathbf{x}_g$	the best position that the whole population has found so far.
z	weighting value of individuals.
w_1	the ratio of stronger population to weaker population, specifically named 'partition', $w_1 = 5/12$.
w_2	the migrating rate, specifically named 'period', $w_2 = 1.2$.
w_3	the adjusting rate, specifically named 'BAR', the adjusting rate, $w_3 = 5/12$.
w_4	the maximum one step size, $w_4 = 1$.
w_5	the number of elitists, $w_5 = 2$, $w_5 \in [0, M]$.

- **Functions**:
  - Initialization Process:
    (1) Initialize $\mathbf{x}_i(t=0)$:
    $$\mathbf{x}_i(t=0) = \boldsymbol{\mathcal{U}}(lb_{\mathbf{x}}, ub_{\mathbf{x}}), i = 1 \dots M \quad (1)$$
    (2) Initialize $\mathbf{x}_g(t=0)$:
    $$\mathbf{x}_g(t=0) = \mathbf{Min}(\{\mathbf{x}_i(t)\}), i = 1 \dots M \quad (2)$$
    (3) Initialize $z(t=0)$:
    $$z(t=0) = \frac{w_4}{t^2} \quad (3)$$
  - Optimization Process:
    (1) Optimize $\hat{\mathbf{x}}_i(t+1)$:
    $$\begin{aligned}
\langle \mathbf{x}_i(t) \rangle &= \mathbf{Sort}(\{\mathbf{x}_i(t)\}), i = 1 \dots M \\
{}^{strong}\hat{\mathbf{x}}_{i,n}(t+1) &= \begin{cases} \mathbf{x}_{j,n}(t) \in \langle \mathbf{x}_i(t) \rangle, j \in [1, M'] & , \ r \times w_2 \leqslant w_1 \\ \mathbf{x}_{j,n}(t) \in \langle \mathbf{x}_i(t) \rangle, j \in (M', M] & , \ \text{o.w.} \end{cases} \\
M' &= \lceil w1 \times M \rceil \\
{}^{weak}\hat{\mathbf{x}}_{i,n}(t+1) &= \begin{cases} \mathbf{x}_{g,n}(t), r \geqslant w_1 \\ \begin{cases} \mathbf{x}_{j,n}(t) + z(t) \times \left(\textbf{Lévy}_{i,n} - 0.5\right), j \in (M', M], r > w_3 \\ \mathbf{x}_{j,n}(t) \in \langle \mathbf{x}_i(t) \rangle, j \in (M', M], \text{o.w.} \end{cases} & , \text{o.w.} \end{cases} \\
\{\hat{\mathbf{x}}_i(t+1)\} &= \{{}^{strong}\hat{\mathbf{x}}_i(t+1)\} \cup \{{}^{weak}\hat{\mathbf{x}}_i(t+1)\}
\end{aligned} \quad (4)$$
    where $M' = \lceil w1 \times M \rceil$, $\mathbf{Lévy}_{i,n} = \mathbf{Lévy}(d, n, T)$ with $d \sim Exp(2 \times T)$.
    (2) Dealing with outliers C:
    $$\mathbf{x}_{i,n}^{\text{fixed}}(t+1) = \begin{cases} ub_x & , \ \mathbf{x}_{i,n}(t+1) > ub_{\mathbf{x}} \\ \mathbf{x}_{i,n}(t+1) & , \ \text{o.w} \\ lb_x & , \ \mathbf{x}_{i,n}(t+1) < lb_{\mathbf{x}} \end{cases} \quad (5)$$
    (3) Select $\mathbf{x}_i(t+1)$ from $\hat{\mathbf{x}}_i(t+1)$:
    $$\begin{aligned} &\langle \hat{\mathbf{x}}_i(t+1) \rangle = \mathbf{Sort}(\{\hat{\mathbf{x}}_i(t+1)\}), i = 1 \dots M \\ &\mathbf{x}_i(t+1) \in \{\langle \hat{\mathbf{x}}_i(t+1) \rangle, i = 1 \dots M - w_5\} \cup \{\langle \mathbf{x}_i(t) \rangle, i = 1 \dots w_5\} \end{aligned} \quad (6)$$

(4) Update $\mathbf{x}_g(t)$ to generate $\mathbf{x}_g(t+1)$:

$$\mathbf{x}_g(t+1) = \mathbf{Min}(\mathbf{x}_g(t) \cup \{\mathbf{x}_i(t+1)\}), i = 1 \ldots M \tag{7}$$

(5) Update $z(t)$ to generate $z(t+1)$:

$$z(t+1) = \frac{w_4}{(t+1)^2} \tag{8}$$

Algorithm 1 Re-framed MBO

```
t ← 0
X(t) ← Init_x(n, M, [lb_x, ub_x]) as Eq.1              ▷ initialize initial population
F(t) ← f(X(t))                                          ▷ evaluate
w ← Init_{Δ:w}(w)                                       ▷ initialize w-relative step-size
x_g(t) ← Init_{Δ:x}(X(t)) as Eq.2                       ▷ initialize x-relative step-size
z ← Init_{Δ:z}(w) as Eq.3                               ▷ initialize z-relative step-size
while stop condition T do
    X̂(t+1) ← Opt_x(X(t), x_g(t), z(t), w) as Eq.4      ▷ generate temporarily updated population
    X̂(t+1) ← C(X̂(t+1)) as Eq.5                        ▷ treatment to outliers
    F(t+1) ← f(X̂(t+1))                                 ▷ evaluate
    X(t+1) ← S(X(t), X̂(t+1)) as Eq.6                   ▷ select and generate finally updated population
    x_g(t+1) ← Init_{Δ:x}(X(t), X(t+1)) as Eq.7         ▷ update x-relative step-size
    z(t+1) ← Opt_{Δ:z}(z(t), t+1) as Eq.8               ▷ update z-relative step-size
    t ← t+1
end while
```