0.0.1 Re-framed Grasshopper Optimization Algorithm (Re-framed GOA)

- **Objectives**:

Objective Problem	
$f(\mathbf{x}_i)$	fitness of $\mathbf{x}_i$, $(f : R^n \to R)$.
n	the dimensionality of the search space.
$[lb_\mathbf{x}, ub_\mathbf{x}]$	the interval of objective variable $\mathbf{x}$, in our case, it is defined in 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, $\mathbf{x}_i \in R^n$.

- **Influencing factors**:

T	maximum iteration, the budget in our cases, in our case, it is defined in IOHprofiler.
M	population size, $M = 100$.
$\mathbf{x}_g$	the best position that the whole population has found so far.
z	a decreasing coefficient.
$[lb_z, ub_z]$	the interval of assisting variable z, $[lb_z, ub_z] = [0.00004.1]$, $[lb_z, ub_z] \subset [-\infty, +\infty]$.
w_1	the intensity of attraction, $w_1 = 0.5$.
w_2	the attractive length scale, $w_2 = 1.5$.

- **Functions**:
  - Initialization Process:

(1) Initialize $\mathbf{x}_i(t=0)$:

$$\mathbf{x}_i(t=0) = \boldsymbol{\mathcal{U}}(lb_{\mathbf{x}}, ub_{\mathbf{x}}), i = 1 \dots M \tag{1}$$

(2) Initialize $\mathbf{x}_g(t=0)$:

$$\mathbf{x}_g(t=0) = \mathbf{Min}(\{\mathbf{x}_i(t)\}), i = 1 \dots M \tag{2}$$

(3) Initialize $z(t)$:

$$z_1(t) = ub_{z_1} - t \times (\frac{ub_{z_1} - lb_{z_1}}{T}) \tag{3}$$

- Optimization Process:

(1) Update $\mathbf{x}_i(t)$ to generate $\hat{\mathbf{x}}_i(t+1)$:

$$\begin{aligned} &\widetilde{D}_{i,j}(t) = 2 + \mathbf{Dist}(\mathbf{x}_i(t), \mathbf{x}_j(t)) \,\mathbf{mod}\, 2 \\ &\hat{\mathbf{x}}_i(t+1) = z(t) \times (\sum_{j=1, j\neq i}^{M} z(t) \times \frac{ub_{\mathbf{x}} - lb_{\mathbf{x}}}{2} \times (w_1 \times e^{\frac{-\widetilde{D}_{i,j}(t)}{w_2}} - e^{-\widetilde{D}_{i,j}(t)}) \times \frac{\mathbf{x}_i(t) - \mathbf{x}_j(t)}{\mathbf{Dist}(\mathbf{x}_i(t), \mathbf{x}_j(t))}) + \mathbf{x}_g \end{aligned} \tag{4}$$

(2) Dealing with outliers C:

$$\mathbf{x}_{i,n}^{\text{fixed}}(t+1) = \begin{cases} ub_x & , \quad \mathbf{x}_{i,n}(t+1) > ub_x \\ \mathbf{x}_{i,n}(t+1) & , \quad \text{o.w} \\ lb_x & , \quad \mathbf{x}_{i,n}(t+1) < lb_x \end{cases} \tag{5}$$

(3) Select $\mathbf{x}_i(t+1)$ from $\hat{\mathbf{x}}_i(t+1)$:

$$\mathbf{x}_i(t+1) = \hat{\mathbf{x}}_i(t+1) \tag{6}$$

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

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

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

$$z(t) = ub_z - t \times (\frac{ub_z - lb_z}{T}) \tag{8}$$

Algorithm 1 Re-framed GOA with population size M; search space $n, [lb_{\mathbf{x}}, ub_{\mathbf{x}}]$; stop condition T; initialization method $Init_{\mathbf{x}}$, optimization method $Opt_{\mathbf{x}}$, treatment C of outliers, and selection S to objective solutions; initialization method $Init_{\Delta}$ and optimization method Opt_{Δ} to step-size Δ.

$t \leftarrow 0$
$\mathbf{X}(t) \leftarrow Init_{\mathbf{x}}(n, M, [lb_{\mathbf{x}}, ub_{\mathbf{x}}])$ as Eq.1 ▷ initialize initial population
$F(t) \leftarrow f(\mathbf{X}(t))$ ▷ evaluate
$w \leftarrow Init_{\Delta:w}(w)$ ▷ initialize w-relative step-size
$\mathbf{x}_g(t) \leftarrow Init_{\Delta:\mathbf{x}}(\mathbf{X}(t))$ as Eq.2 ▷ initialize $\mathbf{x}$-relative step-size
$z(t) \leftarrow Init_{\Delta:z}(z^0)$ as Eq.3 ▷ initialize z-relative step-size
while stop condition T **do**
    $\hat{\mathbf{X}}(t+1) \leftarrow Opt_{\mathbf{x}}(\mathbf{X}(t), \mathbf{x}_g(t), z(t), w)$ as Eq.4 ▷ generate temporarily updated population
    $\hat{\mathbf{X}}(t+1) \leftarrow C(\hat{\mathbf{X}}(t+1))$ as Eq.**??** ▷ treatment to outliers
    $F(t+1) \leftarrow f(\hat{\mathbf{X}}(t+1))$ ▷ evaluate
    $\mathbf{X}(t+1) \leftarrow S(\mathbf{X}(t), \hat{\mathbf{X}}(t+1))$ as Eq.6 ▷ select and generate finally updated population
    $\mathbf{x}_g(t+1) \leftarrow Init_{\Delta:\mathbf{x}}(\mathbf{X}(t), \mathbf{X}(t+1))$ as Eq.7 ▷ update $\mathbf{x}$-relative step-size
    $z(t+1) \leftarrow Opt_{\Delta:z}(z(t), t+1)$ as Eq.8 ▷ update z-relative step-size
    $t \leftarrow t+1$
end while