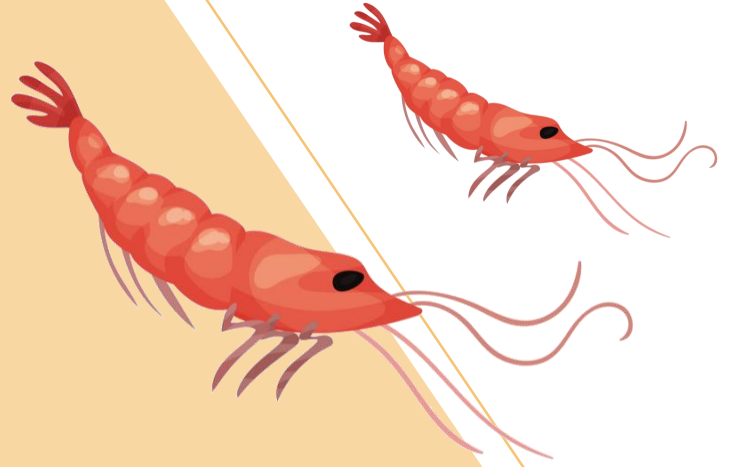


Soft Computing Final Project

# Krill Herd Algorithm



R08546019

干語彤

January 13<sup>th</sup>, 2020

# Agenda

1

Krill Herd Algorithm

2

程式人機介面

3

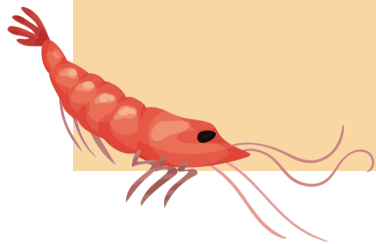
Demo

4

數值測試與比較

5

結論與討論



## Krill Herd Algorithm

- 1) 每一隻磷蝦為**建構解的代理人**，即磷蝦的**位置**
- 2) 模仿磷蝦個體由低密度往高密度聚集、往食物中心聚集的兩種特性
- 3) 移動至**新位置**的距離受到下列五個因素的影響：

Movement  
induced by  
neighbors

Foraging  
activity

Random  
diffusion

- **鄰居**和**迄今最佳位置**對於個別磷蝦的影響  
**Crossover** 食物位置的吸引  
別磷蝦對於過往食物位置經驗的影響  
**Mutation** 隨著迭代次數增加減少**擴散值**

# Krill Herd Algorithm — Pseudo Code

---

## *Krill Herd Algorithm*

---

Begin

**Step 1 : Initialization.** Define lower bound and upper bound of  $X$ . Generate random population (NP) of the krill positions  $X_i$  within the limit variables. Assign values to parameters :  $N_{\max}$  ,  $V_f$  ,  $D_{\max}$  and  $C_t$ . Set induced motion ( $N_i$ ), foraging motion ( $F_i$ ) and physical diffusion ( $D_i$ ) to zero

**Step 2 : Evaluation.** Calculate fitness for each krill individual

**Step 3 : For  $I = 1 : I_{\max}$**  (iteration)

    Calculate time interval  $\Delta_t(I)$

**Induced motion step**

    Evaluate local, target swarm density and the direction of each krill individual

    Calculate the distances between each 2 krill positions ( $d_{sij}$ )

    Identify the sensing distance ( $d_{si}$ ) of each krill individual

**If**  $d_{sij} < d_{si}$

            Calculate and update the induced motion  $N_i(I)$

**End If**

**Foraging motion step**

    Generate center of food density  $X^{\text{food}}(I)$  for the herd krill

    Calculate and update the foraging motion  $F_i(I)$

**Physical diffusion step**

    Calculate and update the diffusion motion  $D_i(I)$

**Genetic operators implementation step**

    Apply crossover and mutation operator on each krill individual

    Move the krill positions after  $t+\Delta_t$  period

    Evaluate the fitness for each krill individual

    Rank the krill positions according to minimum fitness

    Record so far the best krill position and its fitness

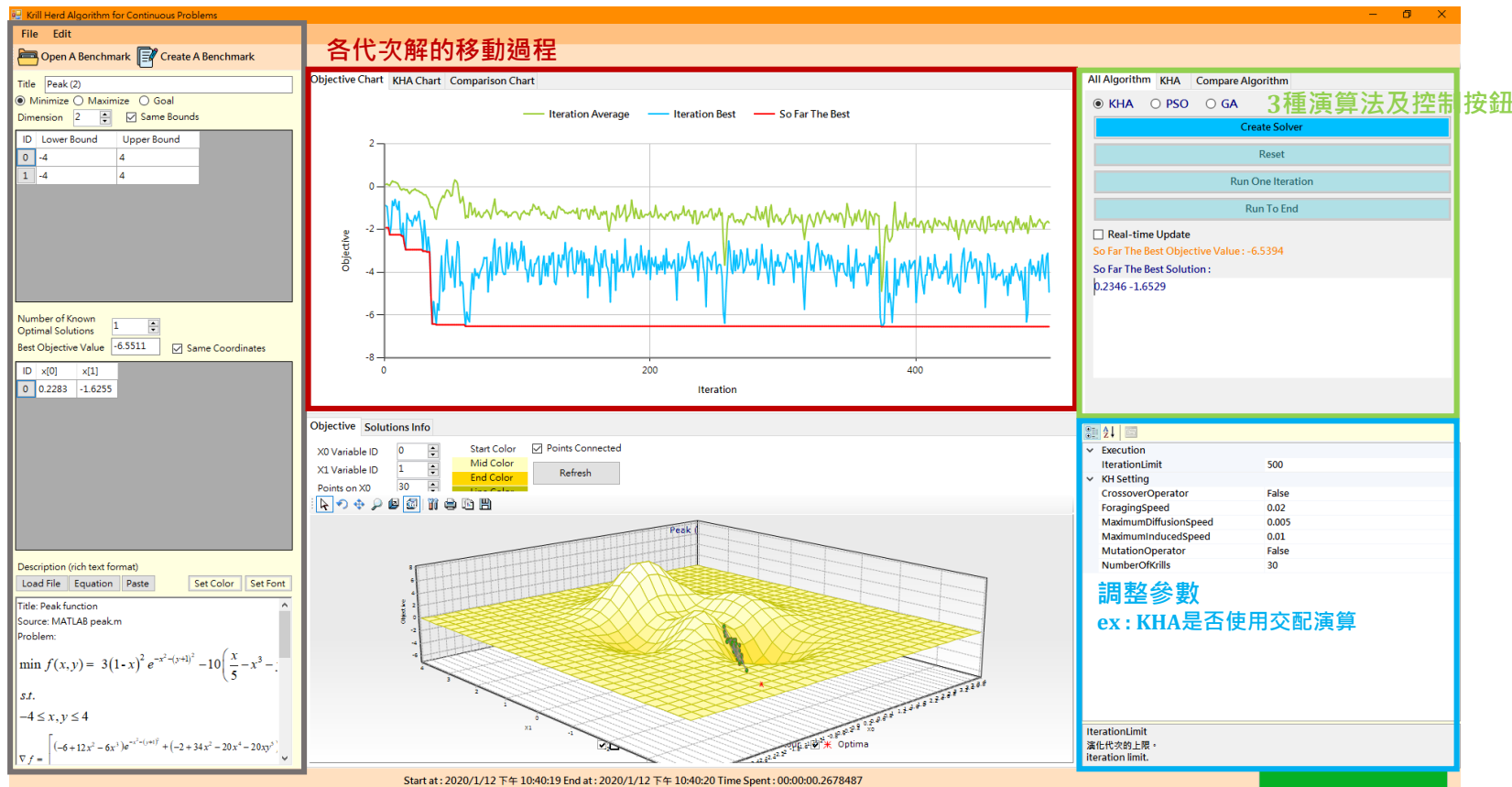
**Step 4 : End For  $I$**  (obtain so far the best solution)

End.

---

# 單一演算法的求解介面

## 讀取連續問題的COP檔



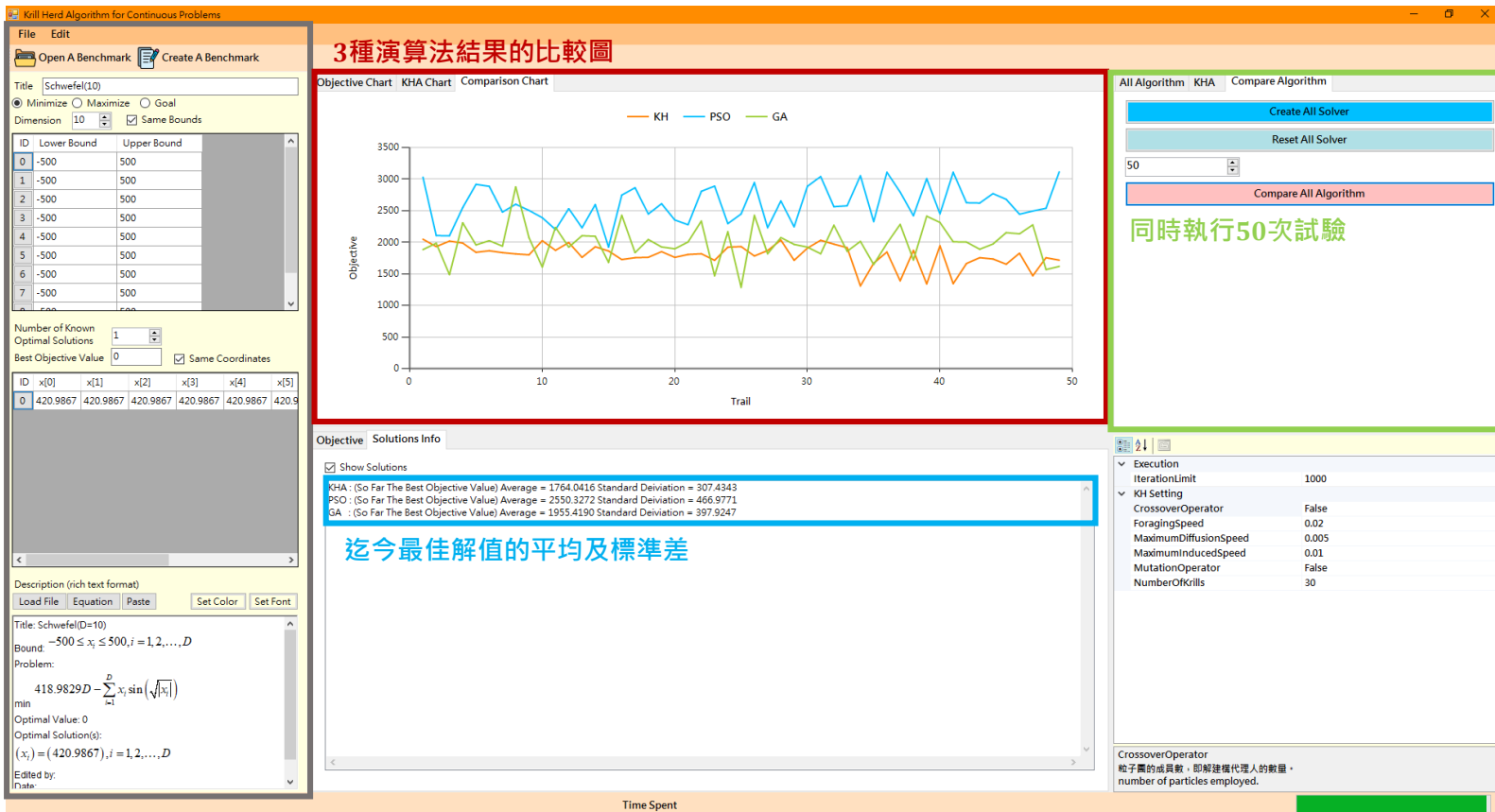
# 比較KHA I、KHA II、KHA III、KHA IV的求解介面

讀取連續問題的COP檔



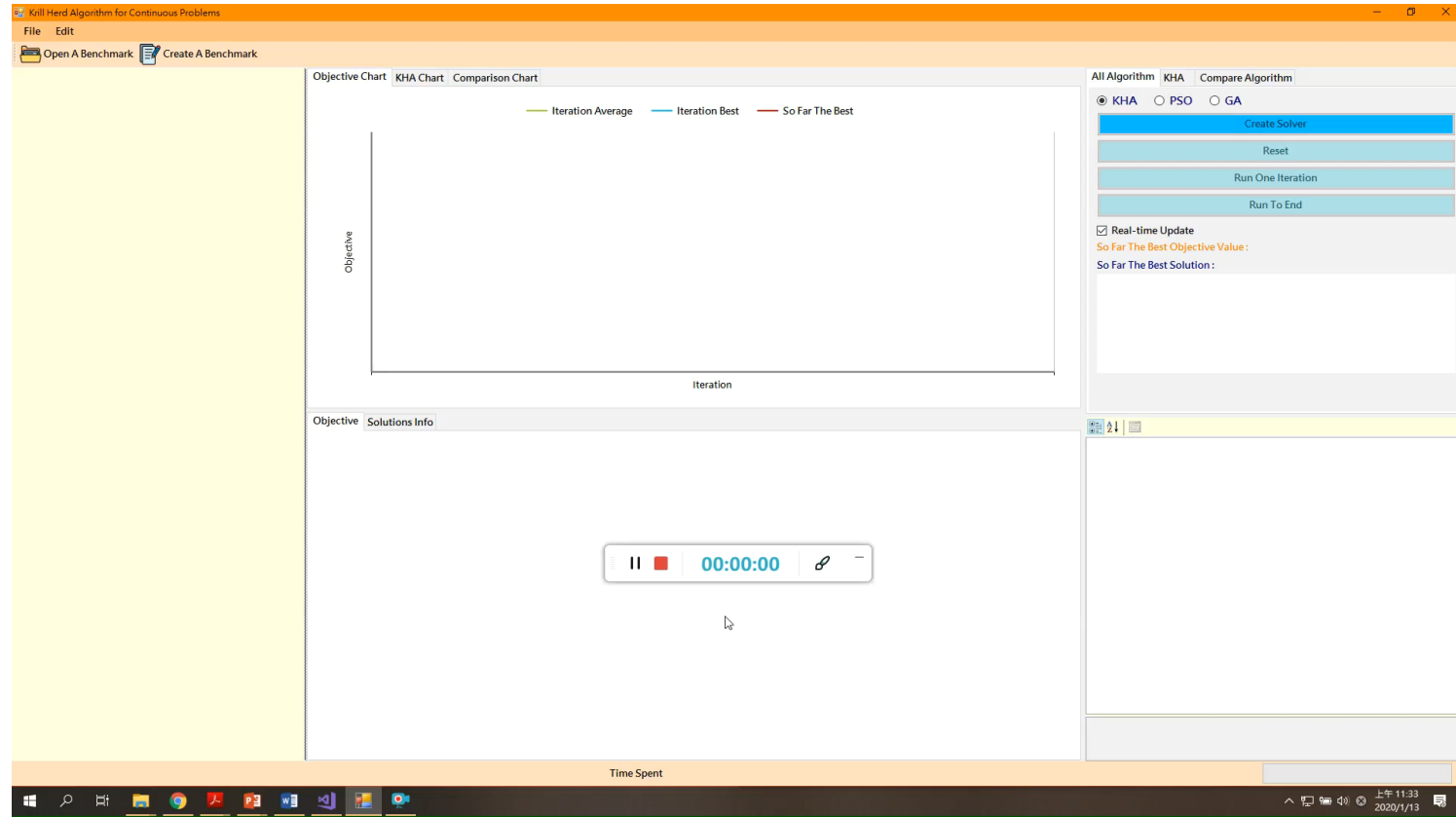
# 比較KHA、PSO、GA的求解介面

讀取連續問題的COP檔



# Demo

<https://drive.google.com/file/d/1CA9rFt29-U2A07HoMDjv-oFIXm2LZn7h/view?usp=sharing>





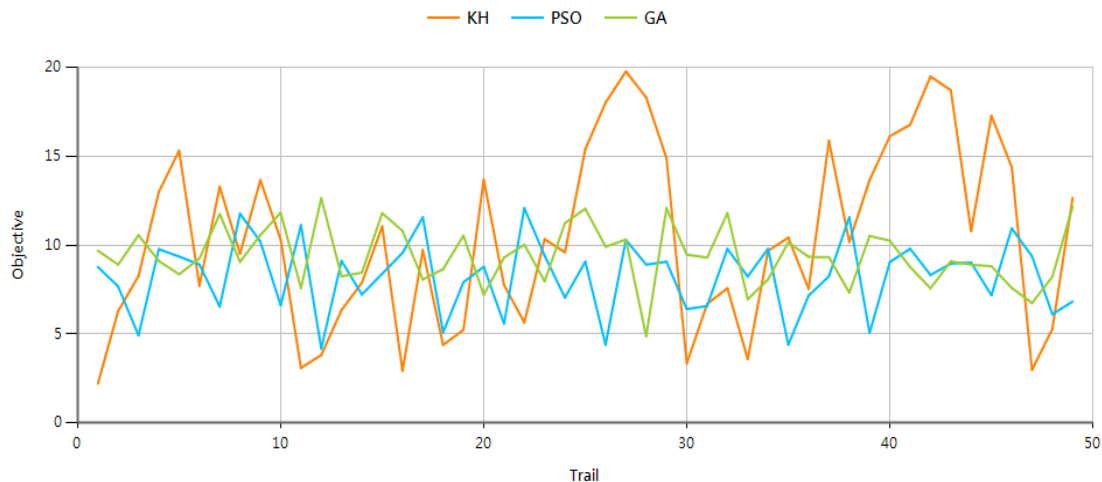
## 數值測試與比較 — KHA I、KHA II、KHA III、KHA IV

		KHA I	KHA II	KHA III	KHA IV
Peak (2)	平均值	-6.3232	-6.4169	-6.2795	<b>-6.4196</b>
	標準差	1.0300	0.9167	1.1291	0.9171
Branin (2)	平均值	0.0022	0.0010	0.0021	<b>0.0000</b>
	標準差	0.0045	0.017	0.0143	0.0000
Ackley (10)	平均值	13.2094	11.0641	8.0807	<b>7.9776</b>
	標準差	6.5264	5.8163	5.0282	4.5699
Schwefel (10)	平均值	1499.4820	1585.3483	462.6957	<b>363.3636</b>
	標準差	570.9953	549.6717	662.9070	519.4757
Girewank (10)	平均值	1.0926	1.0999	0.9949	<b>0.9874</b>
	標準差	0.1598	0.1627	0.1424	0.1412

→ **KHA IV**的平均表現結果最好

→ 無論使用交配或突變運算，皆比沒有使用任何基因運算來得好

# 數值測試與比較 — KHA、PSO、GA



A

→ 固定各演算法  
的參數

6.3870

0.9198

→ **KHA** : Peak (2)、  
Schwefel (10)

0.0008

0.0037

→ **GA** : Branin (2)、  
Ackley (10)

2.3161

少數試驗中仍可  
以找到遠比PSO及

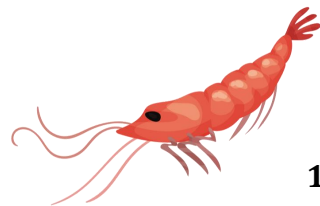
GA更好的目標值

2.1001

Schwefel (10)	平均值	1666.4111	2469.4473	1829.6760
	標準差	394.5674	503.6942	388.2894
Girewank (10)	平均值	1.0982	1.0065	2.0544
	標準差	0.1595	0.1446	1.9769

## 結論與討論

- 1) 使用交配或突變運算(KHA II、KHA III、KHA IV)的求解能力較無使用任何基因運算 (KHA I)來得好，其中又以同時使應用交配及突變運算為最突出
- 2) KHA的求解能力與GA不分上下
- 3) KHA及PSO皆受解代理人數量的影響。當數量增加時，兩種演算法的效率明顯提升，而PSO又比KHA更容易陷在區域最佳解無法跳脫
- 4) KHA在前期進行大範圍的隨機搜索，因此收斂速度較PSO與GA來得慢，但隨著迭代次數的增加，KHA較其他兩種演算法來得活絡，並於後期進行小範圍的最佳解搜尋
- 5) KHA的參數值是否為最佳、鄰居的感知距離及虛擬食物中心的估算是否有其他方法可以替代，以促進演算法的效用



# Thank you for your listening !

Any questions ?

