

# Standardizing Nature-Inspired Optimization Algorithms

## --- A Unified Framework *UNIOA* for Seven Selected Algorithms

Huilin Li(2556057)|supervised by Anna, Carola,



Universiteit  
Leiden  
The Netherlands

# Agenda

## 1. Motivation

## 2. Outline of research

## 3. Steps of research

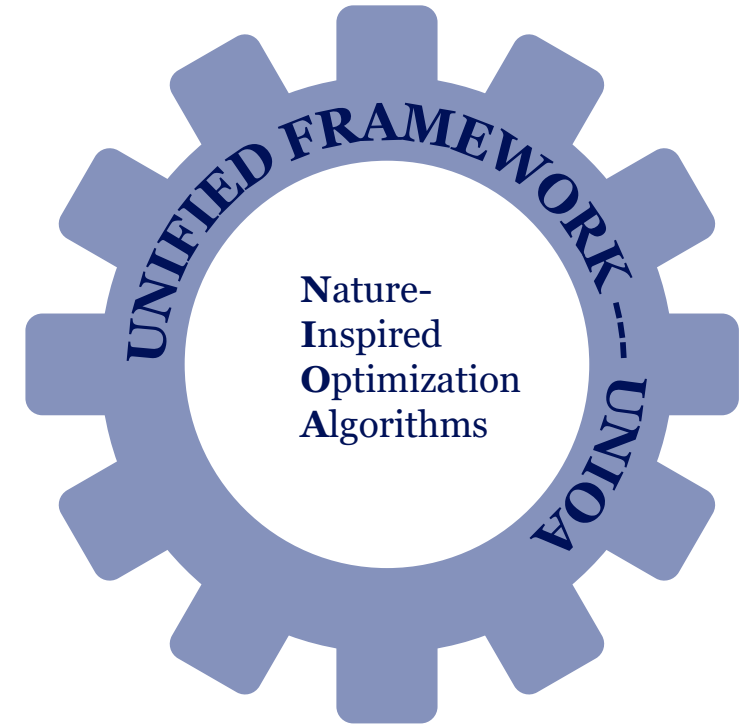
### 3.1 Theoretical Analysis

- ✓ Unified Terminologies
- ✓ Unified Procedure
- ✓ **Unified Framework** --- Unified framework for **NIOA** --- **UNIOA**

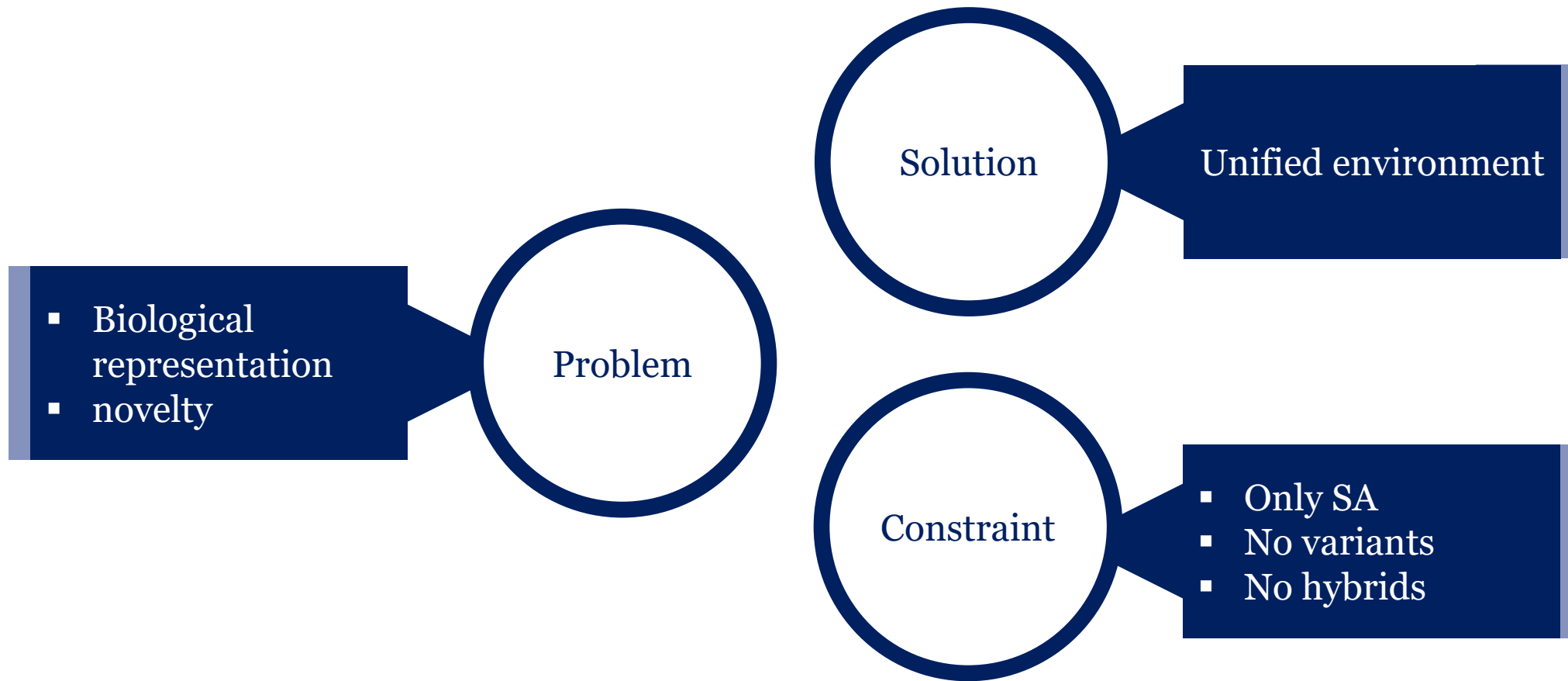
### 3.2 Practical Analysis

- ✓ Experimental Setup
- ✓ Experimental Results

## 4. Conclusion



# 1. Motivation



## 2. Outline of research

- 7 SA

- ☐ Old classical SA

- ✓ PSO

- ☐ New modern SA

- ✓ BA, GOA, CSA, MFO, MBO, BOA

- **Theoretical Analysis**

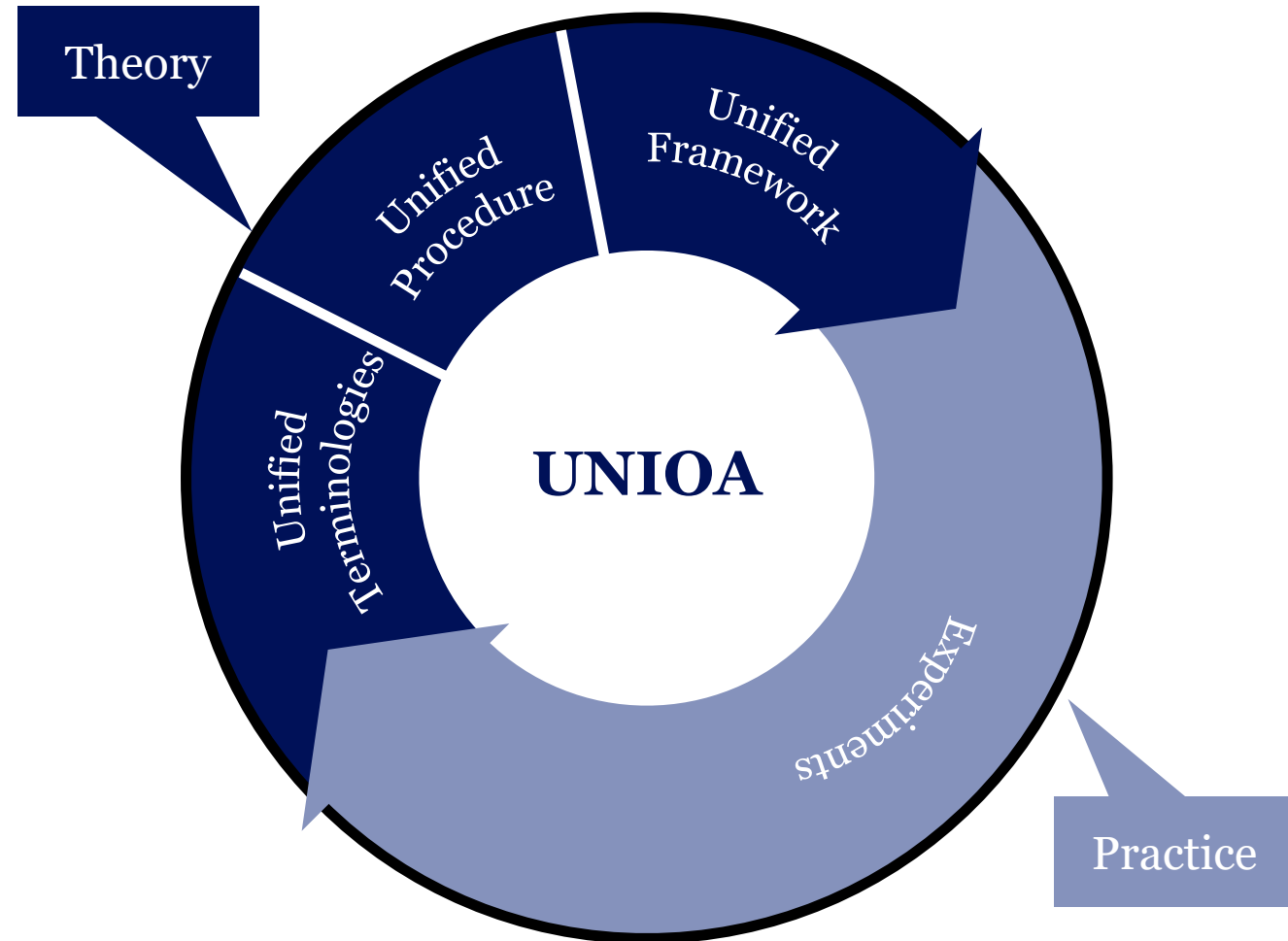
- ☐ Unified terminologies

- ☐ Unified Procedure

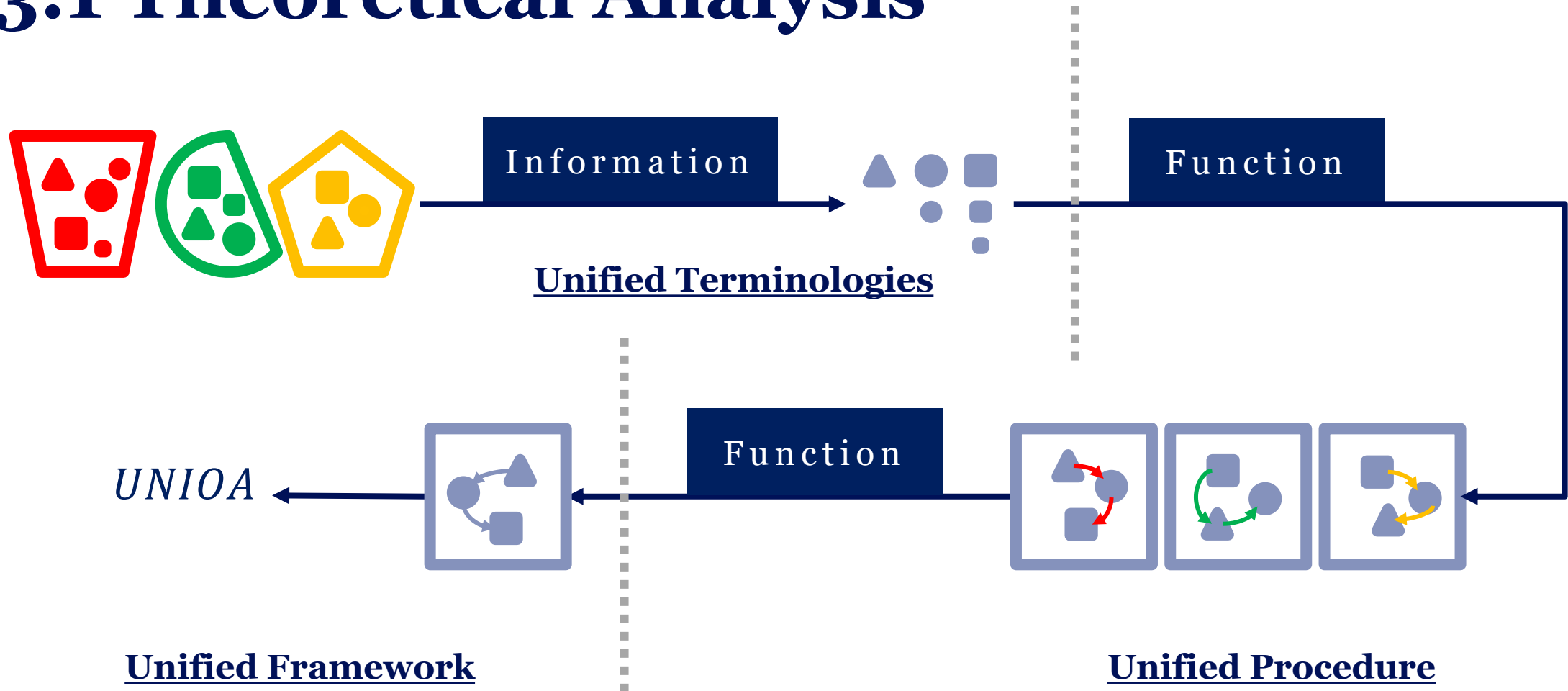
- ☐ Unified Framework --- UNIOA

- **Practical Analysis**

- ☐ Experiments



# 3.1 Theoretical Analysis



# ✓ Unified Terminologies



- Categorize various terminologies
- The information is the principle

2. One possible objective solution.

• *BA*: each bat  $x_i$ . • *GOA*: each grasshopper  $x_i$ . • *CSA*: crow  $x_i$ .  
• *MFO*: moth  $m_i$ . • *MBO*: monarch butterfly  $x_i$ . • *BOA*: butterfly  $x_i$ .  
• *PSO*: particle  $x_i$ .

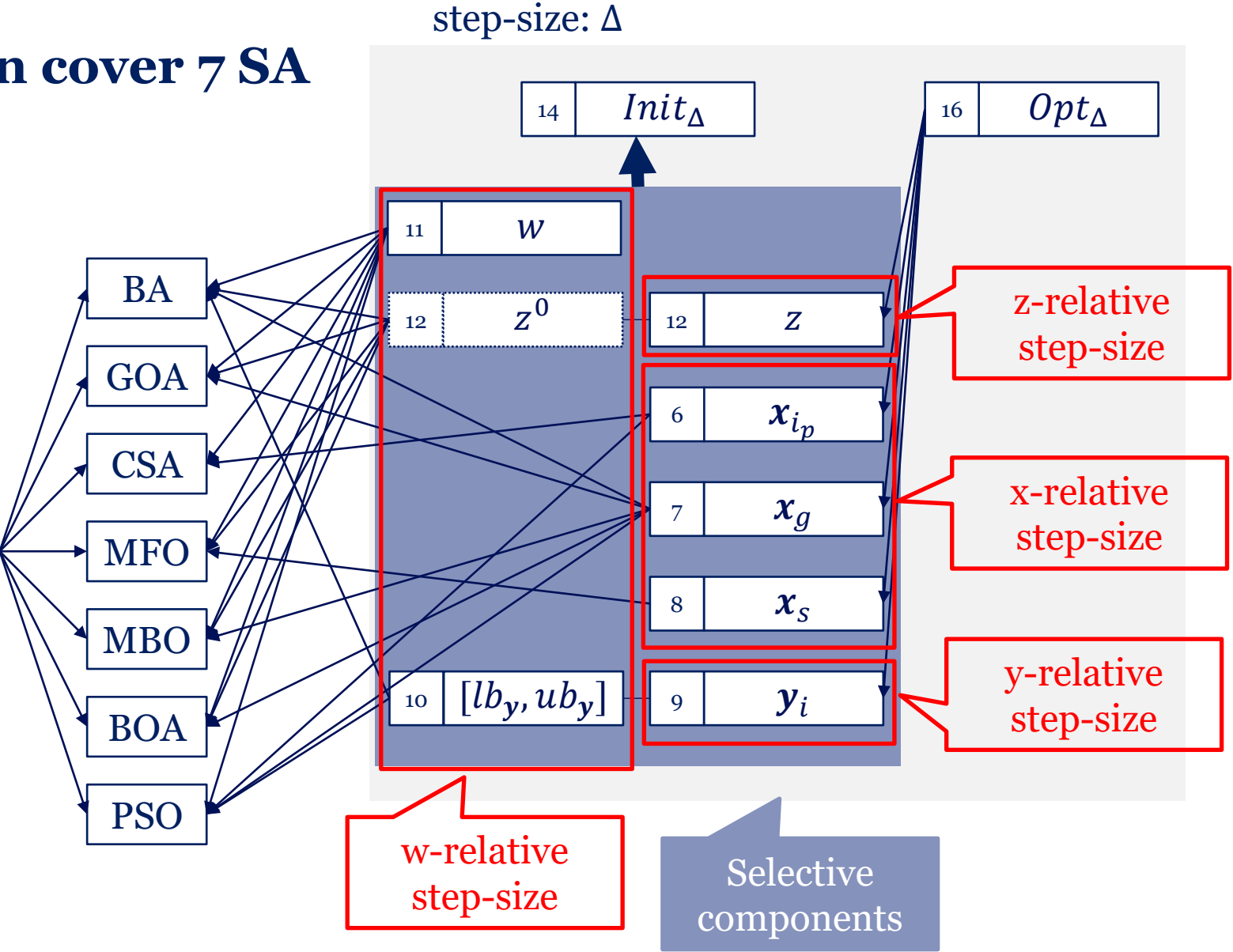
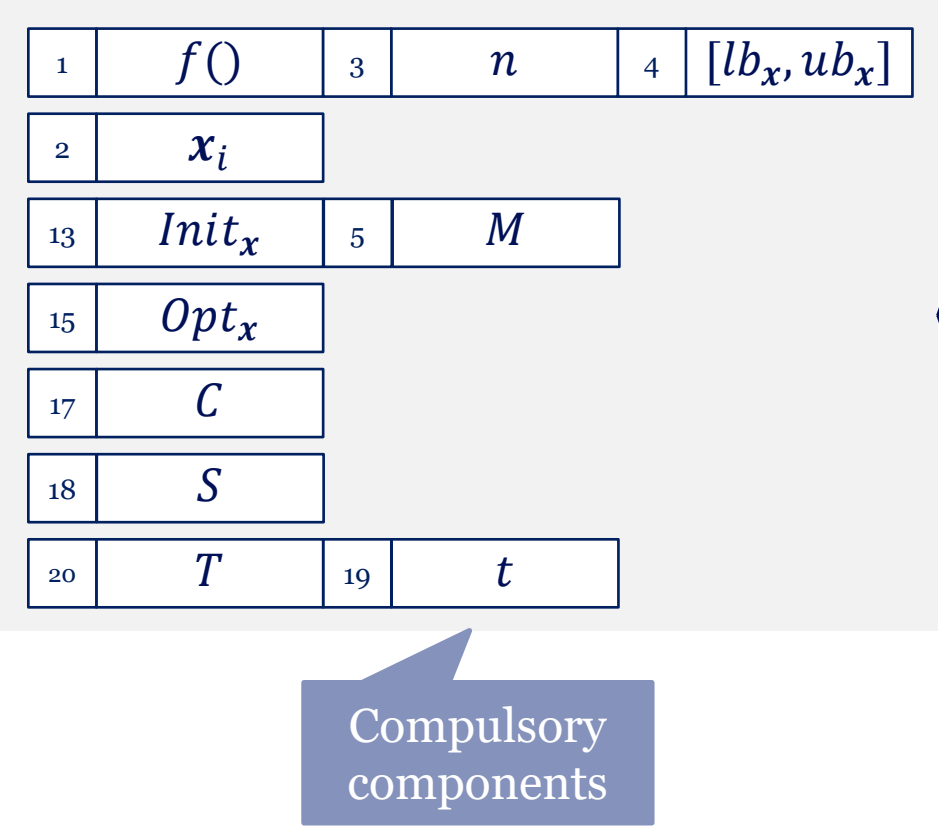
$x_i$ : one objective solution.

Information

Various  
terminologies

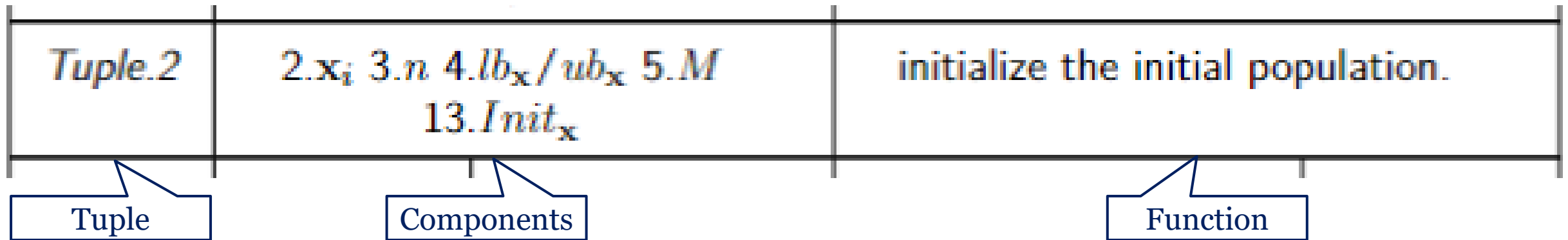
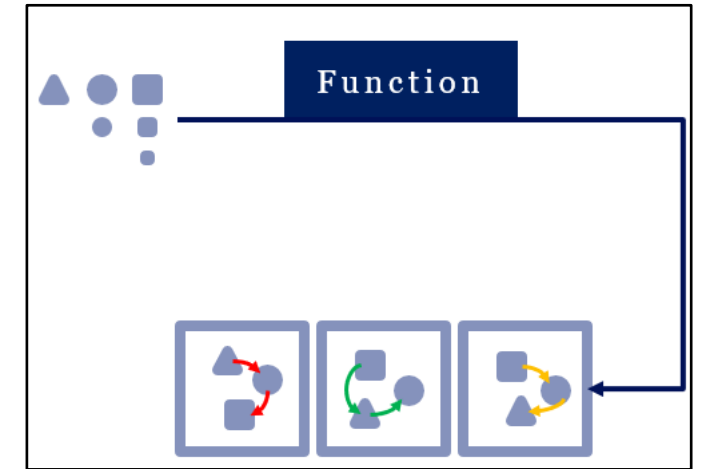
Unified  
terminology

# 20 unified terminologies can cover 7 SA



# ✓ Unified Procedure

- Assign the same position, after categorizing unified terminologies
- The function is the principle





- **8-tuple**
- $NIOA = (f, Init_x, Opt_x, C, T, S, Init_{\Delta}, Opt_{\Delta})$
- **8 tuples can be at the same position when modeling 7 SA**

---

**Algorithm 1** Unified Procedure for seven selected algorithms (the unified positions of eight tuples in Table 3.1)

---

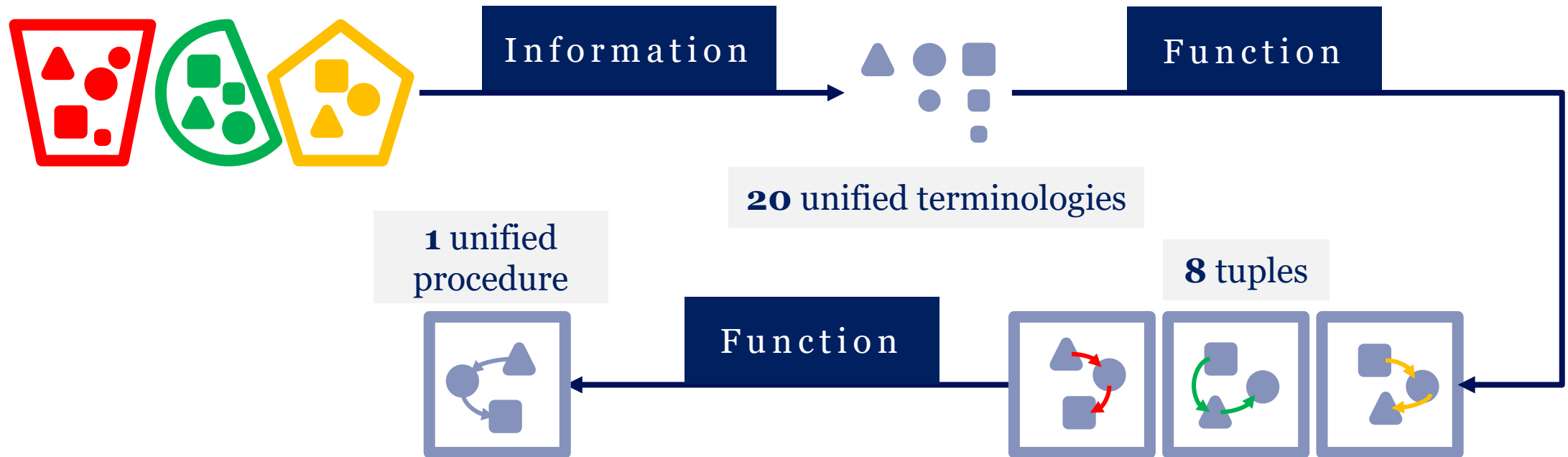
```

1:  $t \leftarrow 0$                                 ▷ iteration counter
2:  $Tuple.2 \text{ } Init_x$                         ▷ initialize population
3:  $Tuple.1 \text{ } f$                                 ▷ evaluation
4:  $Tuple.3 \text{ } Init_{\Delta:w,z,x,y}$         ▷ initialize  $w$ -relative,  $z$ -relative,  $x$ -relative and  $y$ -relative
   step-size  $\Delta$ 
5: while  $Tuple.8 \text{ } T$  do                                ▷ stop strategy
6:    $Tuple.5 \text{ } Opt_{\Delta:y}$  ▷ update  $y$ -relative step-size  $\Delta$ , if  $y$ -relative step-size exists in the
   initialization process
7:    $Tuple.4 \text{ } Opt_x$                                 ▷ update population
8:    $Tuple.6 \text{ } C$                                 ▷ outliers treatment
9:    $Tuple.1 \text{ } f$                                 ▷ evaluation
10:   $Tuple.7 \text{ } S$                                 ▷ selection
11:   $Tuple.5 \text{ } Opt_{\Delta:z,x}$         ▷ update  $z$ -relative and  $x$ -relative step-size  $\Delta$ 
12:   $t \leftarrow t + 1$ 
13: end while

```

---

# ✓ Unified Framework



$$UNIOA = (f, Init_x, Opt_x, C, T, S, Init_{\Delta}, Opt_{\Delta})$$

$$UNIOA = (f, Init_x, Opt_x, C, T, S, Init_{\Delta}, Opt_{\Delta})$$

- **Unified framework**  
***UNIOA* is constructed well**

---

**Algorithm 9** Unified Nature-Inspired Optimization Algorithm — UNIOA

---

```

1:  $t \leftarrow 0$ 
2:  $\mathbf{X}(t) \leftarrow Init_{\mathbf{x}}(n, M, [lb_{\mathbf{x}}, ub_{\mathbf{x}}])$  ▷ initialize initial pop
3:  $F(t) \leftarrow f(\mathbf{X}(t))$  ▷ evaluate
4:  $\Delta_{w,z,y,x}(t) \leftarrow Init_{\Delta:w,z,y,x}(\mathbf{X}(t), w, z^0, [lb_y, ub_y], t)$  ▷ initialize step-size
5: while  $T$  do
6:   if  $y \in \Delta(t=0)$  then
7:      $\mathbf{Y}(t+1) \leftarrow Opt_{\Delta:y}(\mathbf{Y}(t), w)$  ▷ update y-relative step-size
8:      $\hat{\mathbf{X}}(t+1) \leftarrow Opt_{\mathbf{x}}(\mathbf{X}(t), \mathbf{Y}(t+1), \Delta_{w,z,x}(t))$  ▷ temporarily updated pop
9:      $\hat{\mathbf{X}}(t+1) \leftarrow C(\hat{\mathbf{X}}(t+1))$  ▷ outliers treatment
10:     $F(t+1) \leftarrow f(\hat{\mathbf{X}}(t+1))$  ▷ evaluate
11:     $\mathbf{X}(t+1) \leftarrow S(\mathbf{X}(t), \hat{\mathbf{X}}(t+1), \Delta_{w,z}(t))$  ▷ select and generate finally updated pop
12:     $\Delta_{z,x}(t+1) \leftarrow Opt_{\Delta:z,x}(\mathbf{X}(t), \mathbf{X}(t+1), z(t), t+1)$  ▷ update z,x-relative step-size
13:   else
14:      $\hat{\mathbf{X}}(t+1) \leftarrow Opt_{\mathbf{x}}(\mathbf{X}(t), \Delta(t))$  ▷ temporarily updated pop
15:      $\hat{\mathbf{X}}(t+1) \leftarrow C(\hat{\mathbf{X}}(t+1))$  ▷ outliers treatment
16:      $F(t+1) \leftarrow f(\hat{\mathbf{X}}(t+1))$  ▷ evaluate
17:      $\mathbf{X}(t+1) \leftarrow S(\mathbf{X}(t), \hat{\mathbf{X}}(t+1), \Delta_{w,z}(t))$  ▷ select and generate finally updated pop
18:      $\Delta(t+1) \leftarrow Opt_{\Delta}(\mathbf{X}(t), \mathbf{X}(t+1), \Delta(t), t+1)$  ▷ update step-size
19:   end if
20:    $t \leftarrow t + 1$ 
21: end while

```

---

## 3.2 Practical Analysis

IOHprofiler

IOHexperimenter

**A** vs **A**  
ORIGINAL UNIOA

**A**<sub>1</sub> vs **A**<sub>2</sub> ... vs **A**<sub>7</sub>  
UNIOA UNIOA ... UNIOA

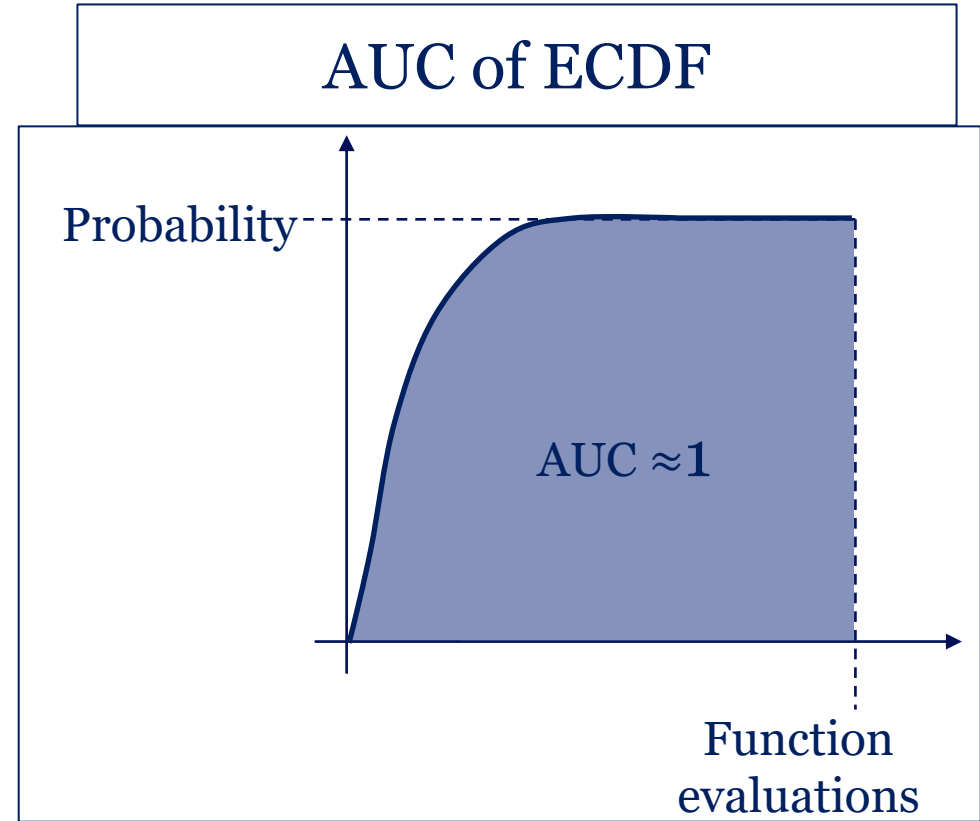
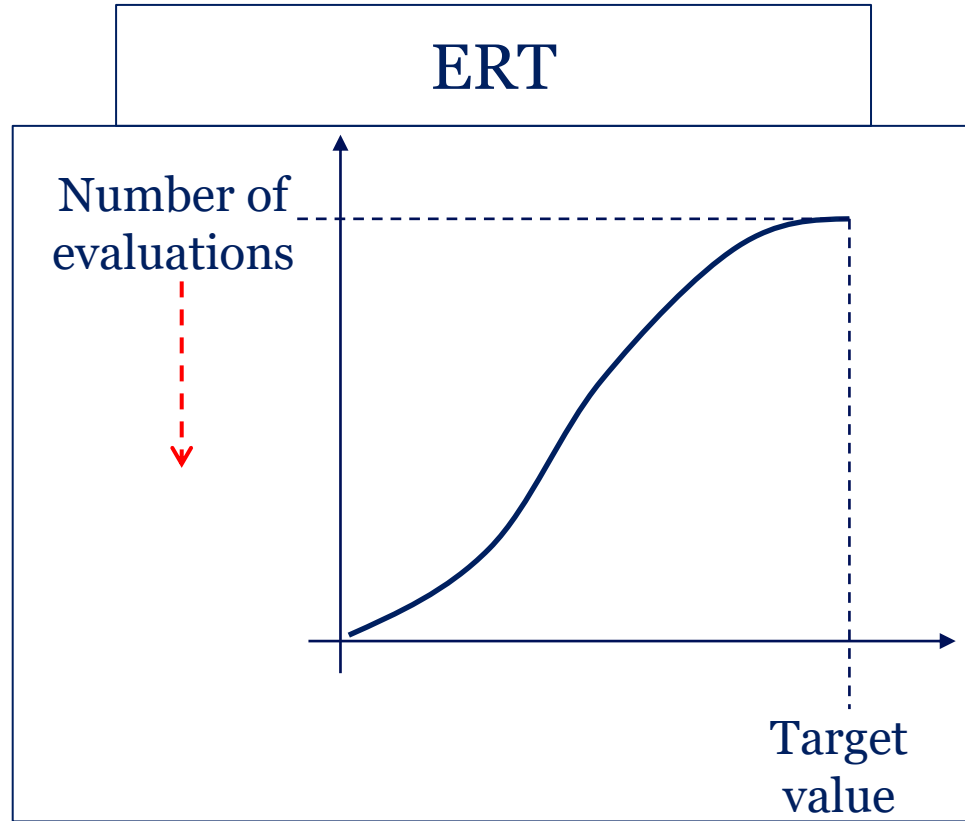
24 BBOB problems

IOHanalyzer

ERT

AUC of ECDF

## 3.2 Practical Analysis



**A** vs **A**  
**ORIGINAL** vs **UNIOA**

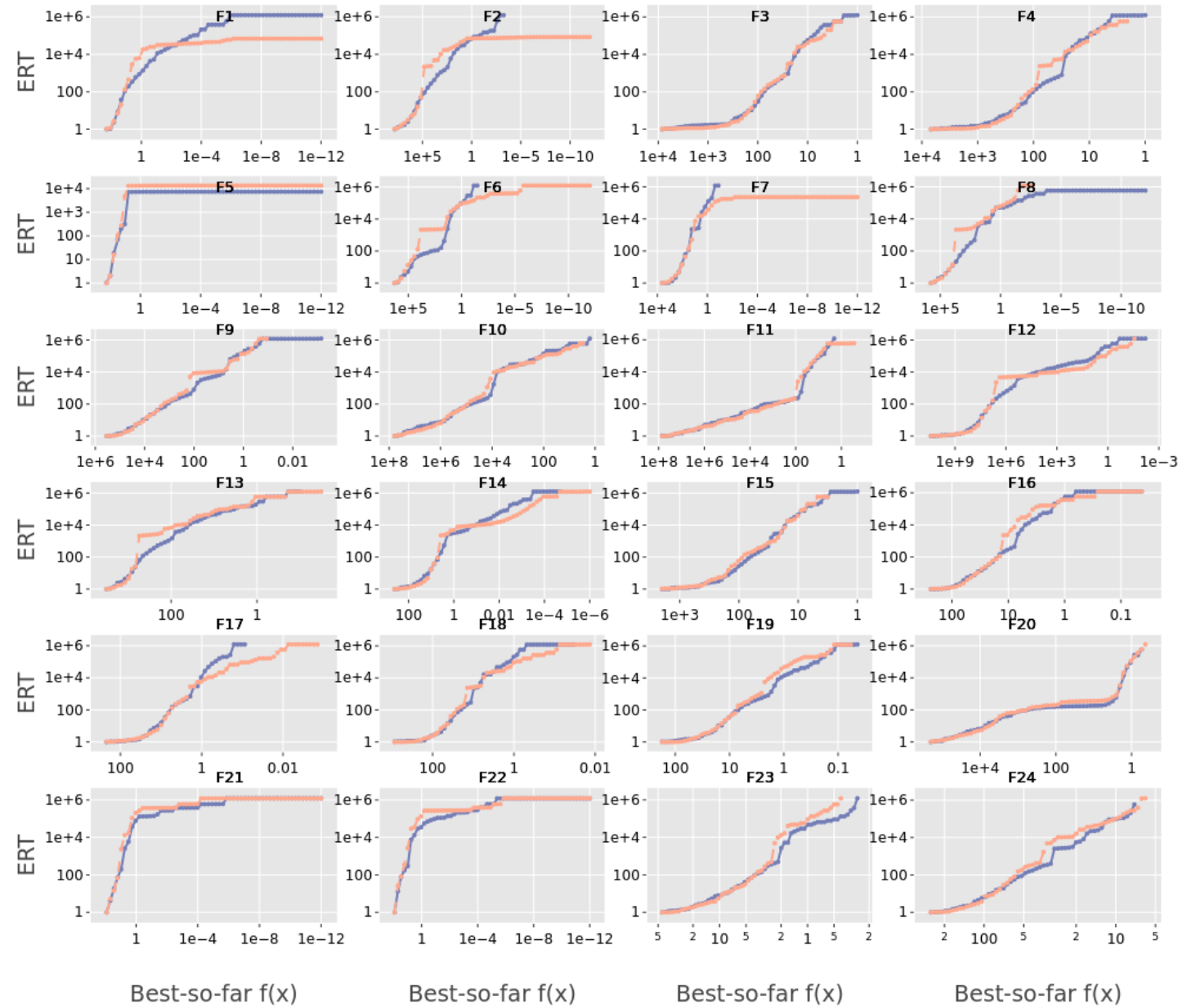
## ERT plot

### Output:

- ✓ Two lines are close. (e.g. MFO)

### Conclusion:

- ✓ At least in these seven algorithms, **UNIOA** performs same as **ORIGINAL** in the view of ERT plot.



— orig\_MFO — UNIOA\_MFO

A  
ORIGINAL

vs

A  
UNIOA

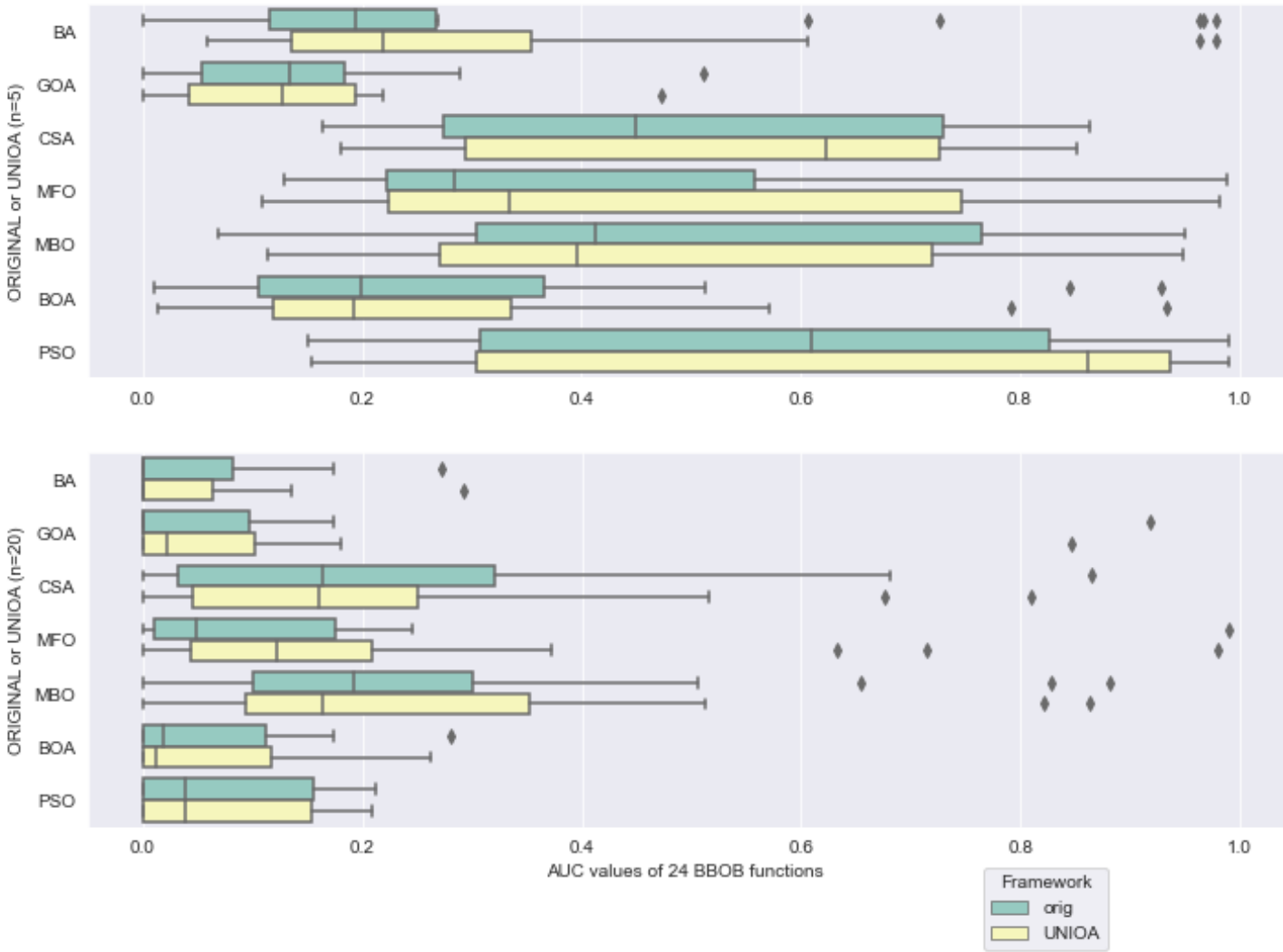
## AUC values

☐ Output:

- ✓ Two boxes are close.
- ✓ Accept Ho.

☐ Conclusion:

- ✓ At least in these seven algorithms, UNIOA performs same as ORIGINAL also in the view of AUC values.



$A_1$  VS  $A_2$  ... VS  $A_7$   
UNIOA UNIOA UNIOA

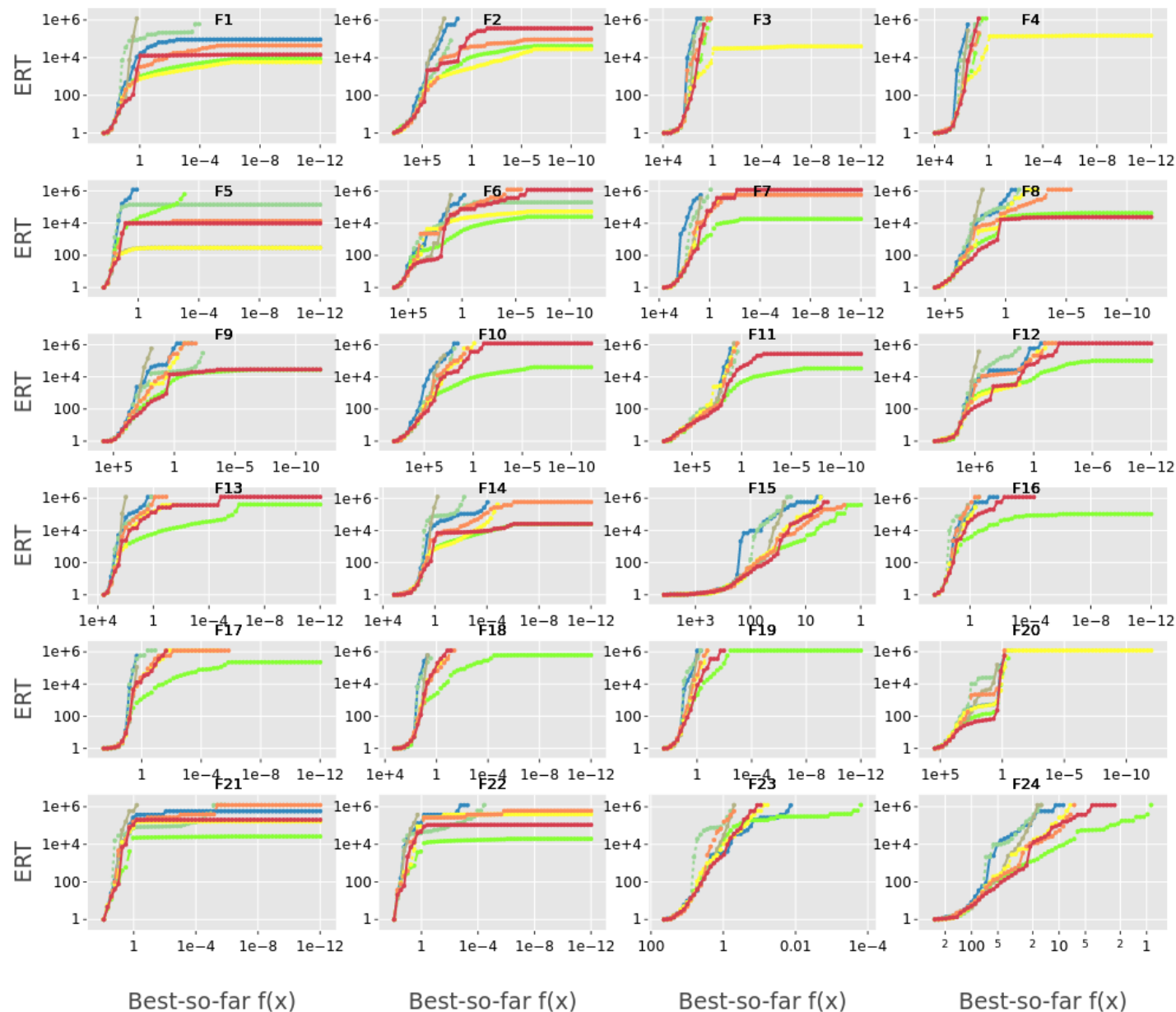
## ERT plot

### □ Output:

- ✓ ERT lines over 7 algorithms in UNIOA

### □ Conclusion:

- ✓ CSA





$A_1$  vs  $A_2$  ... vs  $A_7$   
UNIOA UNIOA UNIOA

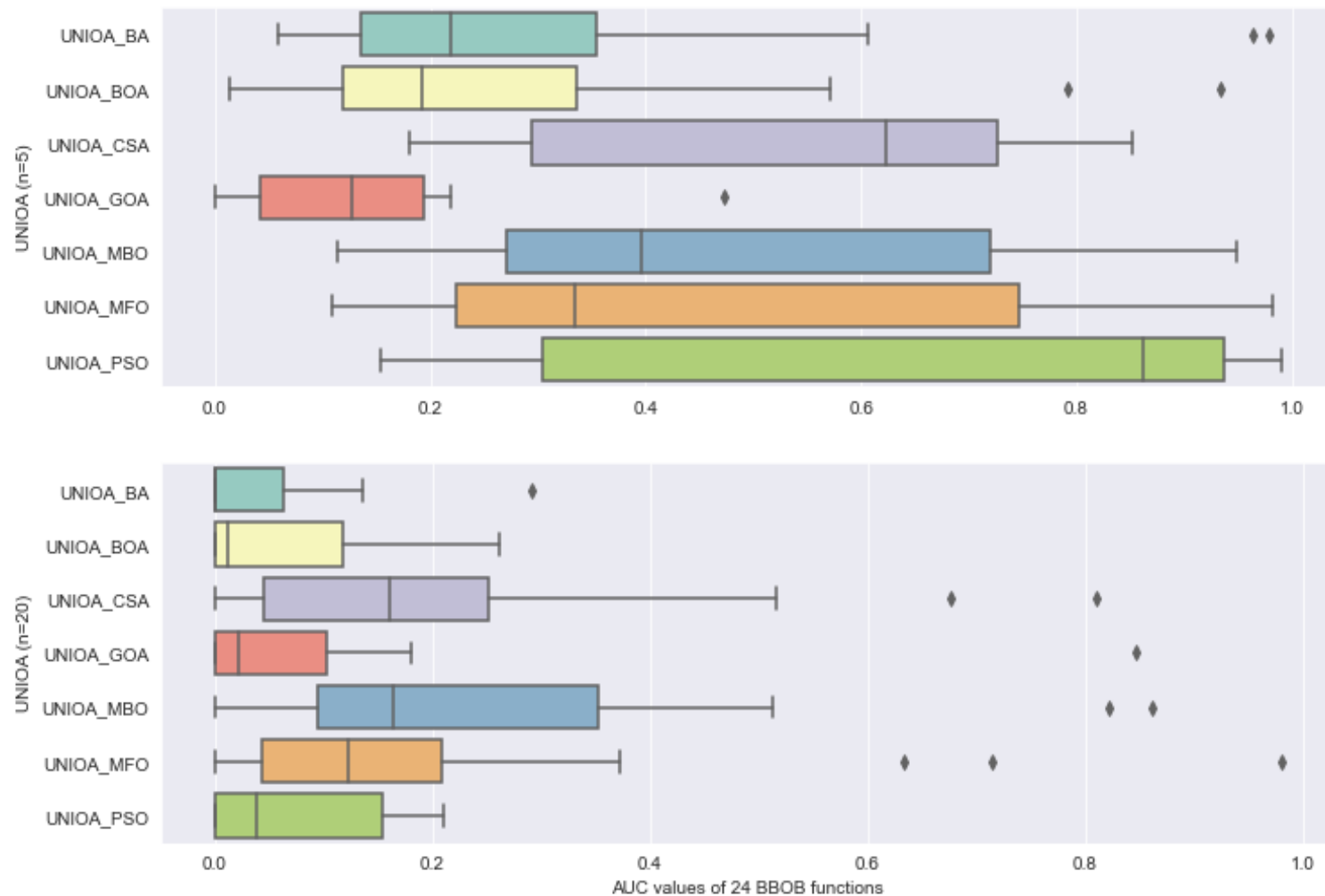
AUC values

#### ❑ Output:

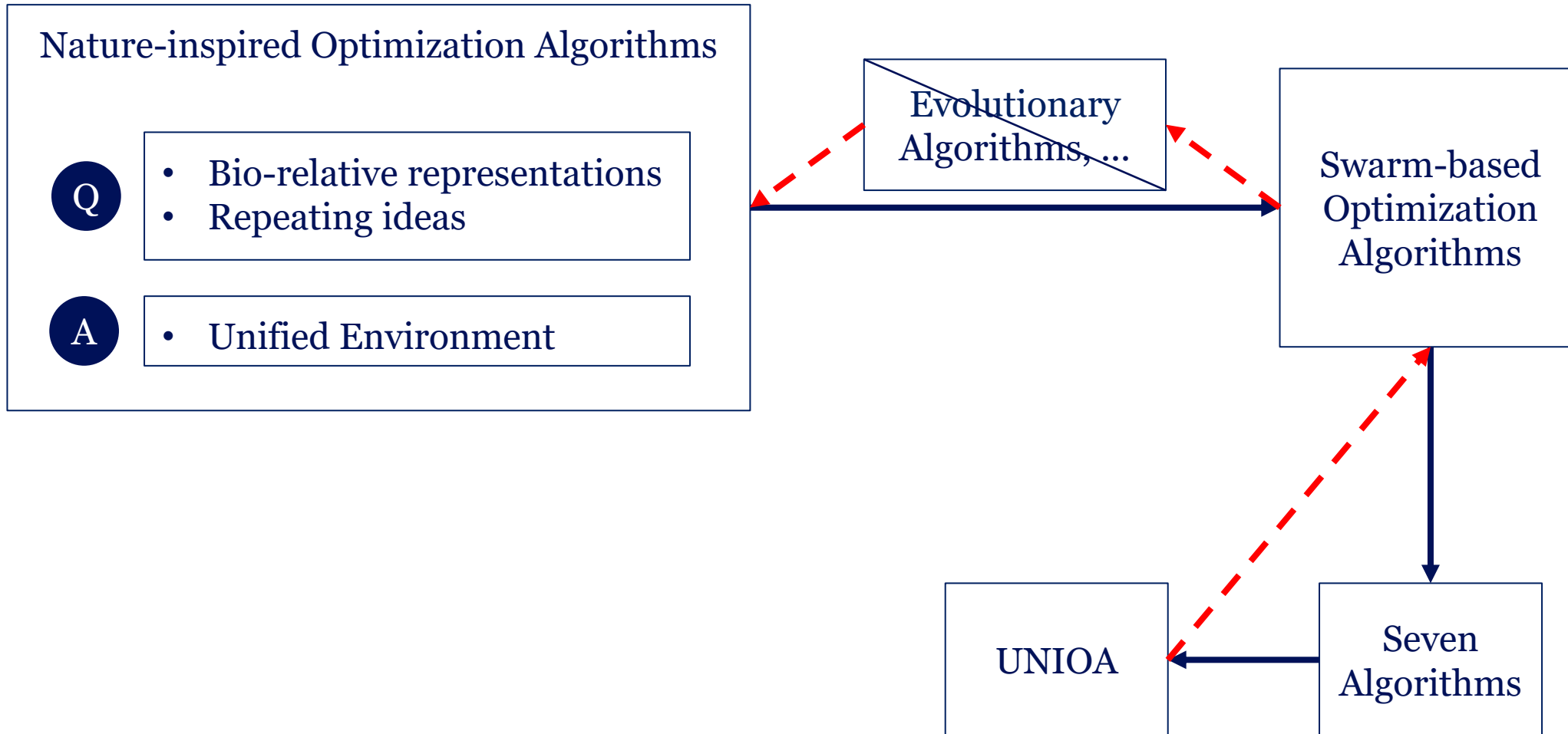
- ✓ AUC boxes over 7 algorithms in UNIOA

#### ❑ Conclusion:

- ✓ PSO, when Dim=5
- ✓ MBO, when Dim=20



# 4. Conclusion



# 4. Conclusion

- ❑ <https://github.com/Huilin-Li/ThesisProject> Huilin
- ❑ <https://github.com/Huilin-Li/UNIOA>

# Thanks !



Universiteit  
Leiden  
The Netherlands