Standardizing Nature-Inspired Optimization Algorithms

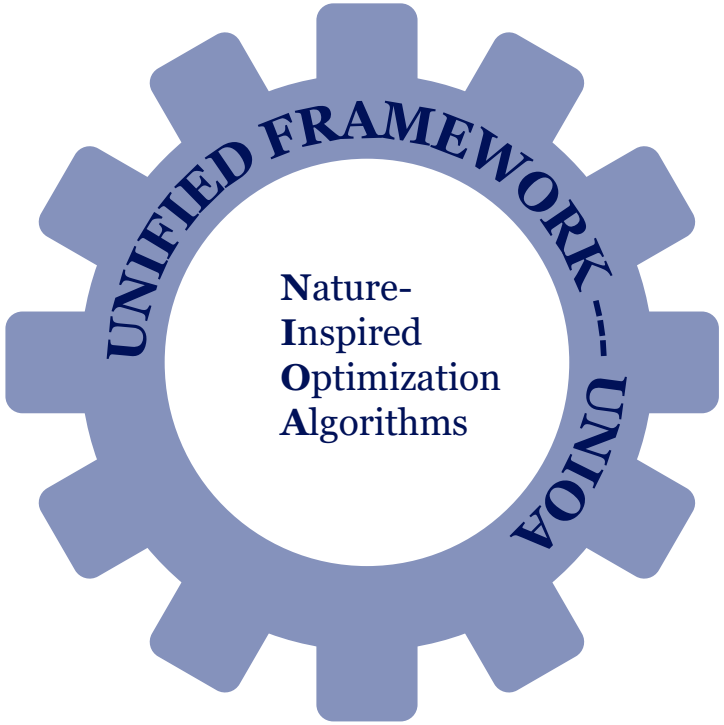
--- A Unified Framework *UNIOA* for Seven Selected Algorithms

Huilin Li(2556057)|supervised by Anna, Carola, Diederick|16-May-2022

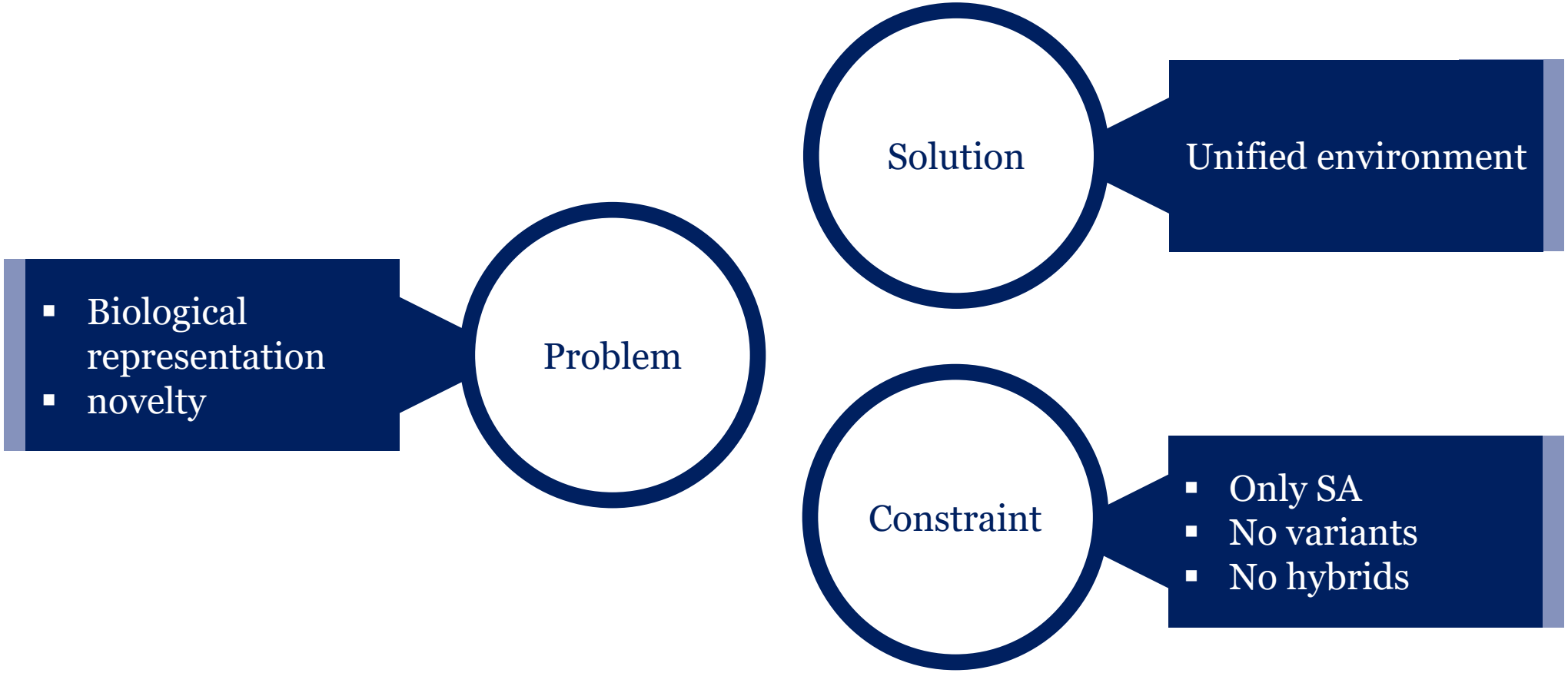
Universiteit Leiden
The Netherlands

Agenda

1. Motivation
2. Outline of research
3. Steps of research
   - 3.1 Theoretical Analysis
     - ✓ Unified Terminologies
     - ✓ Unified Procedure
     - ✓ **Unified Framework** --- **U**nified framework for **NIOA** --- **UNIOA**
   - 3.2 Practical Analysis
     - ✓ Experimental Setup
     - ✓ Experimental Results
4. Conclusion

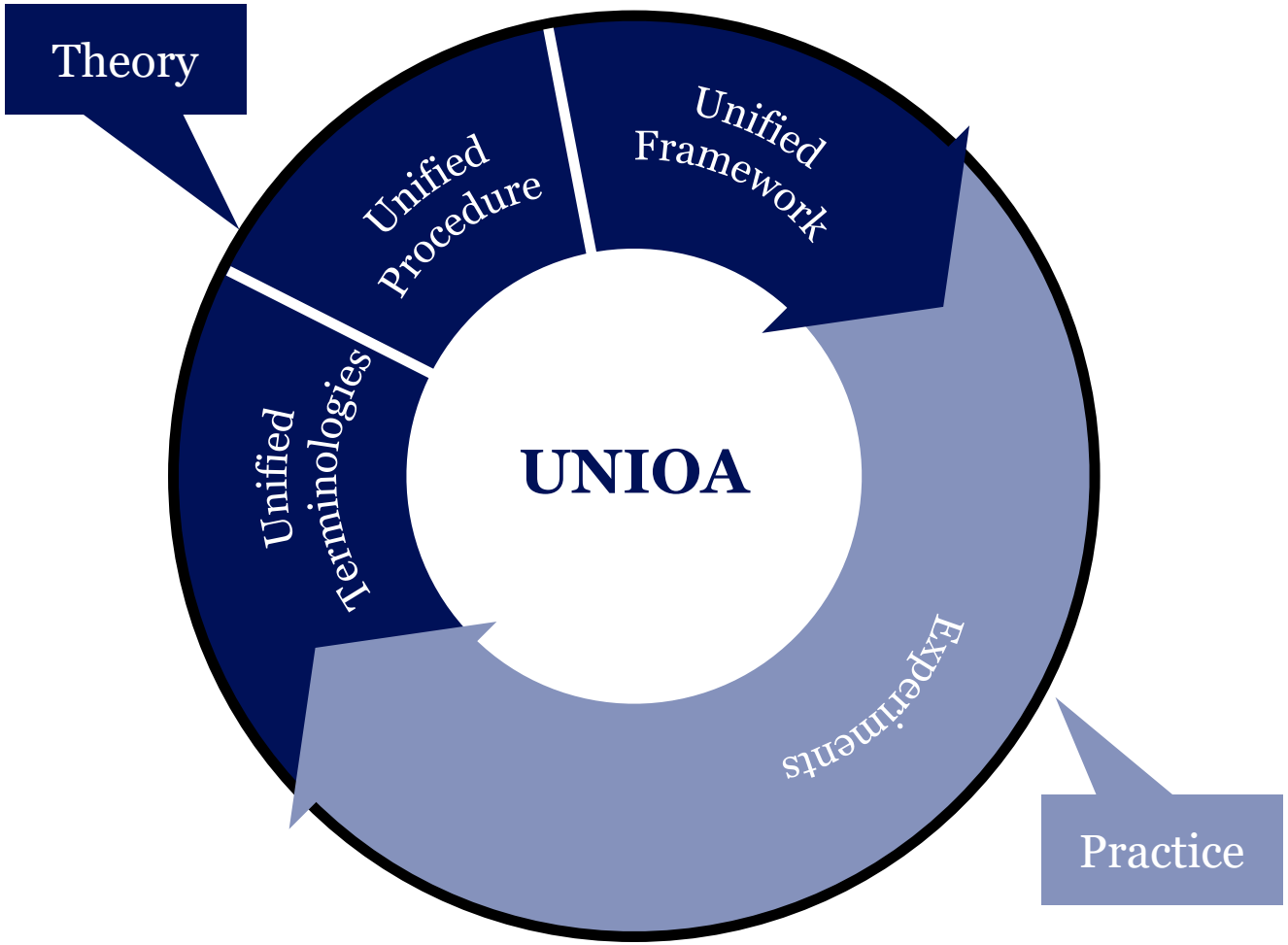


1. Motivation

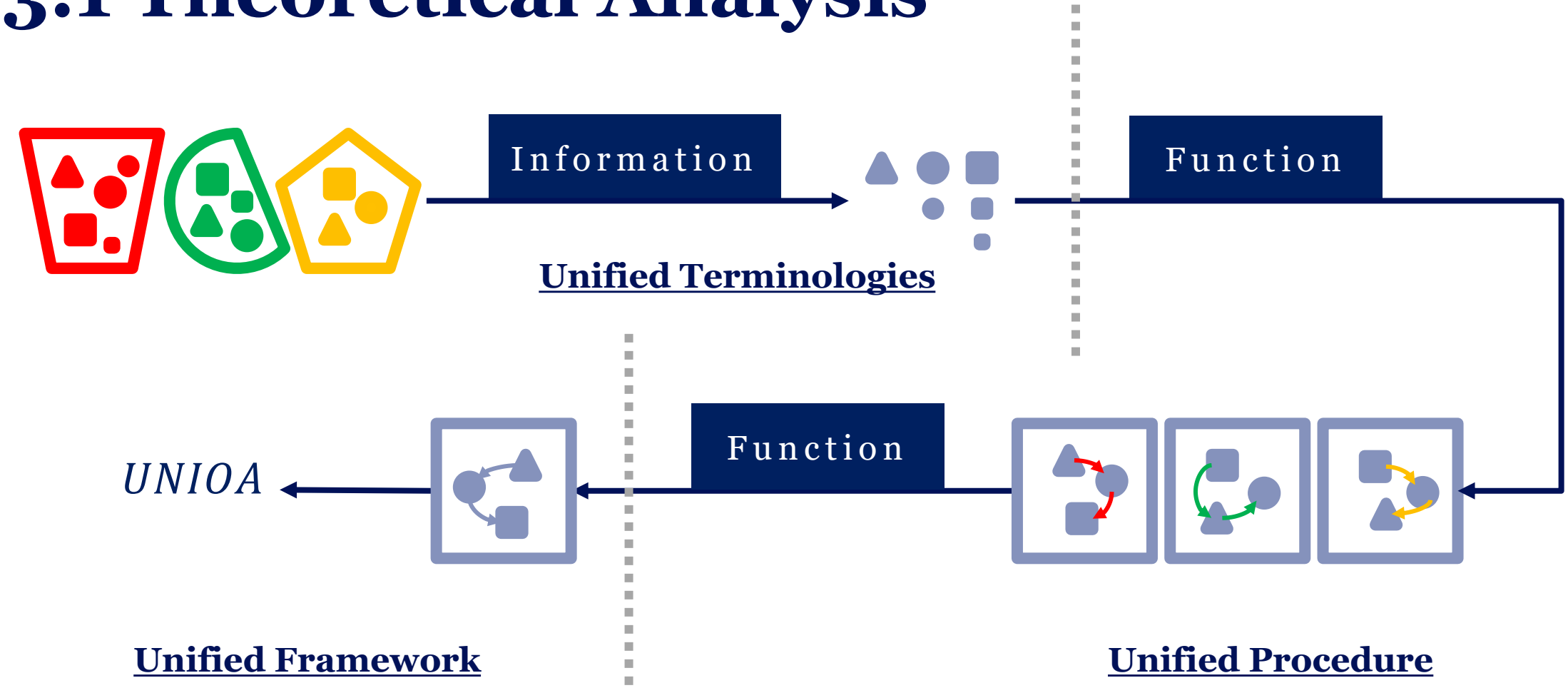


2. Outline of research

- **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**

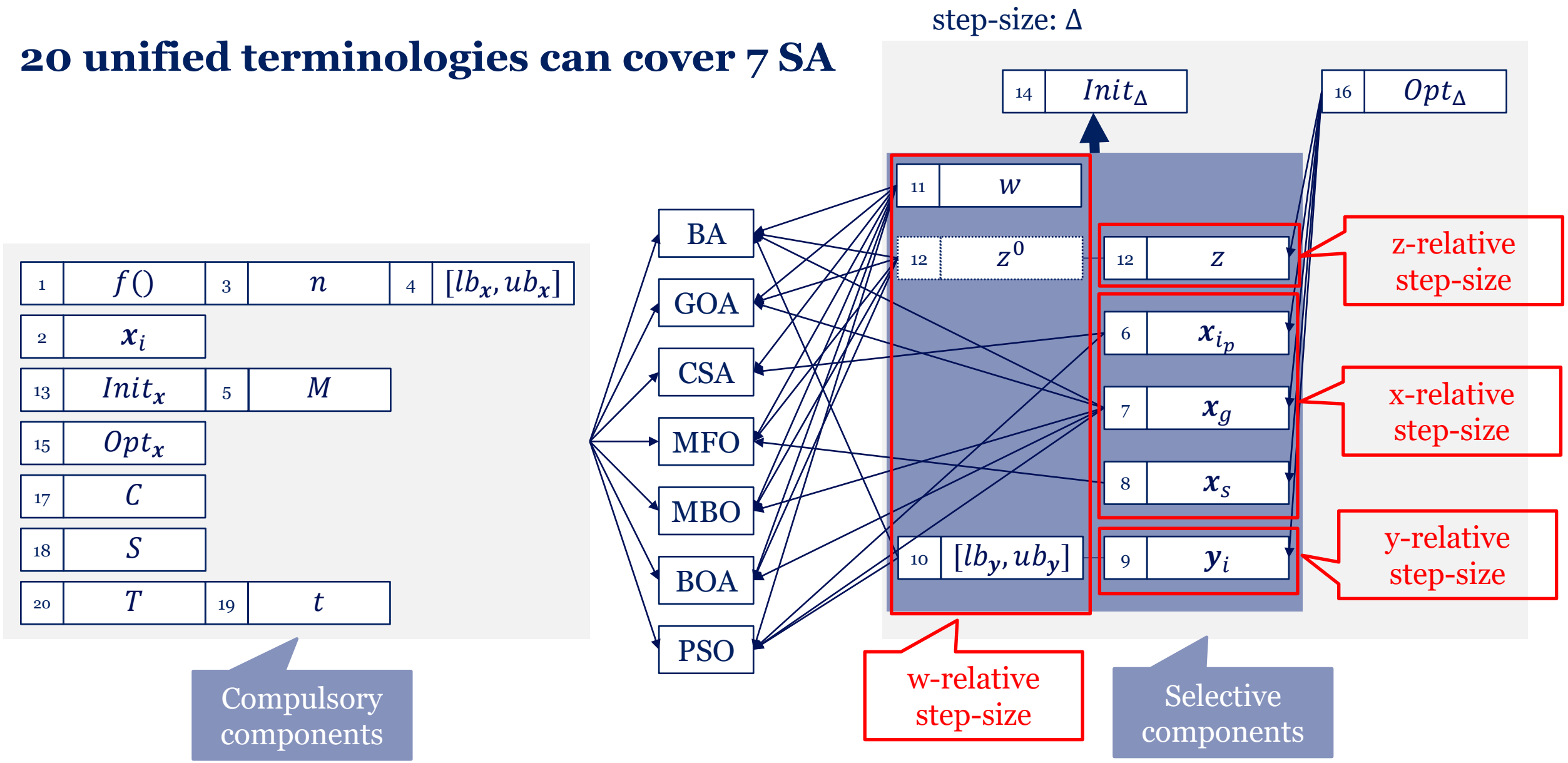
2.One possible objective solution.	•BA: each bat $\mathbf{x}_i$. •GOA: each grasshopper $\mathbf{x}_i$. •CSA: crow $\mathbf{x}_i$. •MFO: moth $\mathbf{m}_i$. •MBO: monarch butterfly $\mathbf{x}_i$. •BOA: butterfly $\mathbf{x}_i$. •PSO: particle $\mathbf{x}_i$.	$\mathbf{x}_i$: one objective solution.

Information

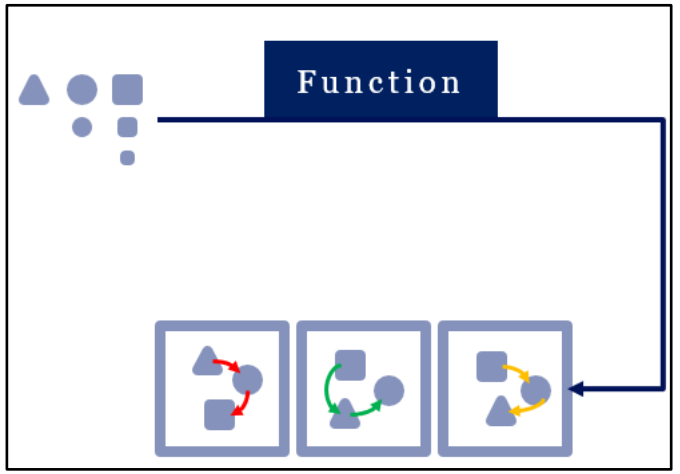
Various terminologies

Unified terminology

20 unified terminologies can cover 7 SA



✓ Unified Procedure



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

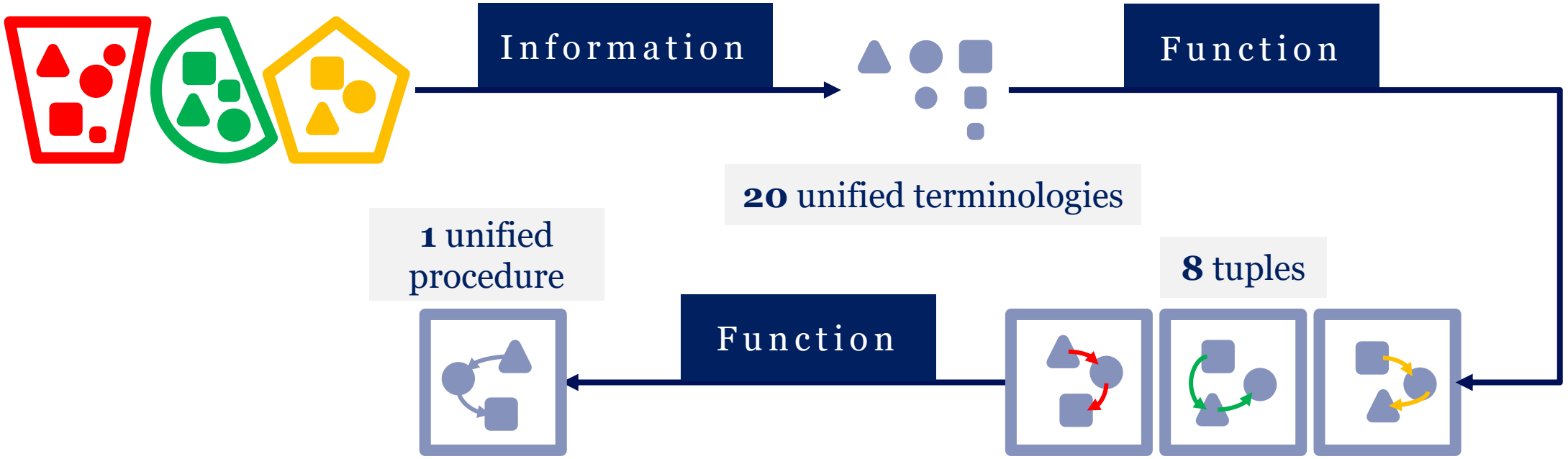
Tuple.2	$2.\mathbf{x}_i$ $3.n$ $4.lb_{\mathbf{x}}/ub_{\mathbf{x}}$ $5.M$ $13.Init_{\mathbf{x}}$	initialize the initial population.
Tuple	Components	Function

- **8-tuple**
- $NIOA = (f, Init_{\boldsymbol{x}}, Opt_{\boldsymbol{x}}, C, T, S, Init_{\Delta}, Opt_{\Delta})$
- **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* $Init_{\mathbf{x}}$ ▷ initialize population
3: *Tuple.1* f ▷ evaluation
4: *Tuple.3* $Init_{\Delta:w,z,\mathbf{x},\mathbf{y}}$ ▷ initialize w-relative, z-relative, $\mathbf{x}$-relative and $\mathbf{y}$-relative step-size Δ
5: **while** *Tuple.8* T **do** ▷ stop strategy
6: *Tuple.5* $Opt_{\Delta:\mathbf{y}}$ ▷ update $\mathbf{y}$-relative step-size Δ, if $\mathbf{y}$-relative step-size exsit in the initialization process
7: *Tuple.4* $Opt_{\mathbf{x}}$ ▷ update population
8: *Tuple.6* C ▷ outliers treatment
9: *Tuple.1* f ▷ evaluation
10: *Tuple.7* S ▷ selection
11: *Tuple.5* $Opt_{\Delta:z,\mathbf{x}}$ ▷ update z-relative and $\mathbf{x}$-relative step-size Δ
12: $t \leftarrow t + 1$
13: **end while**

✓ Unified Framework



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

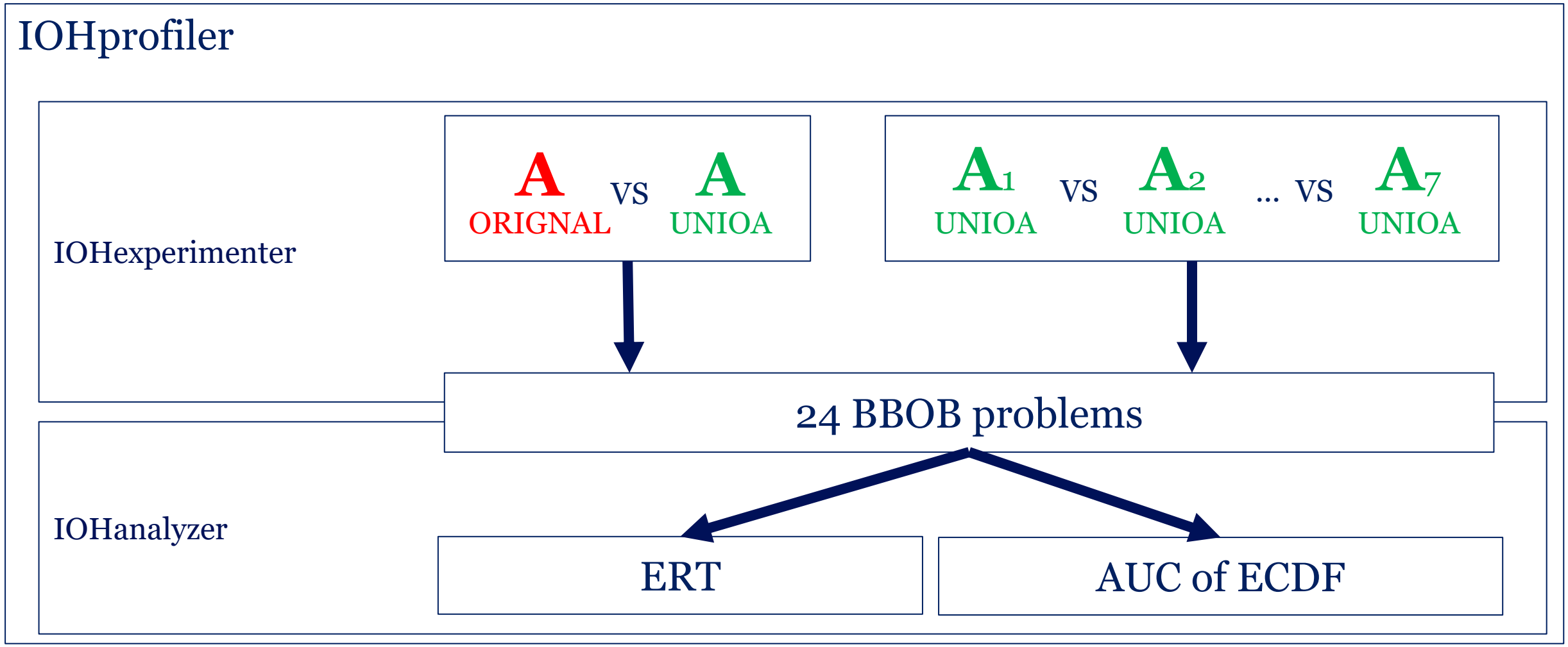
- **Unified framework *UNIOA* is constructed well**

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

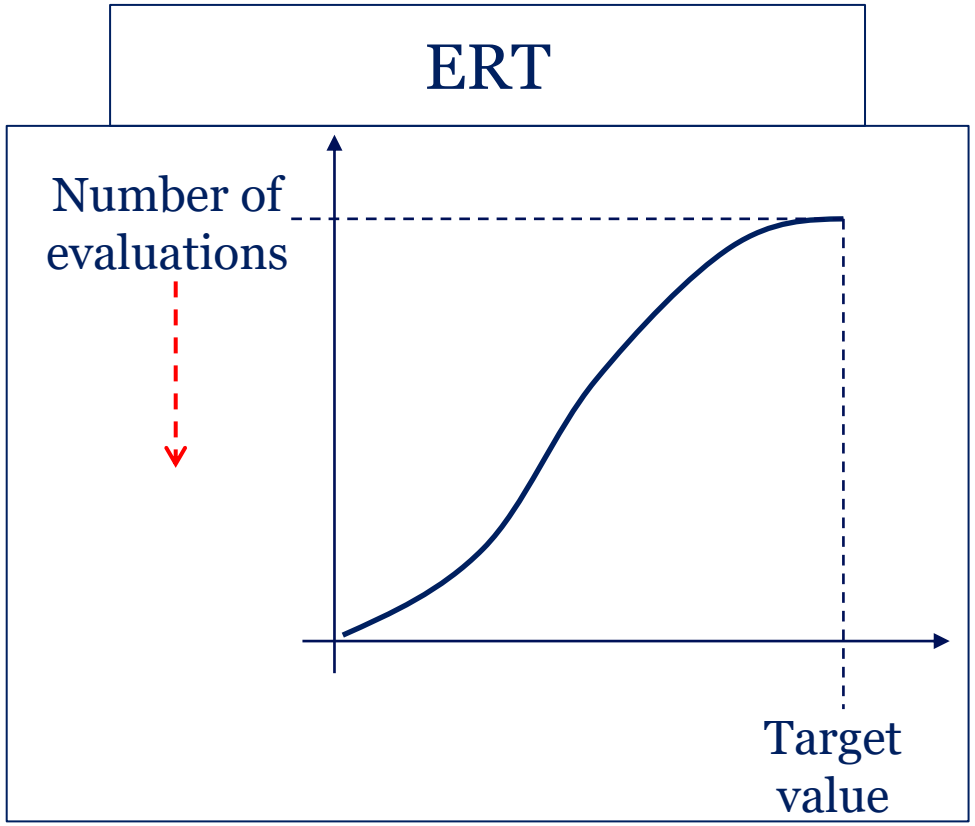
Algorithm 9 Unified Nature-Inspired Optimization Algorithm — UNIOA

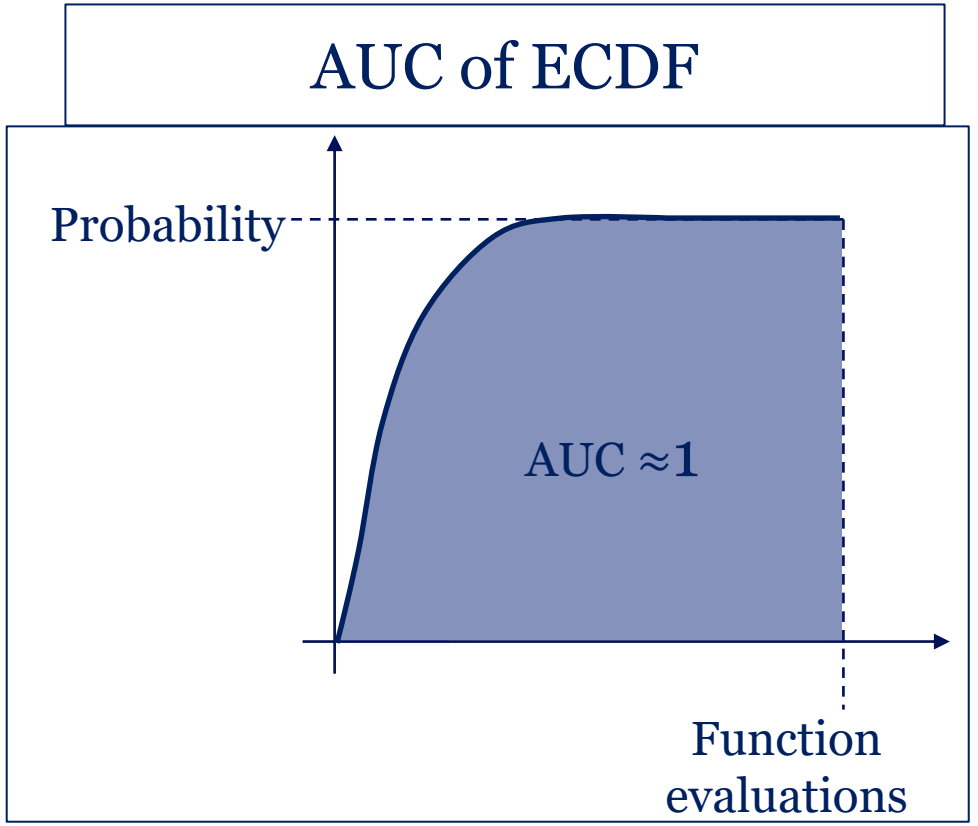
$t \leftarrow 0$
$\mathbf{X}(t) \leftarrow Init_{\mathbf{x}}(n, M, [lb_{\mathbf{x}}, ub_{\mathbf{x}}])$ ▷ initialize initial pop
$F(t) \leftarrow f(\mathbf{X}(t))$ ▷ evaluate
$\Delta_{w,z,\mathbf{y},\mathbf{x}}(t) \leftarrow Init_{\Delta:w,z,\mathbf{y},\mathbf{x}}(\mathbf{X}(t), w, z^0, [lb_{\mathbf{y}}, ub_{\mathbf{y}}], t)$ ▷ initialize step-size
while T **do**
  if $\mathbf{y} \in \Delta(t=0)$ **then**
    $\mathbf{Y}(t+1) \leftarrow Opt_{\Delta:\mathbf{y}}(\mathbf{Y}(t), w)$ ▷ update y-relative step-size
    $\hat{\mathbf{X}}(t+1) \leftarrow Opt_{\mathbf{x}}(\mathbf{X}(t), \mathbf{Y}(t+1), \Delta_{w,z,\mathbf{x}}(t))$ ▷ temporarily updated pop
    $\hat{\mathbf{X}}(t+1) \leftarrow C(\hat{\mathbf{X}}(t+1))$ ▷ outliers treatment
    $F(t+1) \leftarrow f(\hat{\mathbf{X}}(t+1))$ ▷ evaluate
    $\mathbf{X}(t+1) \leftarrow S(\mathbf{X}(t), \hat{\mathbf{X}}(t+1), \Delta_{w,z}(t))$ ▷ select and generate finally updated pop
    $\Delta_{z,\mathbf{x}}(t+1) \leftarrow Opt_{\Delta:z,\mathbf{x}}(\mathbf{X}(t), \mathbf{X}(t+1), z(t), t+1)$ ▷ update $z, \mathbf{x}$-relative step-size
  else
    $\hat{\mathbf{X}}(t+1) \leftarrow Opt_{\mathbf{x}}(\mathbf{X}(t), \Delta(t))$ ▷ temporarily updated pop
    $\hat{\mathbf{X}}(t+1) \leftarrow C(\hat{\mathbf{X}}(t+1))$ ▷ outliers treatment
    $F(t+1) \leftarrow f(\hat{\mathbf{X}}(t+1))$ ▷ evaluate
    $\mathbf{X}(t+1) \leftarrow S(\mathbf{X}(t), \hat{\mathbf{X}}(t+1), \Delta_{w,z}(t))$ ▷ select and generate finally updated pop
    $\Delta(t+1) \leftarrow Opt_{\Delta}(\mathbf{X}(t), \mathbf{X}(t+1), \Delta(t), t+1)$ ▷ update step-size
  end if
  $t \leftarrow t+1$
end while

3.2 Practical Analysis



3.2 Practical Analysis

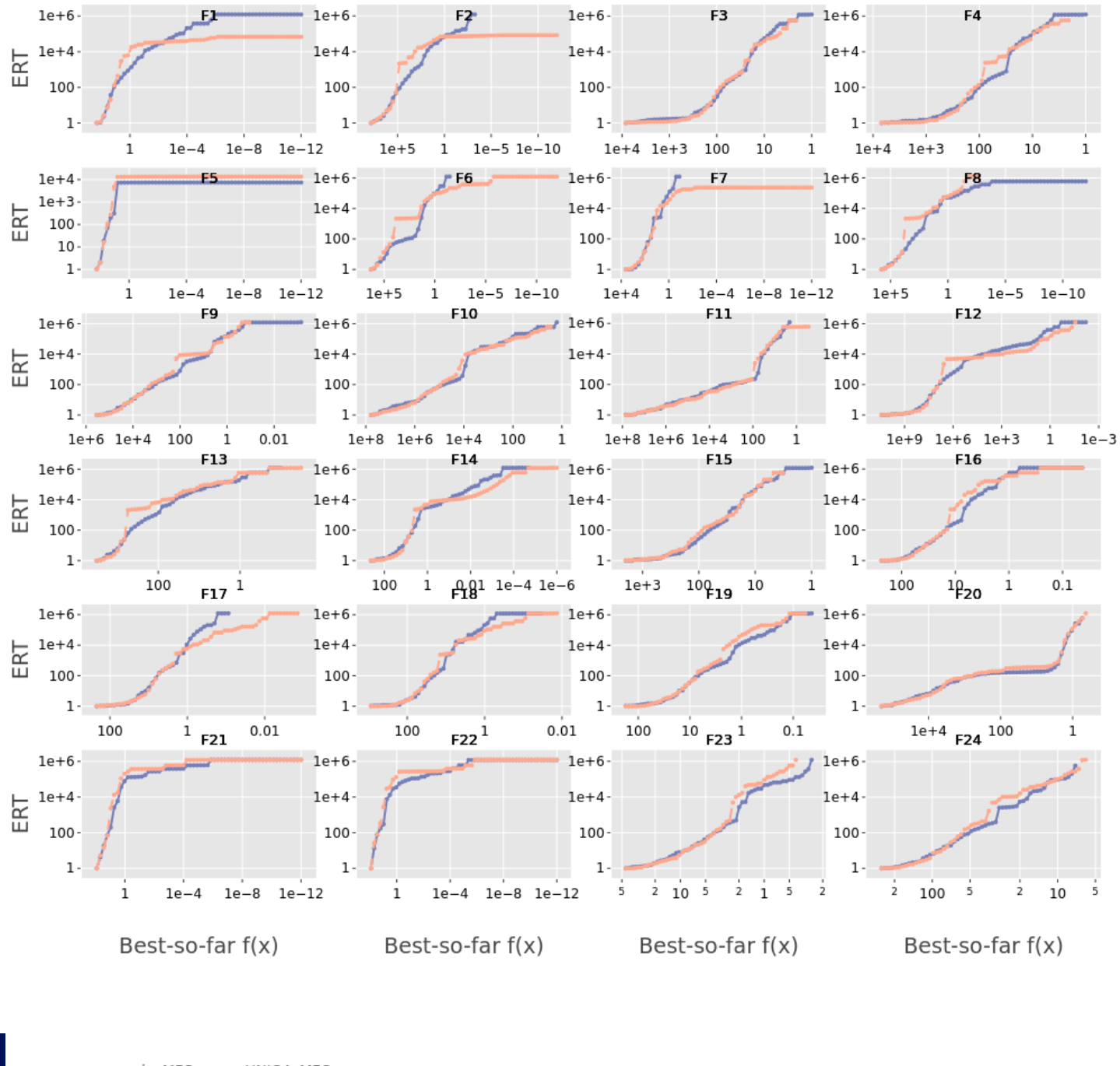




A ORIGNAL vs A UNIOA

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.

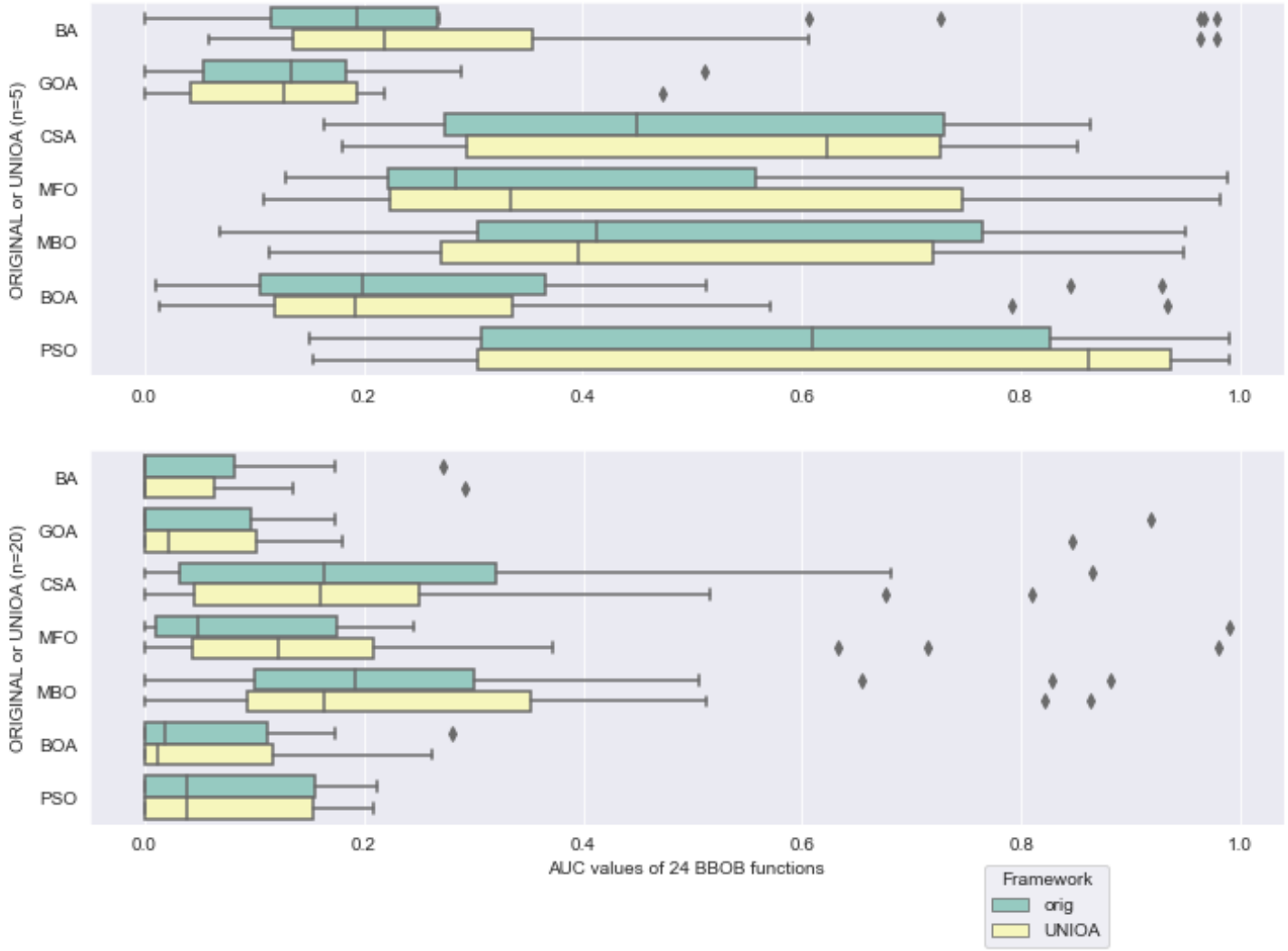


A vs A

ORIGNAL UNIOA

AUC values

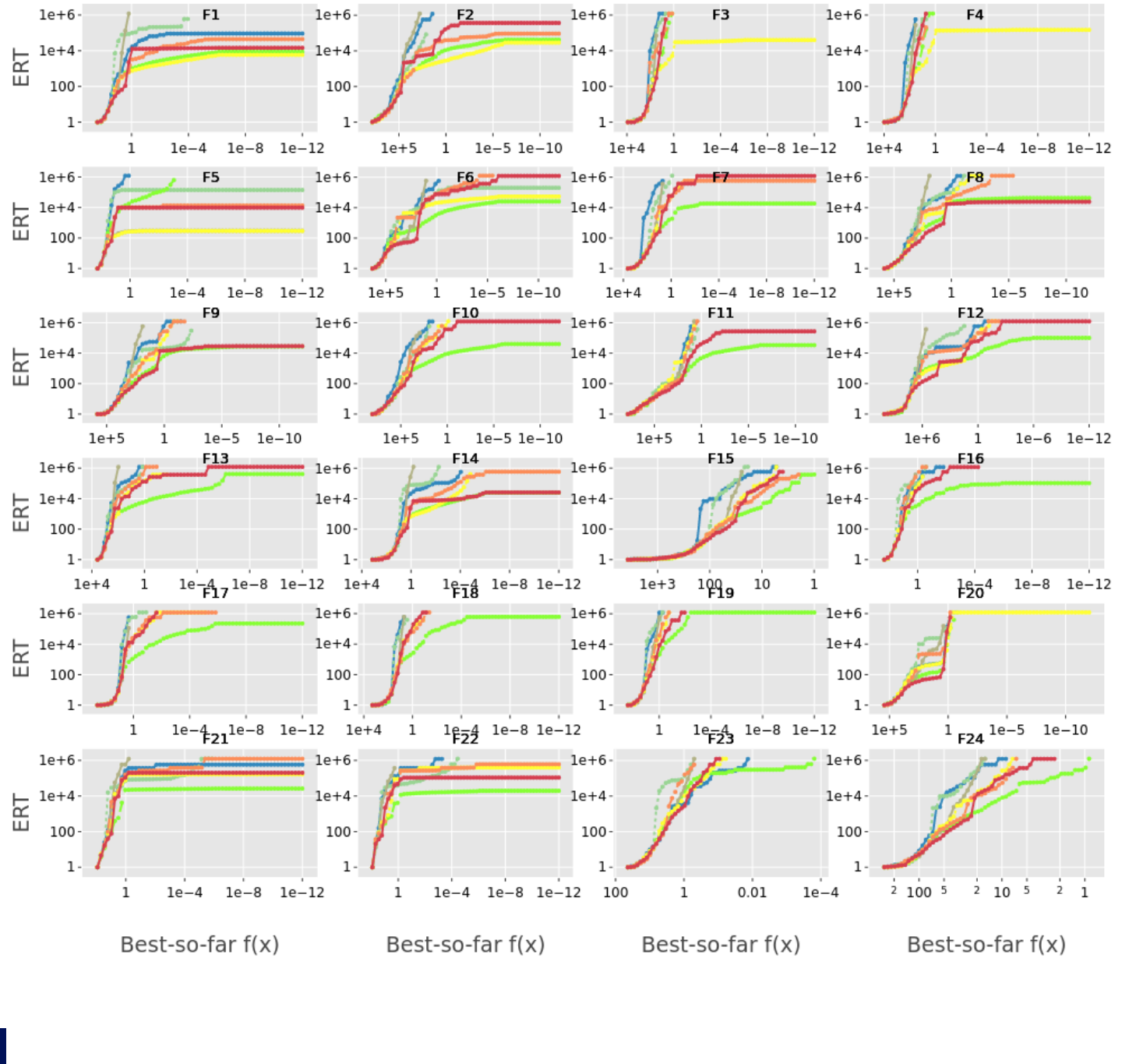
- Output:
  - ✓ Two boxes are close.
  - ✓ Accept Ho.
- Conclusion:
  - ✓ At least in these seven algorithms, UNIOA performs same as ORIGINAL also in the view of AUC values.



A_1 UNIOA vs A_2 UNIOA ... vs A_7 UNIOA

ERT plot

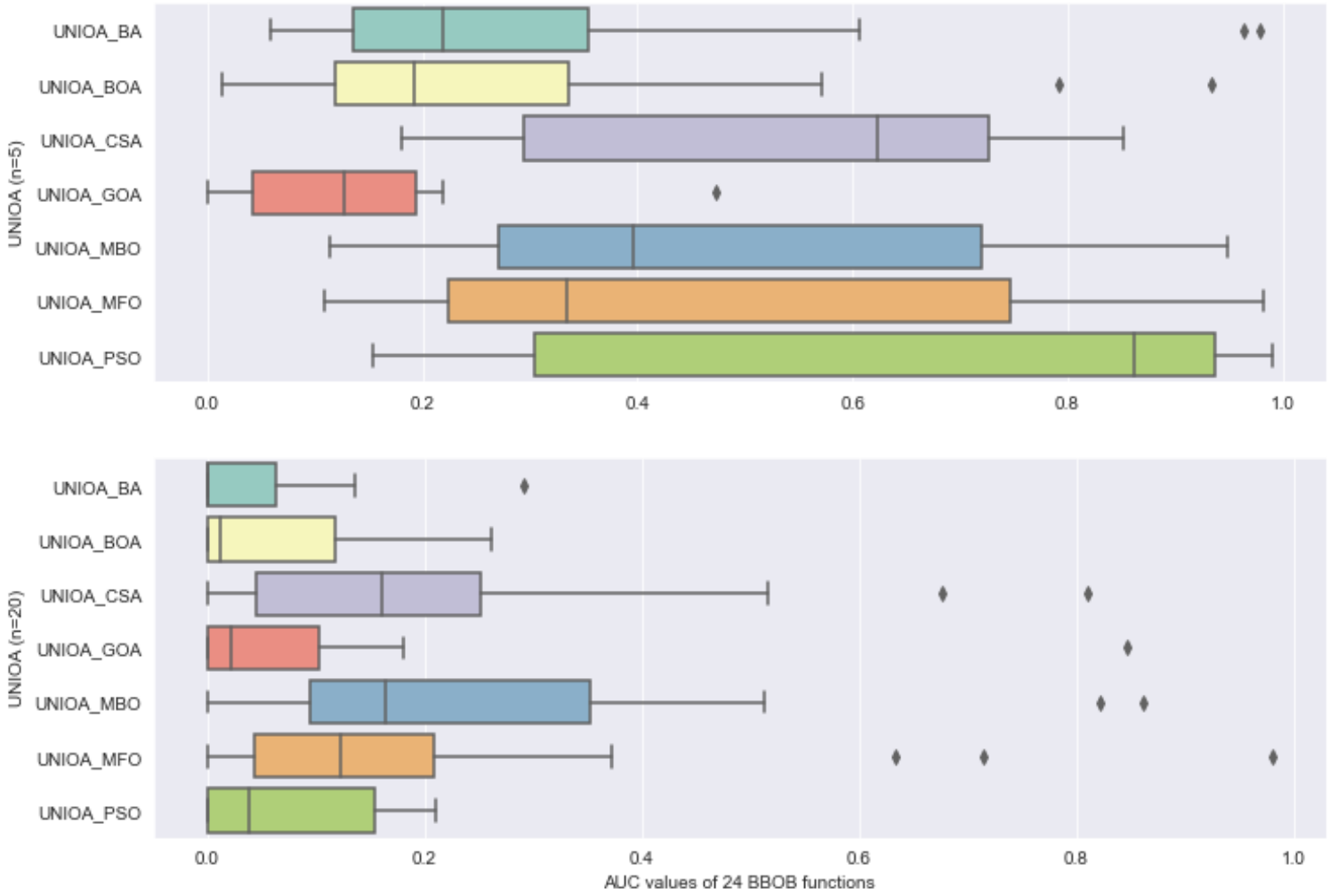
- Output:
  - ✓ ERT lines over 7 algorithms in UNIOA
- Conclusion:
  - ✓ CSA



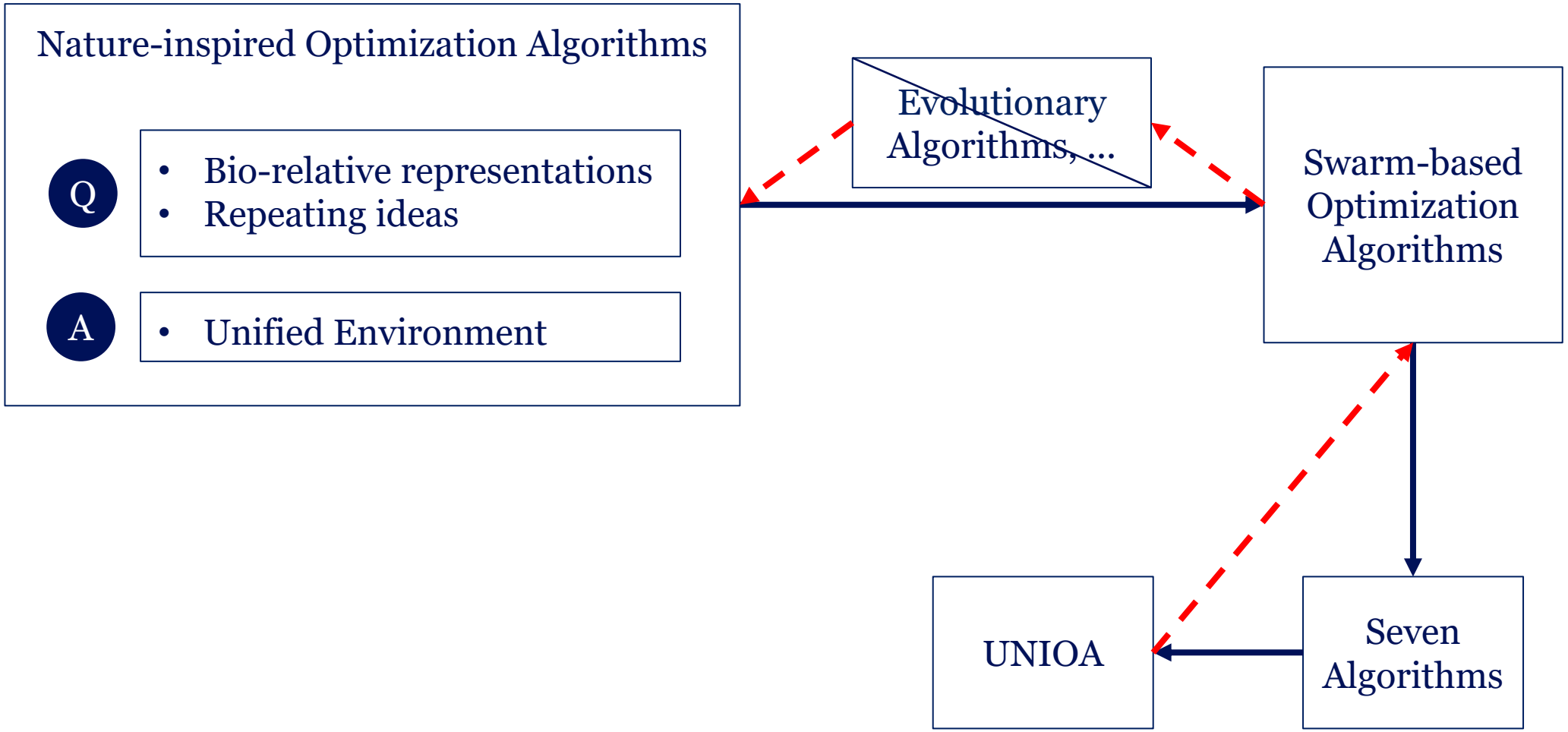
A_1 UNIOA vs A_2 UNIOA ... vs A_7 UNIOA

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