



学习辅助的进化优化

@ IEEE CIS 演化调度与组合优化 Taskforce

詹志辉

2023-07-31



背景介绍：进化计算与机器学习



学习辅助的进化计算

- LEO: Learning-aided Evolution Optimization
 - 群体演化宏知识纵向学习辅助的进化优化
- KLEC: Knowledge Learning for Evolutionary Computation
 - 个体方向微知识平向学习辅助的进化优化
- MKTDE: Meta-Knowledge Transfer-based Differential Evolution
 - 跨越问题元知识横向学习辅助的进化优化



总结

1、学习与优化

- **学习思考**和**进化优化**是人类活动智能化的主要特征，也是人类智能产生和发展的重要途径

人类智能的产生和发展

学习思考

- 对经验、数据的统计、分析、学习、.....
- 通过知识学习、深度思考等形成新的知识和能力，掌握对世界的认知能力和改造能力

.....

进化优化

- 对行为、过程的控制、演化、优化、.....
- 通过自然演化、群体协作等来提升智能水平，通过智能优化来延伸和拓展人类智能

.....

1.1、最优化与智能化问题

- 人类活动从盲目性、随机性到最优化、智能化
- 最优化（Optimization）
 - 是人类智能活动的核心
 - 出现在生产实践、科学研究和社会活动的众多问题中
 - 可以归结为最优化问题，以实现智能化



智能交通控制

在多个约束条件下
最小化交通拥堵情况



电子电路设计

在给定电子元件上
最优化电路效果



传感器网络布署

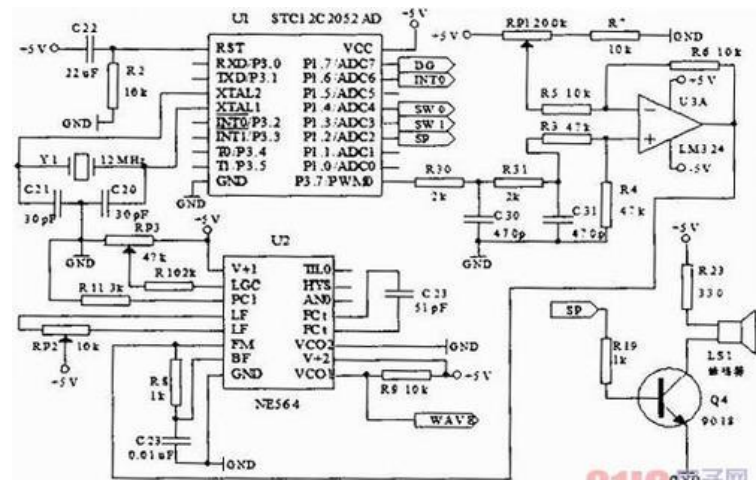
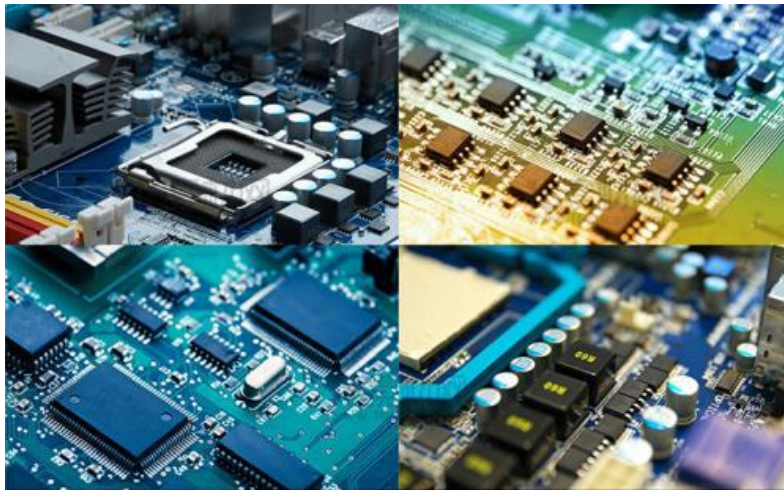
在满足覆盖目标的前提下
最大化寿命、最小化成本

.....

1.1、最优化与智能化问题

□ 例子：电子电路设计

- 众多电子元器件：电阻、电容、电感
- 关系复杂、耦合、动态、大规模、非线性、.....
- 大量局部最优解，难以找到最优解
- 计算一次适应值昂贵（耗材料、耗时间、.....）



1.2、群体与进化计算

智能化和最优化是全球迈向智能时代的重大需求

智能制造
智能设计
智能物流



存在

复杂
优化问题

数学模型
难以精确表达



传统优化方法

- 依赖问题的数学特性进行求解
- 难以适用于复杂优化问题

进化计算优化算法

- 模拟自然界群体智能的优化算法
- 不依赖于问题的数学特性
- 简单易用，快速得到最/次优解

成为求解众多智能化复杂优化问题的重要途径

Zhi-Hui Zhan(詹志辉)(Corresponding Author), L. Shi, K. C. Tan, and J. Zhang, “A Survey on Evolutionary Computation for Complex Continuous Optimization,” *Artificial Intelligence Review*, vol. 55, no. 1, pp. 59-110, Jan. 2022.

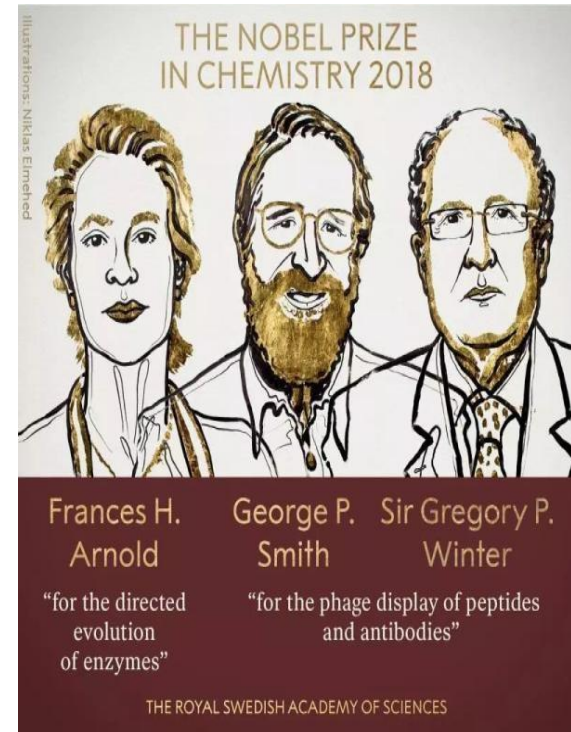
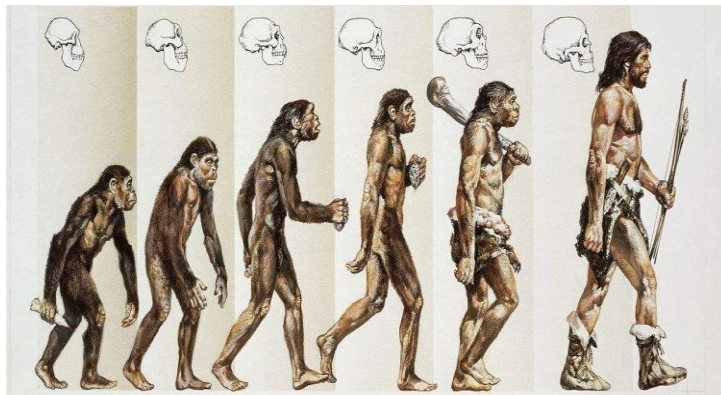
Z. G. Chen(学生), Zhi-Hui Zhan(詹志辉)(Corresponding Author), et al., “Evolutionary Computation for Intelligent Transportation in Smart Cities: A survey,” *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 83-102, May, 2022.

1.2、群体与进化计算

□ 模拟自然界的进化现象和生物的群体智能行为

进化是世界上最强大的工程方法，我们应该利用它来寻找解决问题的新的生物解决方案

-----Frances H. Arnold (阿诺德)
2018年诺贝尔奖获得者



在计算机科学领域，智能算法是人工智能中涉及到最优化问题的一个子域。其算法是受生物进化过程中“优胜劣汰”的自然选择机制和遗传信息的传递规律的影响，通过程序迭代模拟这一过程，把要解决的问题看作环境，在一些可能的解组成的种群中，通过自然演化寻求最优解。

----- 百度百科

1.2、群体与进化计算

□ 模拟自然界的进化现象和生物的群体智能行为

帕里西专门研究过鸟群的波动。故事源之于有一天他在罗马的火车站上空看到了成千上万只鸟儿成群结队地飞翔，鸟群没有统一的指挥却能如同一个整体自由变换形状，这一现象让他着迷。帕里西于是派出了一组物理学家，他们拍摄记录了大量的鸟群飞行数据。-----乔治 帕里西 (Giorgio Parisi)



鸟群飞行与觅食行为



蚁群活动与觅食行为

2021年诺贝尔奖获得者



粒子群优化算法 (Particle Swarm Optimization, PSO) 是计算科学家通过模拟鸟类群体觅食行为表现出来的智能现象，而设计出的对复杂系统进行优化的人工智能 (群体智能) 算法。--张军、詹志辉等编著 《计算智能》

1.2、群体与进化计算



生物进化



蚁群觅食行为

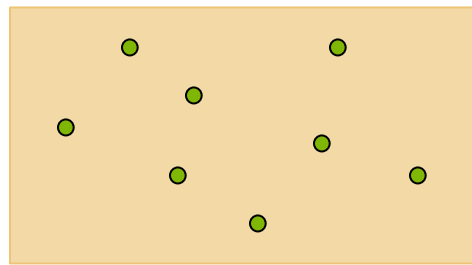
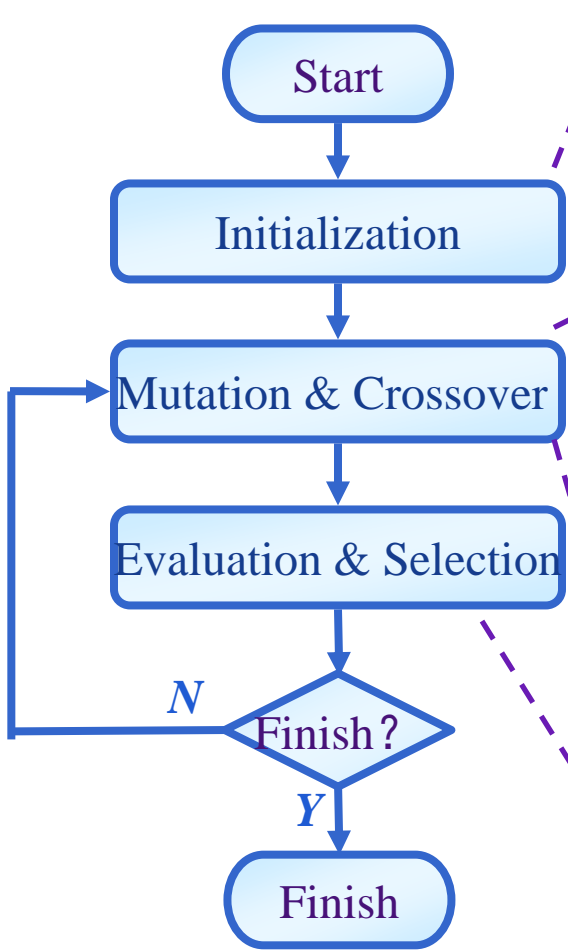


鸟类捕食行为



1.2、群体与进化计算

差分进化 (Differential Evolution, DE)



Randomly generate N individuals
 individual: $X_i = [X_{i1}, X_{i2}, \dots, X_{iD}]$

$$V_i = X_{best} + F(X_{r1} - X_{r2})$$

individual: $X_i = [X_{i1}, X_{i2}, \dots, X_{iD}]$

$$V_i = X_{r1} + F(X_{r2} - X_{r3})$$

mutant vector: $V_i = [V_{i1}, V_{i2}, \dots, V_{iD}]$

offspring vector: $U_i = [U_{i1}, U_{i2}, \dots, U_{iD}]$

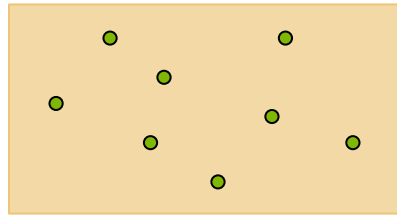
$$u_{id} = \begin{cases} v_{id}, & \text{if } rand(0,1) \leq CR \text{ or } d = ri, \\ x_{id}, & \text{otherwise} \end{cases}$$

F is an 'amplification factor', CR is a 'crossover rate'

If $fitness(U_i)$ is better than $fitness(X_i)$,
 Then $X_i = U_i$

1.2、群体与进化计算

粒子群优化 (Particle Swarm Optimization, PSO)



Randomly generate N particles

position: $X_i = [X_{i1}, X_{i2}, \dots, X_{iD}]$

velocity: $V_i = [V_{i1}, V_{i2}, \dots, V_{iD}]$

personal best: $pBest_i = [P_{i1}, P_{i2}, \dots, P_{iD}]$

global best: $gBest = [G_1, G_2, \dots, G_D]$

$$V_{id} = \omega V_{id} + c_1 r_{1d} (P_{id} - X_{id}) + c_2 r_{2d} (G_d - X_{id})$$

$$X_{id} = X_{id} + V_{id}$$

ω inertia weight, c_1, c_2 acceleration coefficients

If $fitness(X_i)$ is better than $fitness(pBest_i)$,
Then $pBest_i = X_i$

If $fitness(pBest_i)$ is better than $fitness(gBest)$,
Then $gBest = pBest_i$

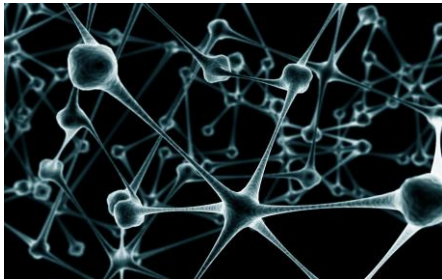
1.4、机器学习与进化计算

通过计算机从经验和数据中进行学习

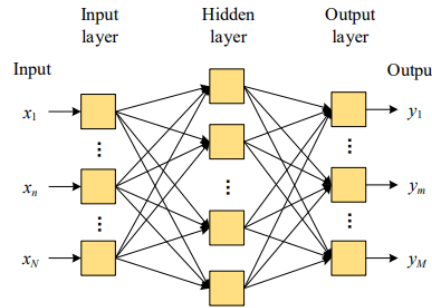
- 随机森林、支持向量机、人工神经网络（深度学习）

人工神经网络（Artificial Neural Network, ANN）是科学家们通过借助数学和物理的方法从信息处理的角度对人脑神经网络进行抽象后建立的简化模型。

--张军、詹志辉等编著 《计算智能》



人脑神经网络



人工神经网络

神经网络和进化计算是人工智能领域的两大分支，联系紧密，包括基于机器学习的进化计算、进化深度学习……

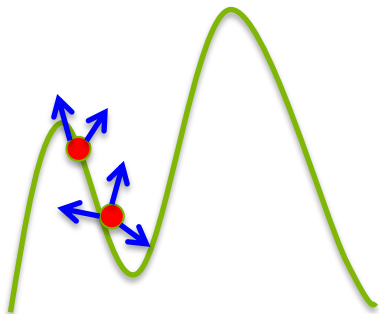
Zhi-Hui Zhan (詹志辉)(通讯作者), et al., “Evolutionary deep learning: A survey,” *Neurocomputing*, vol. 483, pp. 42–58, 2022.

J. Zhang, Zhi-Hui Zhan (詹志辉), et al., “Evolutionary computation meets machine learning: A survey,” *IEEE Computational Intelligence Magazine*, vol. 6, no. 4, pp. 68–75, 2011.

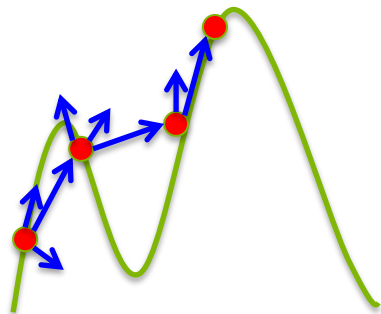
J. Y. Li(Student), Zhi-Hui Zhan(詹志辉)(通讯作者), et al., “Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameter optimization in convolution neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 5, pp. 2338 – 2352, May 2023.

1.5、科学问题与研究思路

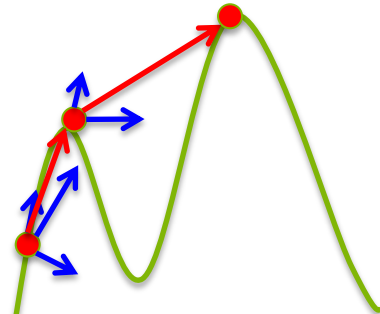
研究动机：学习辅助进化计算



随机方向进行搜索
以前



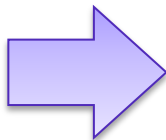
问题适应值引导搜索
现在



知识学习引导搜索
将来

传统搜索和进化计算

- 搜索或进化过程带有随机性或通常仅依靠适应值引导驱动，**全局搜索能力弱，容易落入局部最优**
- 使用单代种群信息，仍然**难以利用历史和全局搜索信息**，导致**收敛速度慢**，需要进一步加速算法收敛



学习辅助的进化计算

- 使用**机器学习进行辅助**，从而增强算法在各类不同问题中的普遍适用性和全局搜索能力
- 促进演化过程中群体数据和跨问题群体数据关联融合，有助于更高效、更深度的**搜索信息学习共享**，**加快算法收敛**

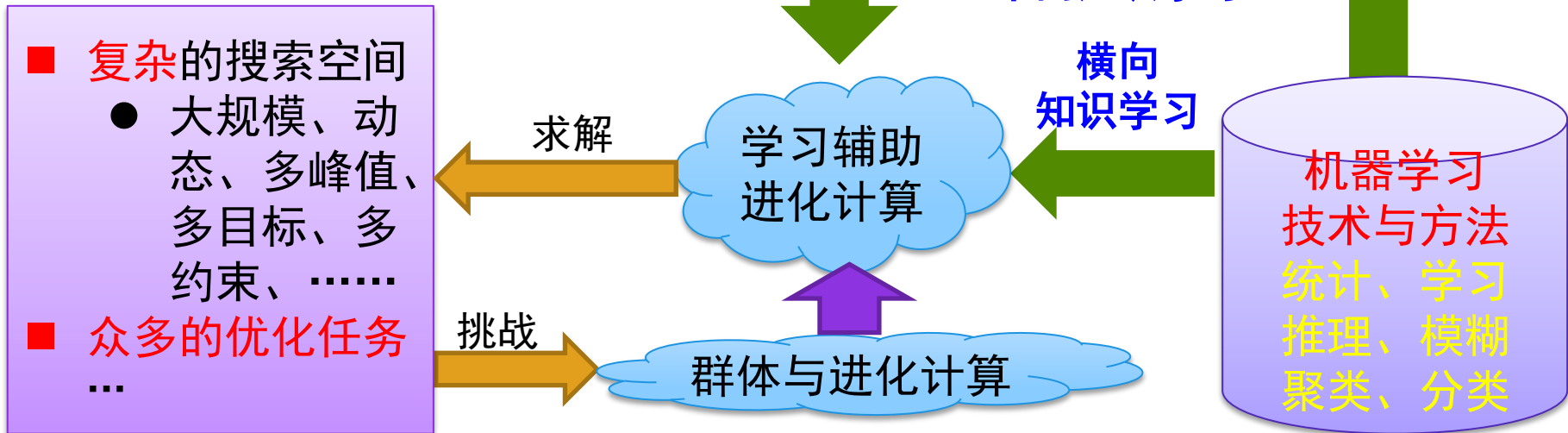
1.5、科学问题与研究思路

两个关键科学问题

学习什么？（宏知识、微知识、元知识）

如何学习？（纵向学习、平向学习、横向学习）

研究思路



1.6、学习辅助的进化优化

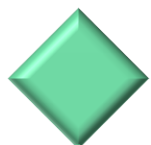
我们的工作

- 提出了群体演化宏知识纵向学习、个体方向微知识平向学习和跨越问题元知识横向学习辅助的进化计算优化方法
- 实现了算法的全局搜索能力增强和提高了多任务优化效能
 - **LEO**: Learning-aided Evolution for Optimization
 - **KLEC**: Knowledge Learning for Evolutionary Computation
 - **MKTDE**: Meta-Knowledge Transfer-based Differential Evolution

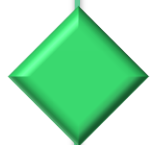
Zhi-Hui Zhan(詹志辉)(第一作者), *et al.*, “Learning-aided evolution for optimization,” *IEEE Transactions on Evolutionary Computation*, DOI:10.1109/TEVC.2022.3232776, Dec. 2022.

Y. Jiang (Student), Zhi-Hui Zhan (詹志辉)(通讯作者), *et al.*, “Knowledge learning for evolutionary computation,” *IEEE Transactions on Evolutionary Computation*, DOI: 10.1109/TEVC.2023.3278132, May 2023.

J.-Y. Li (Student), Zhi-Hui Zhan (詹志辉)(通讯作者), *et al.*, “A meta-knowledge transfer-based differential evolution for multitask optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 719-734, Aug. 2022.

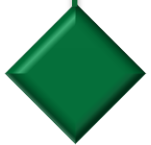


背景介绍：进化计算与机器学习



学习辅助的进化计算

- **LEO: Learning-aided Evolution Optimization**
 - 群体演化宏知识纵向学习辅助的进化优化
- **KLEC: Knowledge Learning for Evolutionary Computation**
 - 个体方向微知识平向学习辅助的进化优化
- **MKTDE: Meta-Knowledge Transfer-based Differential Evolution**
 - 跨越问题元知识横向学习辅助的进化优化



总结



2.1、LEO：学习辅助的进化优化

群体进化宏知识纵向学习辅助的进化优化

- **Zhi-Hui Zhan (詹志辉), et al.**, “Learning-aided evolution for optimization” *IEEE Transactions on Evolutionary Computation*, DOI:10.1109/TEVC.2022.3232776, Dec. 2022.

2.1、LEO：学习辅助的进化优化

问题提出

如何从群体进化过程的数据中获取知识，辅助优化？

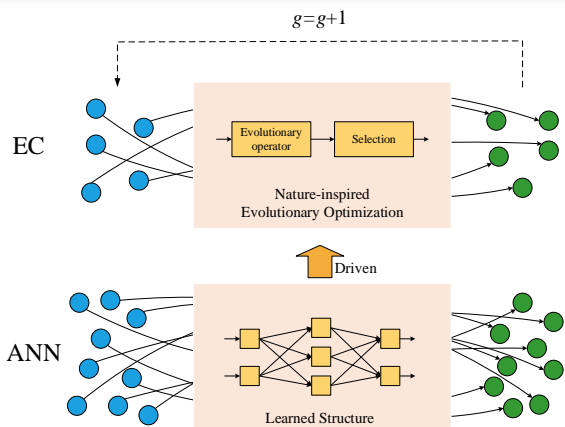
难点所在

- 如何学习进化知识
- 如何利用学习得到的进化知识

主要研究内容

- 建立学习系统对进化知识进行学习
- 提出学习辅助的进化算子

学习辅助的进化优化算法(Learning-aided Evolution for Optimization, LEO)

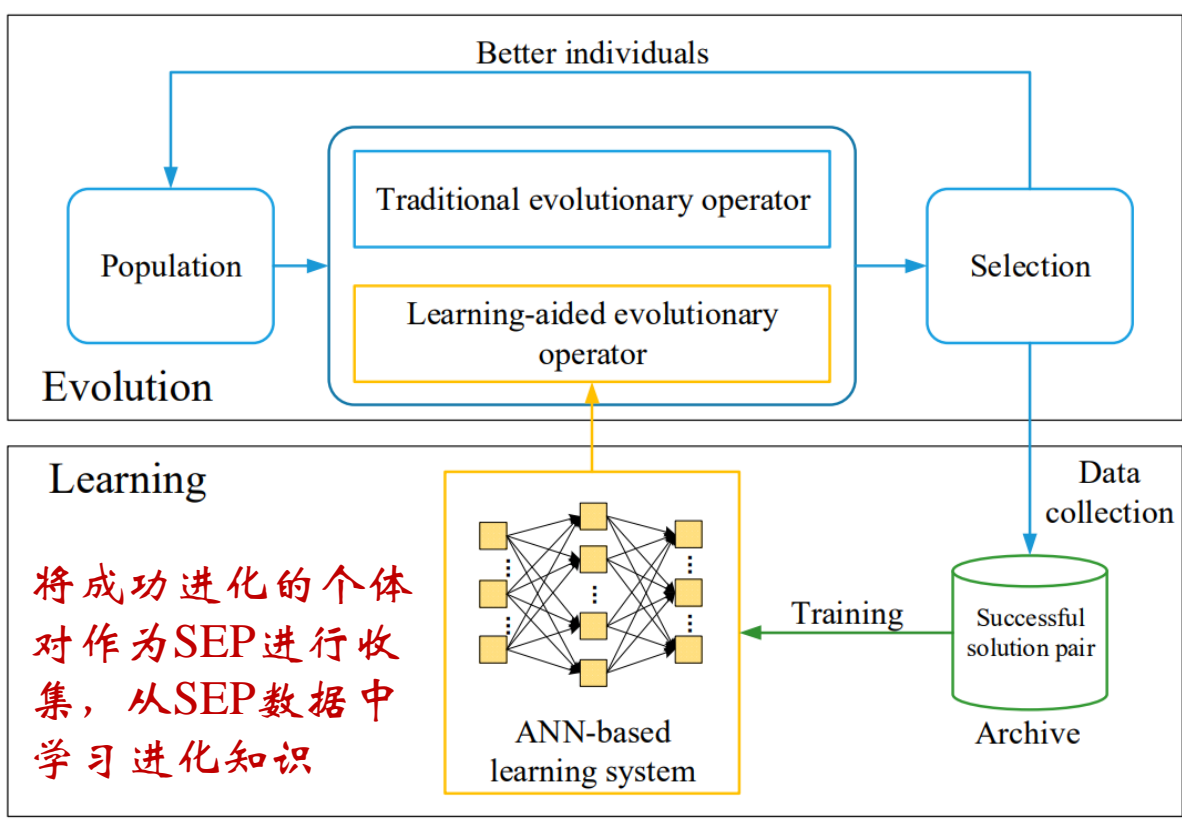


LEO

- 使用神经网络作为学习系统对进化知识进行学习
- 使用学习辅助的进化算子基于进化知识产生更好的解

2.1、LEO: 进化知识学习

Learn evolutionary knowledge from successful evolution pair



将成功进化的个体对作为SEP进行收集，从SEP数据中学习进化知识

$(X_{g,i}, X_{g+1,i})$ is defined as a successful evolution pair (SEP) if the fitness of $X_{g+1,i}$ is better than $X_{g,i}$, where g is the generation index.

Collect SEP in an archive every generation, if the archive reaches the maximum size, the old SEPs will be dropped

Use the collected SEPs in the archive to train the ANN-based learning system

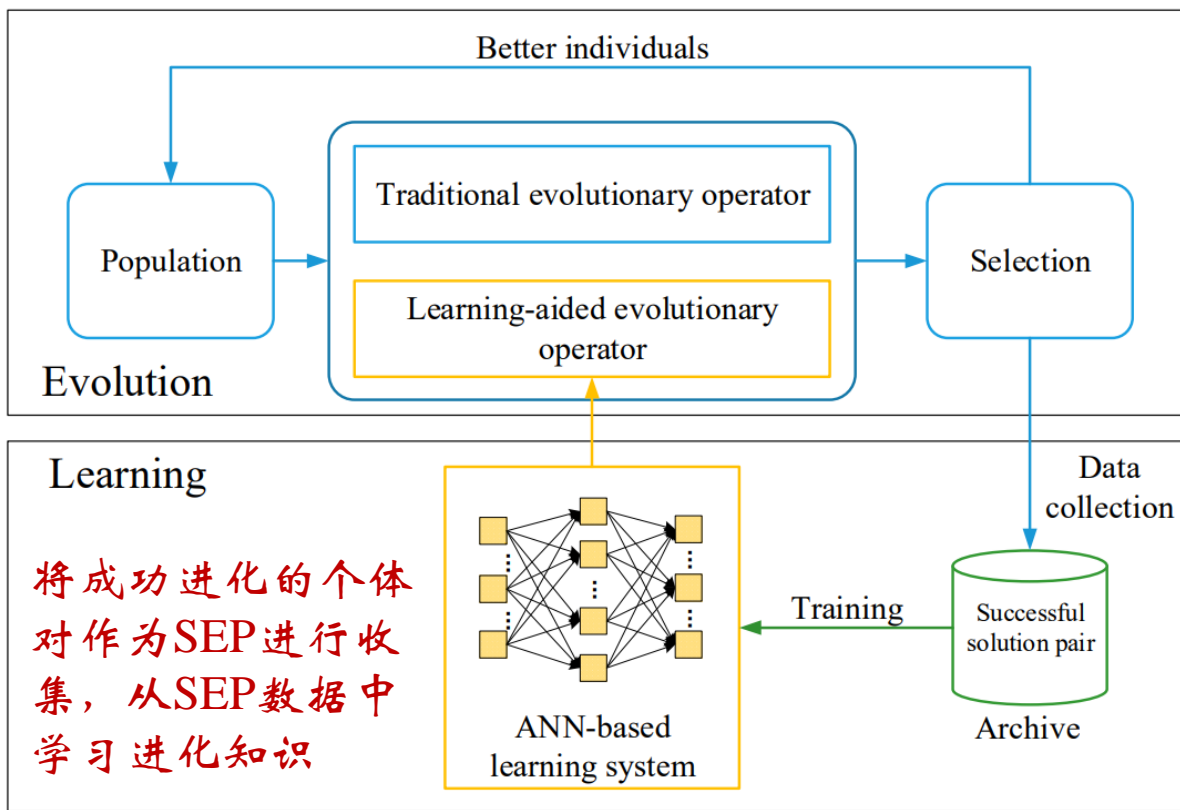
网络的损失函数 $L(X_i)$ 为网络输出

$$MSE = \frac{1}{arch_size} \sum_{(X_i, X'_i) \in arch} |L(X_i) - X'_i|^2$$

网络激活函数 $\sigma(z) = \frac{1}{1 + e^{-z}}$

2.1、LEO：进化知识学习

从成功进化配对中学习进化知识



记 g 为演化代数，如果 $X_{g+1,i}$ 的适应值 $X_{g,i}$ 比好，则 $(X_{g,i}, X_{g+1,i})$ 记为一组成功进化配对 (Successful Evolution Pair, SEP).

每一代收集 SEP 并储存到 Archive 中。如果达到 Archive 的储存上限，初期收集的 SEP 会被丢弃。

使用 Archive 中的 SEP 作为数据对基于 ANN 的学习系统进行训练

将成功进化的个体对作为 SEP 进行收集，从 SEP 数据中学习进化知识

网络的损失函数 $L(X_i)$ 为网络输出

$$MSE = \frac{1}{arch_size} \sum_{(X_i, X'_i) \in arch} |L(X_i) - X'_i|^2$$

网络激活函数 $\sigma(z) = \frac{1}{1 + e^{-z}}$

2.1、LEO：进化知识利用

Learning-aided evolutionary operator (学习辅助的进化算子)

Learning-aided mutation (LM) 和 Learning-aided crossover (LC) 算子

$$\begin{aligned}
 newX_m &= LM(X_A, X_B, X_C) \\
 &= L(X_A) + \alpha \cdot (X_B - X_C)
 \end{aligned}
 \quad \text{where } newX_j = \begin{cases} X_{i,j}, & \text{if } rand_j \leq \beta \\ newX_{m,j}, & \text{otherwise} \end{cases}$$



PSO的学习辅助进化算子

$$\begin{aligned}
 newX_i &= LM(pBest_i, pBest_{r_1}, pBest_{r_2}) \\
 newX_i &= LC(pBest_i, newX_i)
 \end{aligned}$$

DE的学习辅助进化算子

$$\begin{aligned}
 newX_i &= LM(X_i, X_{r_1}, X_{r_2}) \\
 newX_i &= LC(X_i, newX_i)
 \end{aligned}$$

2.1、LEO：整体算法

➤ 伪代码

Algorithm 1: LEO-based EC algorithm

```
1:Begin
2: // Initialization
3:  $g \leftarrow 1$ ; // the generation index
4: Initialize population  $X_g$  and evaluate the fitness;
5: Initialize the weights of ANN randomly;
6: Initialize  $arch$  as an empty set; // to store solution pairs
7: While stop criteria not satisfied Do
8:    $g \leftarrow g + 1$ ;
9:   // Individual Evolution
10:  Sample  $r$  uniformly on  $[0,1]$ ;
11:  If  $g > 1$  and  $r < lp$  Then
12:     $newX \leftarrow$  Evolve  $X_g$  by learning-aided evolutionary operator;
13:  Else
14:    // use operators in traditional EC, e.g., PSO or DE
15:     $newX \leftarrow$  Evolve  $X_g$  by traditional evolutionary operator;
16:  End If
17:  Evaluate the fitness of individuals in  $newX$ ;
18:  // Selection
19:   $X_{g+1} \leftarrow$  selection among  $X_g$  and  $newX$ ;
20:  // SEP Collection
21:  For each individual  $i$  in  $X_{g+1}$  Do
22:    If  $X_{g+1,i}$  is better than  $X_{g,i}$  Then
23:      Add  $(X_{g,i}, X_{g+1,i})$  in  $arch$ ;
24:    End
25:  End For
26:  If number of SEPs in  $arch > arch\_size$  Then
27:     $Arch \leftarrow$  the newest  $arch\_size$  solution pairs;
28:  End If
29:  // Learning System Update
30:  Train the ANN with all data in  $arch$  for one epoch;
31: End While
32:End
```

特点与优势

动态进化知识学习

每一代都收集最新SEPs并对学习系统进行训练,并定期丢弃较早收集的SEPs



高效进化知识利用

通过学习辅助的演化算子,将学习系统的预测输出作为辅助数据,产生更好的个体

2.1、LEO：实验

IEEE CEC 2021 Single-Objective Real Parameter Numerical Optimization Competition

	No.	Functions	$F_i^* = F_i(x^*)$
Unimodal Function	1	Shifted and Rotated Bent Cigar Function (CEC 2017 F1)	100
Basic Functions	2	Shifted and Rotated Schwefel's Function (CEC 2014 F11)	1100
	3	Shifted and Rotated Lunacek bi-Rastrigin Function (CEC 2017 F7)	700
	4	Expanded Rosenbrock's plus Griewangk's Function (CEC2017 f ₁₉)	1900
Hybrid Functions	5	Hybrid Function 1 (N = 3) (CEC 2014 F17)	1700
	6	Hybrid Function 2 (N = 4) (CEC 2017 F16)	1600
	7	Hybrid Function 3 (N = 5) (CEC 2014 F21)	2100
Composition Functions	8	Composition Function 1 (N = 3) (CEC 2017 F22)	2200
	9	Composition Function 2 (N = 4) (CEC 2017 F24)	2400
	10	Composition Function 3 (N = 5) (CEC 2017 F25)	2500

Search range: $[-100,100]^D$

2.1、LEO：实验

Comparisons with PSO, DE, and NL-SHADE-RSP (champion algorithm)

LEO能提高常用算法和冠军算法性能

Problem	L-PSO	PSO	L-DE	DE	L-NL-SHADE-RSP	NL-SHADE-RSP
F01(10D)	0±0	2.38E-05±6.01E-05 (+)(l)	0±0	0±0 (≈)(s)	0±0	0±0 (≈)(s)
F02(10D)	4.10E+00±2.16E+01	8.71E+02±2.70E+02 (+)(l)	3.14E-01±8.09E-02	3.45E+02±1.85E+02 (+)(l)	0±0	0±0 (≈)(s)
F03(10D)	8.41E+00±4.72E+00	7.40E+01±9.91E+00 (+)(l)	1.10E+01±2.63E-01	2.32E+01±4.04E+00 (+)(l)	0±0	0±0 (≈)(s)
F04(10D)	4.06E-01±5.93E-02	6.52E+00±1.04E+00 (+)(l)	4.20E-01±6.02E-02	1.50E+00±5.02E-01 (+)(l)	3.62E-03±7.10E-03	1.43E-02 ±2.14E-02 (+)(l)
F05(10D)	5.19E+01±5.91E+01	8.67E+01±8.02E+01 (+)(m)	1.53E-01±1.54E-01	1.32E-01±1.77E-01 (-)(s)	0±0	0±0 (≈)(s)
F06(10D)	3.68E-01±2.17E-01	2.24E+01±3.39E+01 (+)(l)	3.66E-01±2.43E-01	1.61E-01 ±1.89E-01 (-)(l)	7.17E-02±1.80E-01	7.17E-02±1.80E-01(≈)(s)
F07(10D)	3.62E+01±5.49E+01	1.64E+01±3.60E+01 (-)(l)	1.47E-01±1.92E-01	1.30E-01±1.76E-01 (-)(s)	6.33E-04±1.04E-03	1.38E-03±1.66E-03 (+)(l)
F08(10D)	0±0	3.37E+02±7.96E+01 (+)(l)	0±0	0±0 (≈)(s)	0±0	0±0 (≈)(s)
F09(10D)	0±0	5.43E-04±1.18E-03 (+)(l)	0±0	0±0 (≈)(s)	0±0	0±0 (≈)(s)
F10(10D)	4.81E+01±1.20E-01	5.25E+01±5.84E-01 (+)(l)	4.81E+01±1.20E-01	4.85E+01±1.59E-01 (≈)(s)	1.88E-03±1.00E-03	1.93E-03 ±9.41E-04(+)(s)
F01(20D)	0±0	1.97E+08±5.10E+07 (+)(l)	0±0	0±0 (≈)(s)	0±0	0±0 (≈)(s)
F02(20D)	4.22E+00±2.16E+01	2.74E+03±2.92E+02 (+)(l)	3.71E-01±6.50E-02	6.43E+00±2.77E+00 (+)(l)	0±0	0±0 (≈)(s)
F03(20D)	1.60E+01±8.78E+00	1.87E+02±1.79E+01 (+)(l)	2.02E+01±0.00E+00	2.34E+01±4.32E+00 (+)(l)	0±0	0±0 (≈)(s)
F04(20D)	8.14E-01±1.13E-01	1.73E+01±1.80E+00 (+)(l)	7.65E-01±8.46E-02	2.75E+00±2.32E+00 (+)(l)	1.62E-02±1.70E-02	3.16E-02±1.52E-02 (+)(l)
F05(20D)	1.18E+02±1.04E+02	1.19E+03±3.36E+02 (+)(l)	8.46E-01±9.04E-01	1.97E+00±2.58E+00 (+)(l)	0±0	0±0 (≈)(s)
F06(20D)	2.10E+00±1.84E+00	3.13E+02±1.20E+02 (+)(l)	8.47E-01±5.75E-01	2.11E+00±2.38E+00 (+)(l)	2.13E-02±2.48E-02	3.90E-02±1.82E-02 (+)(l)
F07(20D)	5.51E+01±8.32E+01	2.95E+02±1.58E+02 (+)(l)	8.01E-01±1.45E-01	1.38E+00±2.13E+00 (+)(l)	9.71E-03±1.18E-02	1.52E-02±1.05E-02 (+)(s)
F08(20D)	1.72E+00±3.91E+00	1.08E+03±1.49E+02 (+)(l)	0±0	3.29E+00±5.78E+00 (+)(l)	0±0	0±0 (≈)(s)
F09(20D)	0±0	5.55E+01±1.28E+01 (+)(l)	0±0	0±0 (≈)(s)	0±0	0±0 (≈)(s)
F10(20D)	5.01E+01±4.73E+00	5.86E+01±1.64E+00 (+)(l)	4.88E+01±6.53E-02	4.88E+01±2.70E-02 (≈)(s)	4.12E-03±1.72E-03	4.29E-03±1.54E-03 (+)(s)
+/-	NA	19/0/1	NA	10/7/3	NA	7/13/0

2.1、LEO：实验

Comparisons on 100-D problems with PSO, DE, and NL-SHADE-RSP (champion algorithm)

LEO在高维问题上的效果更加显著

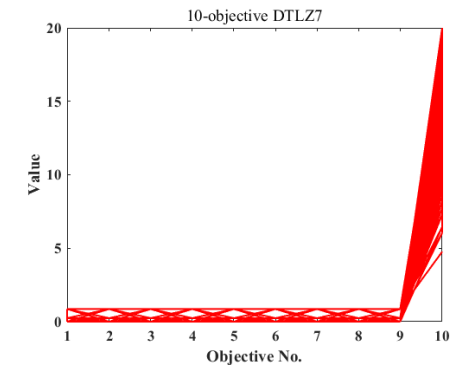
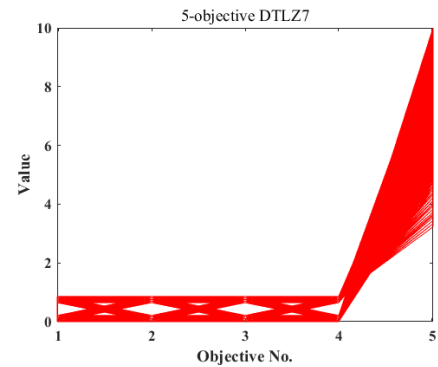
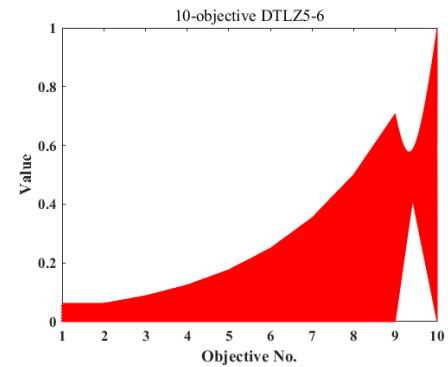
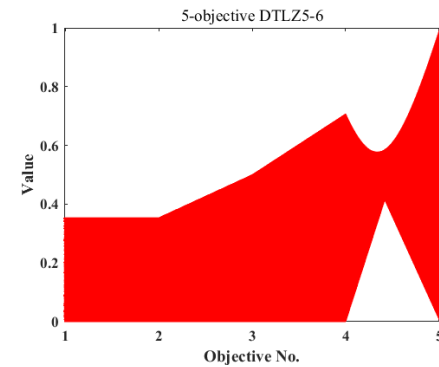
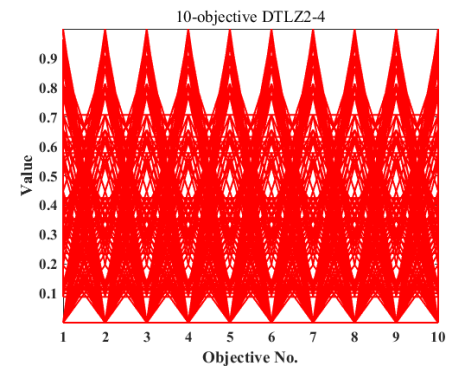
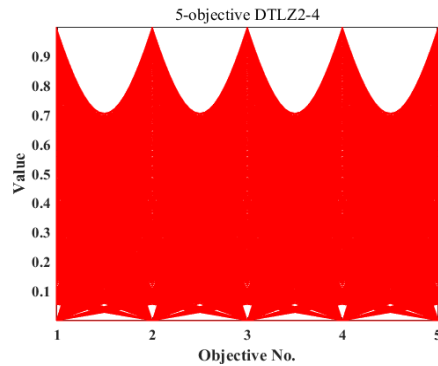
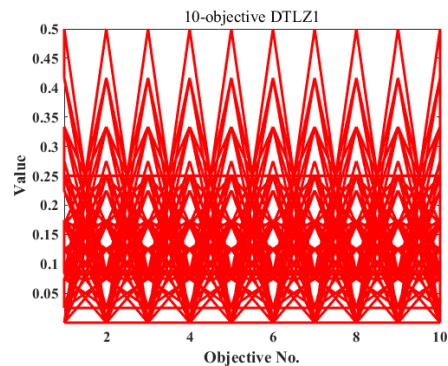
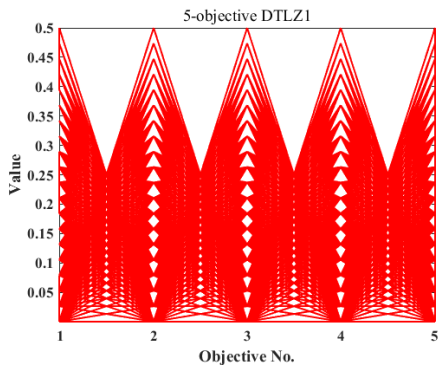
Problem	L-PSO	PSO	L-DE	DE	L-NL-SHADE-RSP	NL-SHADE-RSP
F01	0±0	7.33E+09±7.75E+08(+)(l)	0±0	2.54E-07±1.65E-07(+)(l)	0±0	0±0(≈)(s)
F02	1.17E+03±3.65E+02	2.74E+04±1.03E+03(+)(l)	3.51E+00±4.70E-01	2.55E+04±1.75E+03(+)(l)	2.85E-06±1.69E-07	3.15E-07±3.82E-08(≈)(l)
F03	2.00E+02±4.52E+01	1.32E+03±4.45E+01(+)(l)	1.66E+02±2.20E+01	8.49E+02±4.64E+01(+)(l)	1.78E-07±4.49E-08	2.37E-07±9.57E-08(≈)(l)
F04	6.08E+01±1.32E+01	1.98E+02±2.71E+01(+)(l)	2.51E+01±9.67E-01	7.50E+01±3.05E+00(+)(l)	4.34E-01±5.57E-01	2.14E+00±4.77E+00(+)(l)
F05	2.16E+03±5.63E+02	1.28E+08±3.60E+07(+)(l)	2.64E+02±1.32E+02	2.67E+03±1.24E+03(≈)(l)	4.17E+01±9.53E+00	1.48E+02±4.49E+02(+)(m)
F06	2.56E+02±1.77E+02	6.99E+03±5.03E+02(+)(l)	1.79E+01±4.80E+00	4.00E+03±1.05E+03(+)(l)	4.98E+00±1.47E+00	2.24E+01±2.67E+00(+)(l)
F07	1.34E+03±2.98E+02	1.09E+08±3.40E+07(+)(l)	1.36E+02±1.80E+02	6.71E+02±6.71E+02(+)(l)	5.12E-02±8.99E-03	9.14E+01±1.57E+01(+)(l)
F08	1.39E+03±8.20E+01	1.09E+04±5.53E+02(+)(l)	1.25E+02±2.57E+01	2.66E+02±7.12E+01(+)(l)	1.39E+00±2.20E+00	3.23E+01±6.95E+02(+)(s)
F09	1.52E+02±3.46E+01	1.05E+03±4.15E+01(+)(l)	1.88E+00±4.60E+00	7.27E+00±1.40E+01(+)(l)	1.52E-02±3.46E-01	1.93E-02±3.51E-04(≈)(s)
F10	6.66E+02±4.67E+01	1.05E+03±8.74E+01(+)(l)	5.24E+02±2.66E+01	5.44E+02±3.51E+01(≈)(l)	2.67E-02±3.67E-01	3.86E-02±1.41E-01(+)(s)
+/≈/-	NA	10/0/0	NA	8/2/0	NA	6/4/0
Adjusted <i>p</i> -value	NA	0.0016	NA	0.0016	NA	0.0269

2.1、LEO：实验



DTLZ benchmark for multi-/many-objective optimization

- Objective number: 5, 10, 15, 20



2.1、LEO：实验

Comparisons with NSGA-II, MOEA/D, and RVEA

LEO能提高多目标算法性能

Prob.	# obj.	L-NSGA-II	NSGA-II	L-MOEA/D	MOEA/D	L-RVEA	RVEA
DTLZ1	5	2.33E+00±1.46E+00	1.28E+00±7.33E-01(-)(l)	2.34E-01±3.63E-01	2.88E-01±3.67E-01(+)(s)	4.63E-01±2.75E-01	4.69E-01±3.06E-01(≈)(s)
	10	2.12E+01±6.97E+00	2.30E+01±9.76E+00(+)(m)	3.39E-01±3.18E-01	1.03E-01±2.23E-03(-)(l)	4.96E-01±4.33E-01	1.95E-01±9.76E-02(-)(l)
	15	2.30E+01±1.06E+01	3.27E+01±1.47E+01(+)(l)	3.19E-01±1.82E-01	3.92E-01±3.14E-01(+)(m)	6.10E-01±4.77E-01	7.15E-01±8.64E-01(≈)(s)
	20	2.66E+01±1.31E+01	3.39E+01±1.43E+01(+)(l)	3.72E-01±1.90E-01	4.79E-01±3.66E-01(+)(m)	8.20E-01±6.15E-01	9.75E-01±8.96E-01(+)(m)
DTLZ2	5	2.54E-01±1.30E-02	2.83E-01±1.37E-02(+)(l)	2.13E-01±5.04E-04	2.13E-01±5.35E-04(≈)(s)	2.12E-01±8.15E-04	2.14E-01±5.52E-04(≈)(l)
	10	1.44E+00±1.85E-01	1.37E+00±1.36E-01(-)(m)	5.36E-01±5.05E-02	4.13E-01±2.89E-03(-)(l)	5.33E-01±5.97E-02	4.25E-01±1.59E-03(-)(l)
	15	1.51E+00±1.57E-01	1.40E+00±9.42E-02(-)(l)	1.10E+00±8.98E-02	1.11E+00±9.05E-02(≈)(s)	9.22E-01±1.31E-01	9.56E-01±1.08E-01(+)(m)
	20	1.54E+00±1.24E-01	1.51E+00±9.12E-02(≈)(s)	8.93E-01±2.02E-02	9.00E-01±7.49E-02(≈)(s)	1.00E+00±1.00E-01	1.01E+00±1.04E-01(≈)(s)
DTLZ3	5	6.13E+01±1.10E+01	3.67E+01±1.57E+01(-)(l)	1.17E+01±8.53E+00	1.93E+01±1.14E+0(+)(l)	1.24E+01±5.93E+00	1.44E+01±6.82E+00(+)(m)
	10	7.84E+02±1.40E+02	7.85E+02±1.76E+02(≈)(s)	9.42E+00±5.75E+00	1.40E+00±1.35E+00(-)(l)	1.29E+01±5.38E+00	9.39E+00±4.08E+00(-)(l)
	15	6.77E+02±1.75E+02	7.66E+02±1.78E+02(+)(l)	1.10E+01±8.23E+00	1.00E+01±6.62E+00(≈)(s)	2.21E+01±1.27E+01	2.10E+01±1.30E+01(-)(s)
	20	6.65E+02±2.10E+02	7.35E+02±1.36E+02(+)(l)	1.43E+01±7.74E+00	1.28E+01±6.90E+00(-)(s)	2.44E+01±1.37E+01	2.63E+01±9.97E+00(+)(s)
DTLZ4	5	2.61E-01±2.52E-02	2.80E-0±1.26E-02(+)(l)	6.42E-01±2.29E-01	6.55E-01±2.04E-01(+)(s)	2.22E-01±3.89E-02	2.72E-01±1.12E-01(+)(l)
	10	9.56E-01±8.38E-02	1.40E+00±1.10E-01(+)(l)	9.20E-01±1.33E-01	7.33E-01±1.09E-01(-)(l)	5.92E-01±5.81E-02	4.43E-01±2.67E-03(-)(l)
	15	1.13E+00±7.83E-02	1.44E+00±1.11E-01(+)(l)	1.10E+00±1.38E-01	1.11E+00±1.32E-01(≈)(s)	8.82E-01±8.40E-02	8.95E-01±7.39E-02(≈)(s)
	20	1.22E+00±7.30E-02	1.46E+00±8.17E-02(+)(l)	1.16E+00±1.27E-01	1.19E+00±1.21E-01(≈)(m)	8.87E-01±5.84E-02	9.01E-01±5.89E-02(≈)(m)
DTLZ5	5	1.62E-01±4.20E-02	1.39E-01±3.34E-02(-)(l)	2.16E-02±1.60E-03	2.22E-02±1.73E-03(≈)(m)	2.12E-01±6.63E-02	2.16E-01±7.93E-02(≈)(s)
	10	3.36E-01±9.32E-02	2.58E-01±7.55E-02(-)(l)	7.71E-02±8.47E-03	1.87E-02±3.19E-04(-)(l)	4.05E-01±1.63E-01	3.85E-01±7.49E-02(≈)(s)
	15	4.74E-01±1.85E-01	6.26E-01±1.61E-01(+)(l)	3.73E-01±1.96E-01	3.74E-01±1.89E-01(≈)(s)	6.59E-01±1.25E-01	6.65E-01±1.31E-01(≈)(s)
	20	5.36E-01±2.23E-01	6.46E-01±2.26E-01(+)(m)	2.65E-01±1.10E-01	2.37E-01±5.54E-03(-)(m)	6.40E-01±1.44E-01	6.35E-01±1.44E-01(≈)(s)
DTLZ6	5	4.53E+00±1.16E+00	4.88E+00±7.50E-01(+)(m)	2.12E-01±3.77E-01	2.42E-01±4.53E-01(+)(s)	3.59E-01±2.25E-01	2.95E-01±2.07E-01(≈)(m)
	10	7.50E+00±8.91E-01	6.03E+00±7.91E-01(-)(l)	1.95E-01±3.81E-01	2.45E-01±7.14E-01(+)(s)	5.20E-01±4.00E-01	3.50E-01±2.21E-01(≈)(l)
	15	7.85E+00±6.35E-01	7.78E+00±7.45E-01(≈)(s)	4.91E-01±2.28E-01	4.59E-01±2.26E-01(-)(s)	6.77E-01±3.74E-01	6.64E-01±2.74E-01(≈)(s)
	20	8.00E+00±7.11E-01	7.88E+00±5.91E-01(≈)(s)	2.38E-01±6.53E-04	2.38E-01±1.56E-03(≈)(s)	9.37E-01±7.37E-01	1.07E+00±7.31E-01(+)(s)
DTLZ7	5	6.91E-01±8.87E-02	6.74E-01±5.62E-02(≈)(m)	1.07E+00±1.91E-01	1.04E+00±1.36E-01(≈)(s)	6.04E-01±6.74E-02	6.08E-01±6.75E-02(≈)(s)
	10	9.08E+00±2.81E+00	6.75E+00±2.12E+00(-)(l)	2.12E+00±3.71E-01	2.36E+00±5.08E-01(+)(l)	1.68E+00±1.49E-01	1.67E+00±2.58E-01(≈)(s)
	15	2.17E+01±3.82E+00	2.45E+01±4.27E+00(+)(l)	6.56E+00±8.88E-01	6.54E+00±1.12E+00(≈)(s)	2.53E+00±3.33E-01	2.61E+00±3.76E-01(≈)(m)
	20	3.43E+01±5.26E+00	3.45E+01±7.29E+00(≈)(m)	3.38E+00±6.65E-01	3.59E+00±5.59E-01(+)(m)	3.25E+00±8.65E-01	3.27E+00±8.40E-01(≈)(s)
+/-/		NA	14/6/8	NA	9/11/8	NA	6/17/5

2.1、LEO：实验

算法理论效率分析

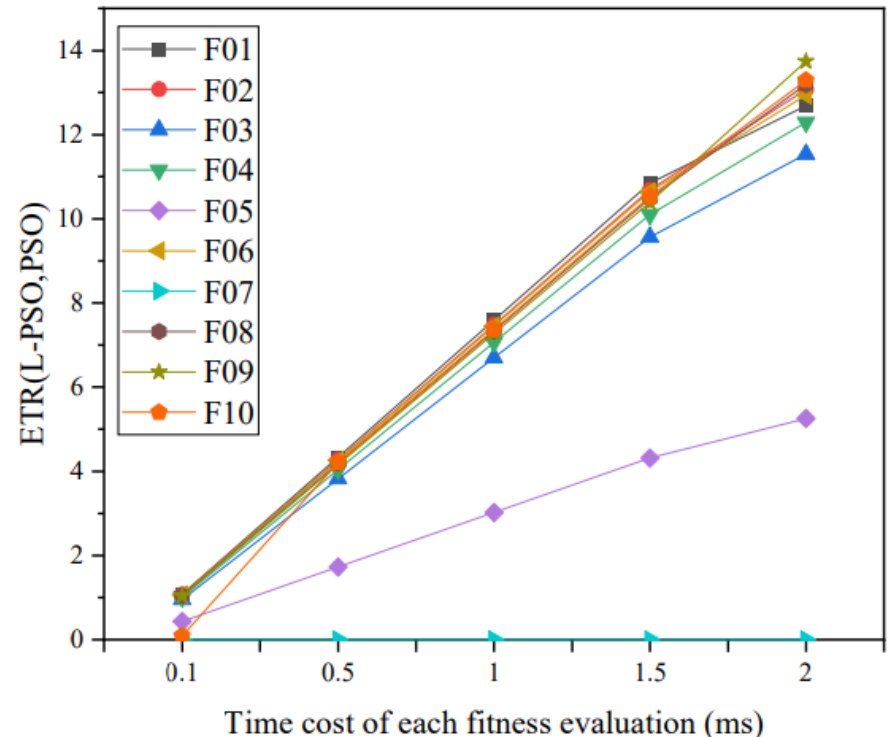
- To study the time efficiency of LEO, this paper proposes a novel indicator metric named **error reduction-time increase rate (i.e., ETR)**, which can measure the benefit of additional time cost when using LEO.
- 提出一种度量“性能提升效率”指标
- 每多花一份时间带来的精度提升效果（越大越好）

LEO能用少量的额外时间开销带来大幅的精度提升

$$ETR(A, B) = \frac{ERR(A, B)}{TIR(A, B)}$$

$$TIR(A, B) = \frac{T_A - T_B}{T_B} \times 100\%$$

$$ERR(A, B) = \begin{cases} \frac{E_B - E_A}{E_B} \times 100\%, & \text{if } E_B > E_A \\ 0, & \text{otherwise} \end{cases}$$



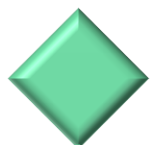
2.1、LEO：实验

学习辅助演化算子的参数调查

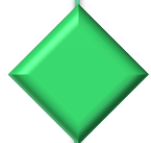
LEO对参数不敏感

Problem	L-PSO($\alpha=0.5$)	L-PSO($\alpha=0.1$)	L-PSO($\alpha=0.3$)	L-PSO($\alpha=0.7$)	L-PSO($\alpha=0.9$)
F01(10D)	0±0	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)
F02(10D)	4.10E+00±2.16E+01	4.22E+00±2.16E+01(≈)(s)	3.46E+00±9.47E+00 (≈)(s)	4.20E+00±1.33E+01(≈)(s)	4.10E+00±2.16E+01(≈)(s)
F03(10D)	8.41E+00±4.72E+00	8.87E+00±4.52E+00(≈)(s)	8.66E+00±4.71E+00(≈)(s)	8.35E+00±5.13E+00 (≈)(s)	8.41E+00±4.72E+00(≈)(s)
F04(10D)	4.06E-01±5.93E-02	6.00E-01±1.46E-01(+)(l)	4.49E-01±7.69E-02(≈)(l)	4.14E-01±5.74E-02(≈)(s)	4.06E-01±5.93E-02 (≈)(s)
F05(10D)	5.19E+01±5.91E+01	6.88E+01±7.77E+01(+)(m)	6.75E+01±8.55E+01(+)(m)	6.44E+01±5.24E+01(+)(m)	5.19E+01±5.91E+01 (≈)(s)
F06(10D)	3.68E-01±2.17E-01	4.10E-01±2.88E-01(≈)(s)	3.91E-01±2.02E-01(≈)(s)	3.55E-01±1.43E-01 (≈)(s)	3.68E-01±2.17E-01(≈)(s)
F07(10D)	3.62E+01±5.49E+01	3.24E+01±5.30E+01(≈)(s)	3.45E+01±5.45E+01(≈)(s)	3.20E+01±3.61E+01 (≈)(s)	3.62E+01±5.49E+01(≈)(s)
F08(10D)	0±0	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)
F09(10D)	0±0	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)
F10(10D)	4.81E+01±1.20E-01	4.82E+01±1.74E-01(≈)(m)	4.82E+01±2.70E-01(≈)(m)	4.81E+01±1.26E-01 (≈)(s)	4.81E+01±1.20E-01 (≈)(s)
Number of + / ≈ / -	NA	2 / 8 / 0	1 / 9 / 0	1 / 9 / 0	0 / 10 / 0

Problem	L-PSO($\beta=0.9$)	L-PSO($\beta=0.1$)	L-PSO($\beta=0.3$)	L-PSO($\beta=0.5$)	L-PSO($\beta=0.7$)
F01(10D)	0±0	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)
F02(10D)	4.10E+00±2.16E+01	3.46E+01±6.82E+01(+)(l)	4.37E+01±7.31E+01(+)(l)	8.03E+00±3.00E+01(≈)(s)	1.33E+01±7.82E-02(+)(l)
F03(10D)	8.41E+00±4.72E+00	8.10E+00±4.97E+00 (≈)(s)	1.10E+01±3.77E+00(+)(l)	9.80E+00±4.29E+00(≈)(m)	9.48E+00±4.32E+00(≈)(m)
F04(10D)	4.06E-01±5.93E-02	4.91E-01±1.39E-01(≈)(m)	4.45E-01±1.99E-01(≈)(m)	4.70E-01±9.18E-02(≈)(m)	4.70E-01±9.18E-02(≈)(m)
F05(10D)	5.19E+01±5.91E+01	6.21E+01±7.53E+01(+)(s)	1.19E+02±1.30E+02(+)(l)	8.45E+01±8.62E+01(+)(m)	8.45E+01±8.62E+01(+)(m)
F06(10D)	3.68E-01±2.17E-01	9.76E-01±2.10E+00(+)(m)	4.58E-01±4.28E-01(+)(s)	4.13E-01±5.30E-01(≈)(s)	4.13E-01±5.30E-01(≈)(s)
F07(10D)	3.62E+01±5.49E+01	2.28E+01±4.41E+01 (-)(m)	4.36E+01±7.28E+01(+)(s)	3.23E+01±5.30E+01(≈)(s)	3.23E+01±5.30E+01(≈)(s)
F08(10D)	0±0	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)
F09(10D)	0±0	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)	0±0 (≈)(s)
F10(10D)	4.81E+01±1.20E-01	4.81E+01±1.41E-01 (≈)(s)	4.82E+01±4.20E-01(≈)(m)	4.82E+01±3.35E-01(≈)(m)	4.82E+01±3.35E-01(≈)(m)
Number of + / ≈ / -	NA	3 / 6 / 1	5 / 5 / 0	1 / 9 / 0	2 / 8 / 0



背景介绍：进化计算与机器学习



学习辅助的进化计算

- **LEO: Learning-aided Evolution Optimization**
 - 群体演化宏知识纵向学习辅助的进化优化
- **KLEC: Knowledge Learning for Evolutionary Computation**
 - 个体方向微知识平向学习辅助的进化优化
- **MKTDE: Meta-Knowledge Transfer-based Differential Evolution**
 - 跨越问题元知识横向学习辅助的进化优化



总结



2.2、KLEEC：知识学习的进化计算

个体方向微知识平向学习辅助的进化优化

- Y. Jiang (Student), **Zhi-Hui Zhan(詹志辉)(通讯作者)**, *et al.*, “Knowledge learning for evolutionary computation” *IEEE Transactions on Evolutionary Computation*, DOI: 10.1109/TEVC.2023.3278132, May 2023.

2.2、KLEEC: 知识学习的进化计算

问题提出

如何利用个体进化过程的历史数据指导进化?

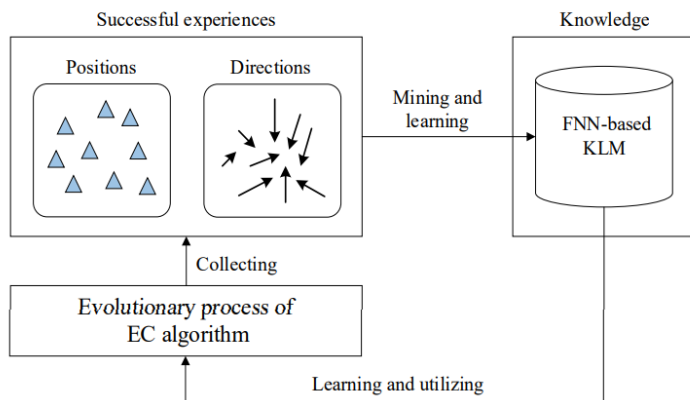
难点所在

- 如何从历史数据筛选合适的知识
- 如何利用学习得到的进化知识

主要研究内容

- 学习成功的历史进化方向
- 提出知识学习框架指导个体进化

个体进化方向微观知识学习的进化计算算法(Knowledge Learning for Evolutionary Computation, KLEEC)

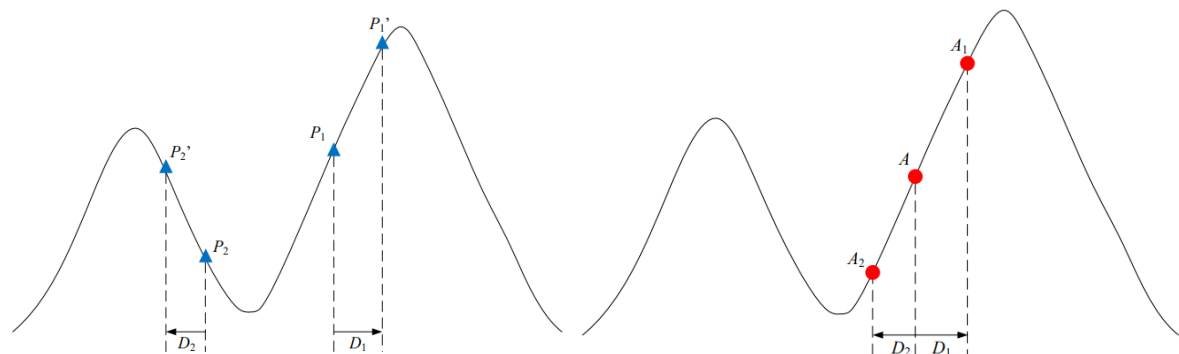


KLEEC

- 筛选和收集演化历史中的**成功进化方向**
- 采用神经网络学习位置与进化方向之间的对应关系，并**引导个体向更有效的方向演化**

2.2、KLEC：知识学习的进化计算

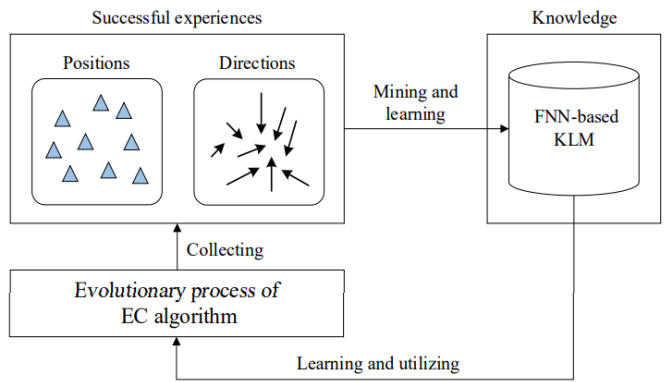
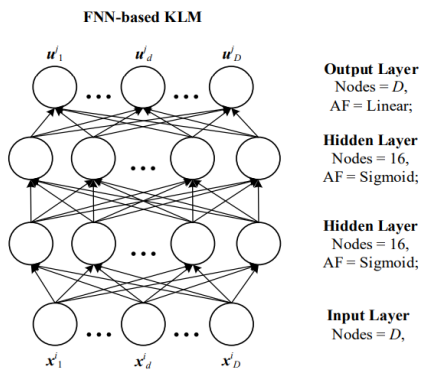
收集和学习演化历史中的成功进化方向



(X_i, D_i) 代表一条成功经验，其中 X_i 是个体位置， D_i 代表 X_i 的成功进化方向。

将收集的成功经验存放在一个档案中。这个档案中的历史经验会每代更新。

每代演化从，采用神经网络进行学习位置与进化方向之间的关系。

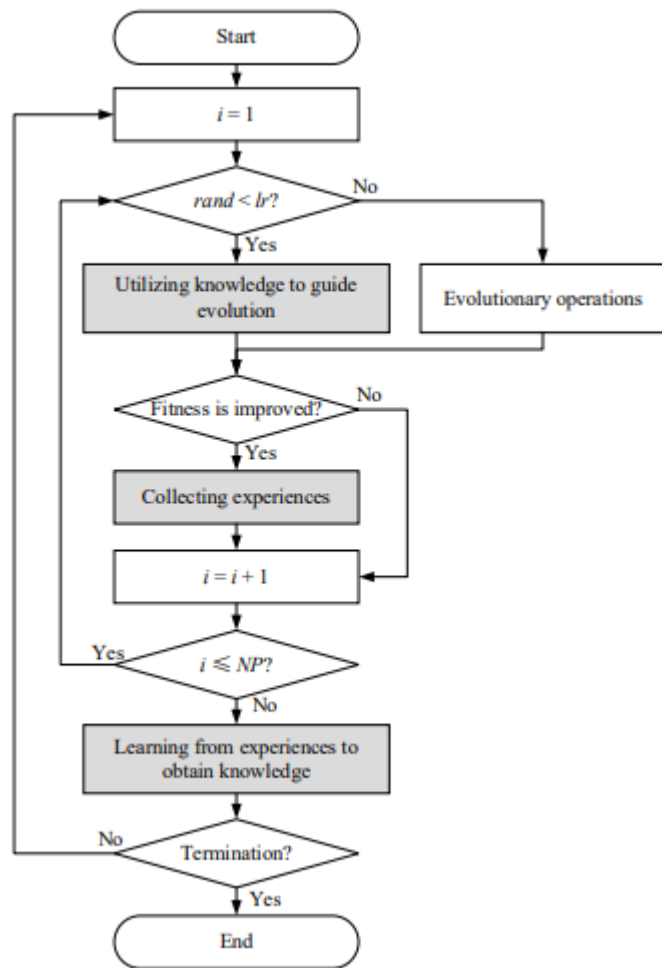


网络的损失函数
 $f(x_i)$ 为网络输出

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i))^2$$

2.2、KLECC: 知识学习的进化计算

知识学习的进化计算



进化操作

DE

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \text{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases}$$

PSO

$$v_i = w \cdot v_i + c_1 \cdot r_1 \cdot (pbest_i - x_i) + c_2 \cdot r_2 \cdot (gbest - x_i)$$

知识学习

$$\text{applyDirection}(x_{r1}, d_{r1}) = x_{r1} + 2 \times \text{rand}(0,1) \times d_{r1}$$

其中， d_{r1} 代表神经网络根据个体当前位置生成的最合适的进化方向。

2.2、KLEC：知识学习的进化计算

基于知识学习的差分进化算法和粒子群算法（KLDE和KLPSO）

Algorithm 1 KLDE

Begin

```

1: Initialize population  $P$ , experience list  $Q$ , and KLM.
2: While  $FES \leq MaxFES$ 
3:   For each individual  $x_i$ :
4:     If  $\text{rand}(0, 1) < lr$ :
5:        $d_{r1} = \text{utilizeKnowledge}(\text{KLM}, x_{r1})$ ; //Section III-C
6:        $v_i = \text{applyDirection}(x_{r1}, d_{r1})$ ; //Equation (9)
7:     Else:
8:        $v_i = \text{mutation}(x_{r1}, x_{r2}, x_{r3}, DE/\text{rand}/1)$ ; //Equation (2)
9:     End If
10:     $u_i = \text{crossover}(v_i, x_i)$ ; //Equation (3)
11:    If  $f(u_i) < f(x_i)$ :
12:      collecting( $Q, x_i, u_i - x_i$ );
13:       $x_i = u_i$ ;
14:    End If
15:  End For
16:  learningKnowledge(KLM,  $Q$ ); //Section III-B-2)
17:  clear( $Q$ );
18: End While

```

End

Algorithm 2 KLPSO

Begin

```

1: Initialize population  $P$ , experience list  $Q$ , and KLM.
2: While  $FES \leq MaxFES$ 
3:   For each particle  $x_i$ :
4:      $temp_i = x_i$ ;
5:     If  $\text{rand}(0, 1) < lr$ :
6:        $d_i = \text{utilizeKnowledge}(\text{KLM}, x_i)$ ; //Section III-C
7:        $x_i = \text{applyDirection}(x_i, d_i)$ ; //Equation (9)
8:     Else
9:       updateVelocity( $x_i, v_i, pbest_i, gbest$ ); //Equation (5)
10:       $x_i = x_i + v_i$ ; //Equation (6)
11:    End If
12:    If  $f(x_i) < f(temp_i)$ :
13:      collecting( $Q, temp_i, x_i - temp_i$ );
14:    End If
15:    If  $f(x_i) < f(pbest_i)$ :
16:       $pbest_i = x_i$ ;
17:      If  $f(pbest_i) < f(gbest)$ :
18:         $gbest = pbest_i$ ;
19:      End If
20:    End If
21:  End For
22:  learningKnowledge(KLM,  $Q$ ); //Section III-B-2)
23:  clear( $Q$ );
24: End While

```

End

2.2、KLEC: 实验

Comparisons with JADE, ADDE, HyDE-DF, and jSO
(champion algorithm)

KLEC能提高常用算法和冠军算法性能

Func	KL-JADE	JADE	KL-ADDE	ADDE	KL-jSO	jSO	KL-HyDE-DF	HyDE-DF
F_1	0.00E+00	0.00E+00 ≈	6.18E-02	3.72E-02 ≈	0.00E+00	0.00E+00 ≈	8.07E+00	4.98E+02 +
F_2	1.12E+05	1.62E+05 +	1.20E+05	1.19E+05 ≈	0.00E+00	0.00E+00 ≈	8.99E-02	2.99E+01 +
F_3	2.51E+01	3.79E+01 +	5.86E+01	5.87E+01 +	5.86E+01	5.91E+01 +	6.36E+01	5.65E+01 ≈
F_4	6.36E+01	1.83E+02 +	3.03E+01	3.21E+01 ≈	2.20E+01	1.02E+01 -	3.56E+01	4.43E+01 +
F_5	0.00E+00	0.00E+00 ≈	0.00E+00	0.00E+00 ≈	4.03E-08	1.48E-08 -	1.91E-02	5.94E-02 +
F_6	9.65E+01	2.18E+02 +	7.14E+01	6.96E+01 ≈	5.32E+01	3.91E+01 -	6.76E+01	8.45E+01 +
F_7	5.99E+01	1.75E+02 +	3.61E+01	3.48E+01 ≈	2.49E+01	1.12E+01 -	3.40E+01	4.14E+01 +
F_8	1.13E+01	7.19E-01 -	0.00E+00	0.00E+00 ≈	0.00E+00	0.00E+00 ≈	9.79E-02	3.41E+01 +
F_9	3.98E+03	8.07E+03 +	3.68E+03	4.03E+03 +	1.88E+03	1.75E+03	2.00E+03	2.42E+03 +
F_{10}	9.06E+01	8.31E+01 ≈	3.57E+01	3.10E+01	3.80E+00	8.04E+00 +	3.17E+01	4.50E+01 +
F_{11}	4.91E+03	3.17E+03 -	6.53E+03	4.90E+03 -	3.20E+02	1.44E+02 -	2.14E+04	6.12E+03 -
F_{12}	5.19E+02	4.30E+03 +	2.48E+02	2.25E+04 ≈	1.13E+01	2.14E+01 +	6.55E+03	3.80E+02 ≈
F_{13}	5.93E+04	2.69E+04 -	3.17E+03	3.32E+04 +	2.37E+01	2.30E+01 -	7.76E+01	1.03E+02 +
F_{14}	8.58E+03	3.99E+03 ≈	1.72E+02	4.74E+04 +	1.88E+00	3.25E+00 +	1.17E+02	1.24E+02 +
F_{15}	9.40E+02	1.87E+03 +	7.49E+02	7.47E+02 ≈	6.58E+01	1.25E+02 +	4.24E+02	3.80E+02 ≈
F_{16}	3.19E+02	4.31E+02 +	2.17E+02	2.53E+02 +	2.93E+01	3.88E+01 +	9.21E+01	1.02E+02 +
F_{17}	2.13E+05	7.27E+05 ≈	5.74E+05	6.75E+05 +	2.10E+01	2.13E+01 +	2.37E+02	3.50E+02 +
F_{18}	6.92E+03	6.14E+02 -	4.88E+01	3.44E+04 ≈	4.20E+00	7.37E+00 +	6.06E+01	5.77E+01 -
F_{19}	2.87E+02	5.19E+02 +	2.34E+02	2.98E+02 +	2.49E+01	3.53E+01 +	7.31E+01	1.33E+02 +
F_{20}	2.63E+02	4.01E+02 +	2.35E+02	2.36E+02 +	2.24E+02	2.11E+02	2.32E+02	2.41E+02 +
F_{21}	6.66E+02	1.36E+03 +	1.00E+02	1.00E+02 +	1.00E+02	1.00E+02 +	1.00E+02	1.00E+02 +
F_{22}	4.17E+02	5.33E+02 +	3.85E+02	3.86E+02 ≈	3.73E+02	3.48E+02 -	3.85E+02	3.96E+02 +
F_{23}	5.04E+02	6.34E+02 +	4.60E+02	4.60E+02 -	4.46E+02	4.24E+02 -	4.60E+02	4.67E+02 +
F_{24}	3.87E+02	3.87E+02 -	3.87E+02	3.87E+02 +	3.87E+02	3.87E+02 +	3.93E+02	3.86E+02 -
F_{25}	1.55E+03	2.49E+03 +	1.24E+03	1.21E+03 -	1.20E+03	9.13E+02 -	1.06E+03	1.27E+03 +
F_{26}	5.03E+02	5.07E+02 +	5.06E+02	5.02E+02 ≈	4.98E+02	4.94E+02 -	5.11E+02	5.08E+02 ≈
F_{27}	3.69E+02	3.47E+02 ≈	3.69E+02	3.25E+02 -	3.00E+02	3.09E+02 +	3.14E+02	3.45E+02 +
F_{28}	6.40E+02	1.27E+03 +	5.70E+02	5.83E+02 ≈	4.33E+02	4.46E+02 +	5.34E+02	5.28E+02 ≈
F_{29}	2.23E+03	2.20E+03 ≈	3.46E+03	7.91E+03 +	1.98E+03	1.97E+03 -	3.04E+03	2.94E+03 -
Number of +/≈/-		17 / 7 / 5	+ / ≈ / -	10 / 15 / 4	+ / ≈ / -	13 / 3 / 13	+ / ≈ / -	20 / 5 / 4

2.2、KLEC：实验

Comparisons with SaDPSO, HPSO-TAVC, TAPSO, and AWPSO

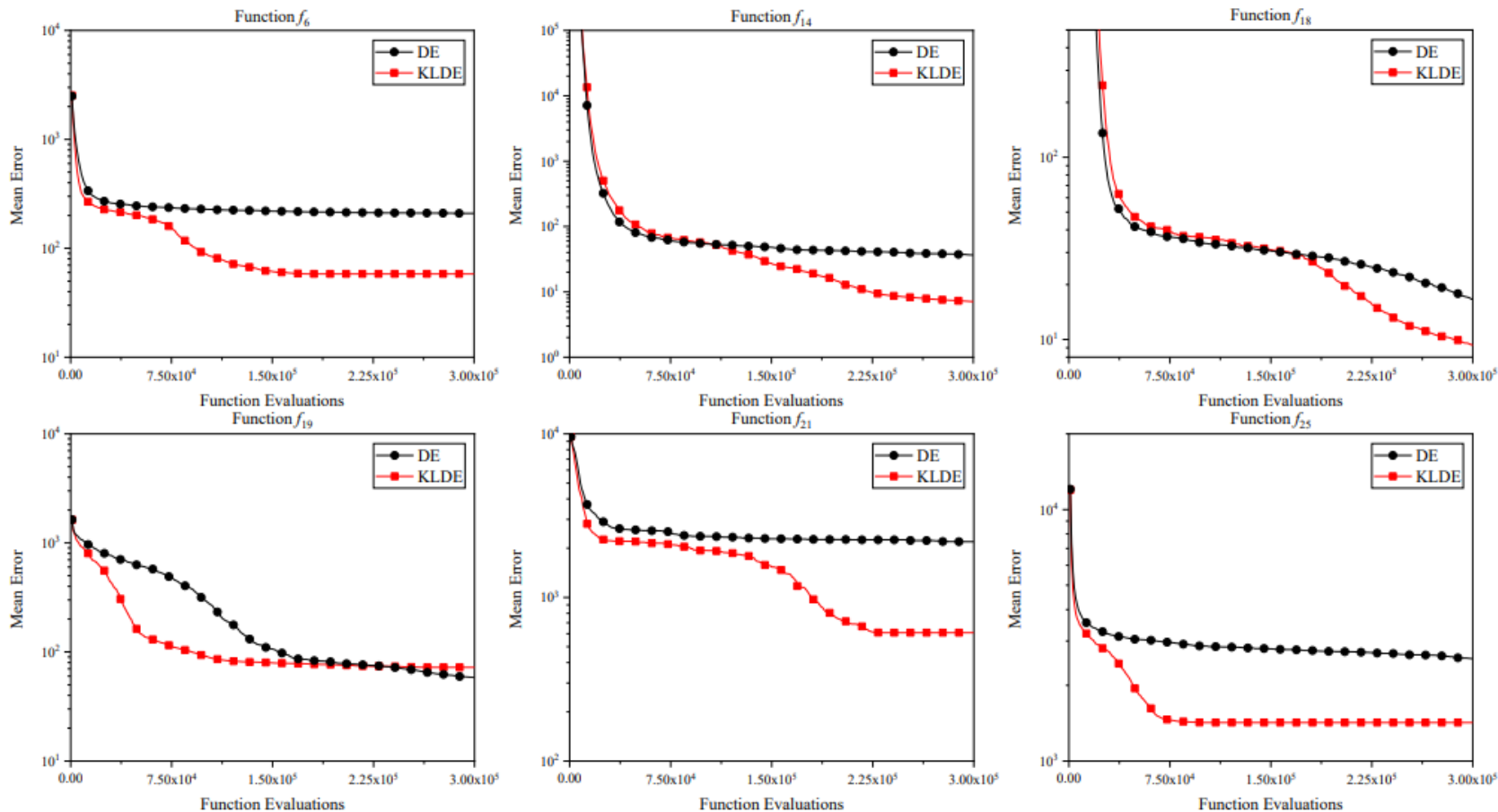
KLEC能提高常用算法和冠军算法性能

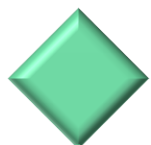
Func	KL-SaDPSO	SaDPSO	KL-HPSO-T VAC	HPSO-TVAC	KL-TAPSO	TAPSO	KL-AWPSO	AWPSO
F_1	8.82E+01	4.98E+02 ≈	1.15E+07	2.01E+07 +	2.19E+03	2.46E+03 ≈	4.85E+03	1.16E+08 ≈
F_2	1.32E-05	7.10E-06 ≈	1.91E+02	3.46E+02 +	9.89E+00	3.48E+01 ≈	2.71E-02	3.51E-04 -
F_3	1.56E+00	1.72E+00 ≈	9.94E+01	1.03E+02 ≈	4.91E+01	6.26E+01 +	8.58E+01	1.03E+02 +
F_4	4.82E+01	4.85E+01 ≈	1.26E+02	1.29E+02 ≈	4.62E+01	4.16E+01 ≈	4.60E+01	5.21E+01 +
F_5	9.58E-03	3.35E-02 +	1.23E+01	1.33E+01 ≈	1.96E-06	3.51E-07 -	1.59E-01	5.43E-01 +
F_6	7.93E+01	7.41E+01 ≈	1.96E+02	2.16E+02 +	6.64E+01	6.95E+01 ≈	6.62E+01	8.41E+01 +
F_7	5.27E+01	5.22E+01 ≈	1.00E+02	1.04E+02 ≈	4.30E+01	4.57E+01 ≈	4.10E+01	5.14E+01 +
F_8	3.17E+00	8.87E+00 +	1.20E+03	8.88E+02 ≈	1.10E-01	2.65E-01 +	3.02E+00	5.57E+00 +
F_9	2.54E+03	2.29E+03 -	3.23E+03	3.13E+03 ≈	2.37E+03	2.48E+03 ≈	2.48E+03	2.69E+03 ≈
F_{10}	7.55E+01	8.12E+01 ≈	1.24E+02	1.20E+02 ≈	5.26E+01	4.93E+01 ≈	9.56E+01	9.02E+01 ≈
F_{11}	1.84E+03	2.19E+03 ≈	3.27E+06	3.67E+06 ≈	1.65E+05	3.09E+04 -	1.24E+05	1.35E+06 +
F_{12}	1.46E+03	1.42E+03 ≈	5.01E+05	1.19E+06 +	1.31E+04	1.39E+04 ≈	2.04E+04	7.22E+05 +
F_{13}	4.75E+02	4.38E+02 ≈	9.44E+03	1.26E+04 ≈	2.67E+03	6.90E+03 +	1.00E+04	5.99E+03 ≈
F_{14}	3.23E+02	8.64E+02 ≈	1.03E+05	1.41E+05 ≈	2.11E+03	3.12E+03 ≈	9.23E+03	1.16E+04 ≈
F_{15}	5.52E+02	5.14E+02 ≈	9.53E+02	1.02E+03 ≈	7.24E+02	7.03E+02 ≈	5.57E+02	6.34E+02 ≈
F_{16}	1.35E+02	1.34E+02 ≈	3.04E+02	4.06E+02 +	1.46E+02	1.62E+02 ≈	2.16E+02	2.10E+02 ≈
F_{17}	2.69E+03	2.67E+03 ≈	2.17E+05	2.26E+05 ≈	1.08E+05	1.05E+05 ≈	2.37E+05	1.48E+05 -
F_{18}	3.25E+02	3.44E+02 +	1.28E+05	1.37E+05 ≈	4.01E+03	5.83E+03 +	7.59E+03	3.00E+04 ≈
F_{19}	1.96E+02	1.94E+02 ≈	3.32E+02	3.34E+02 ≈	1.97E+02	2.10E+02 ≈	2.56E+02	2.37E+02 ≈
F_{20}	2.50E+02	2.53E+02 ≈	3.29E+02	3.44E+02 +	2.41E+02	2.42E+02 ≈	2.51E+02	2.54E+02 ≈
F_{21}	1.43E+02	1.00E+02 -	1.33E+03	8.49E+02 ≈	1.00E+02	1.60E+02 +	1.19E+03	7.21E+02 ≈
F_{22}	4.05E+02	4.10E+02 ≈	6.65E+02	7.34E+02 +	3.99E+02	4.02E+02 ≈	4.49E+02	4.61E+02 +
F_{23}	4.76E+02	4.84E+02 ≈	8.54E+02	9.53E+02 +	4.76E+02	4.74E+02 ≈	5.37E+02	5.45E+02 ≈
F_{24}	3.83E+02	3.81E+02 ≈	3.95E+02	3.99E+02 ≈	3.89E+02	3.88E+02 ≈	3.88E+02	3.91E+02 +
F_{25}	6.06E+02	8.30E+02 ≈	2.53E+03	2.55E+03 ≈	1.35E+03	1.40E+03 ≈	1.13E+03	1.16E+03 ≈
F_{26}	5.07E+02	5.17E+02 +	6.85E+02	7.37E+02 ≈	5.10E+02	5.09E+02 ≈	5.33E+02	5.41E+02 ≈
F_{27}	3.32E+02	3.23E+02 ≈	4.12E+02	4.07E+02 ≈	3.43E+02	3.35E+02 -	4.24E+02	4.44E+02 +
F_{28}	5.90E+02	6.09E+02 ≈	9.03E+02	9.29E+02 ≈	5.15E+02	5.30E+02 ≈	6.48E+02	6.43E+02 ≈
F_{29}	1.35E+04	6.37E+03 -	1.45E+05	1.91E+05 ≈	3.83E+03	3.92E+03 ≈	9.05E+03	7.77E+03 ≈
Number of +/≈/-		4 / 22 / 3	+ / ≈ / -	8 / 21 / 0	+ / ≈ / -	5 / 22 / 3	+ / ≈ / -	11 / 16 / 2

2.2、KLEC：实验

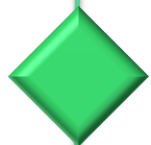


KLEC收敛速度明显优于现有算法



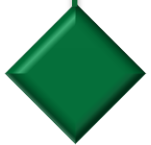


背景介绍：进化计算与机器学习



学习辅助的进化计算

- **LEO: Learning-aided Evolution Optimization**
 - 群体演化宏知识纵向学习辅助的进化优化
- **KLEC: Knowledge Learning for Evolutionary Computation**
 - 个体方向微知识平向学习辅助的进化优化
- **MKTDE: Meta-Knowledge Transfer-based Differential Evolution**
 - 跨越问题元知识横向学习辅助的进化优化



总结

2.2、MKTDE：元知识学习辅助进化优化



跨越问题元知识横向学习辅助的进化优化

- J.-Y. Li (Student), **Zhi-Hui Zhan (詹志辉)(Corresponding Author)**, *et al.*, “A meta-knowledge transfer-based differential evolution for multitask optimization,” ***IEEE Transactions on Evolutionary Computation***, vol. 26, no. 4, pp. 719-734,, Aug. 2022.

2.2、MKTDE：元知识迁移的差分进化

问题提出

需要对多个任务问题进行求解，**任务间差异大**

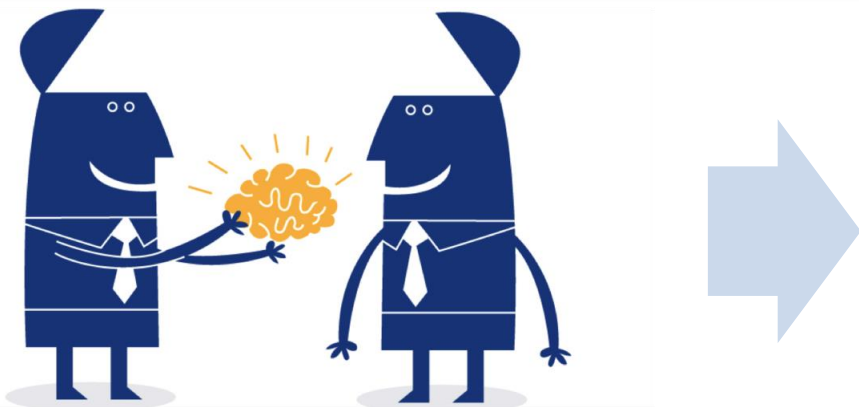
难点所在

- 如何利用知识学习，**提高多任务求解能力**
- 如何**避免负面迁移**？

主要研究内容

- 通过**跨问题元知识迁移**，提高算法多任务求解能力
- 在迁移时增加**控制机制**

基于元知识迁移的差分进化算法 (Meta-Knowledge Transfer based Differential Evolution, MKTDE)



MKTDE算法

- 学习跨问题通用元知识，并实现**元知识迁移**，提高算法多解能力
- 多种群对多任务协同求解，高效进行知识迁移，提高求解效率

2.2、MKTDE：研究动机

MKTDE的出发点

挑战一

现有常用的知识迁移依赖于任务之间的相关性，如何避免对任务相关性的依赖，提高迁移通用性？



元知识迁移方法

发现了异质任务在元层具有相通性的特点和规律，挖掘算法演化规律的通用元知识并进行迁移，避免对任务相关性的依赖

挑战二

相关性低的任务知识迁移容易导致负面的迁移效果，如何避免负面迁移，提高迁移效果？



精英解迁移方法

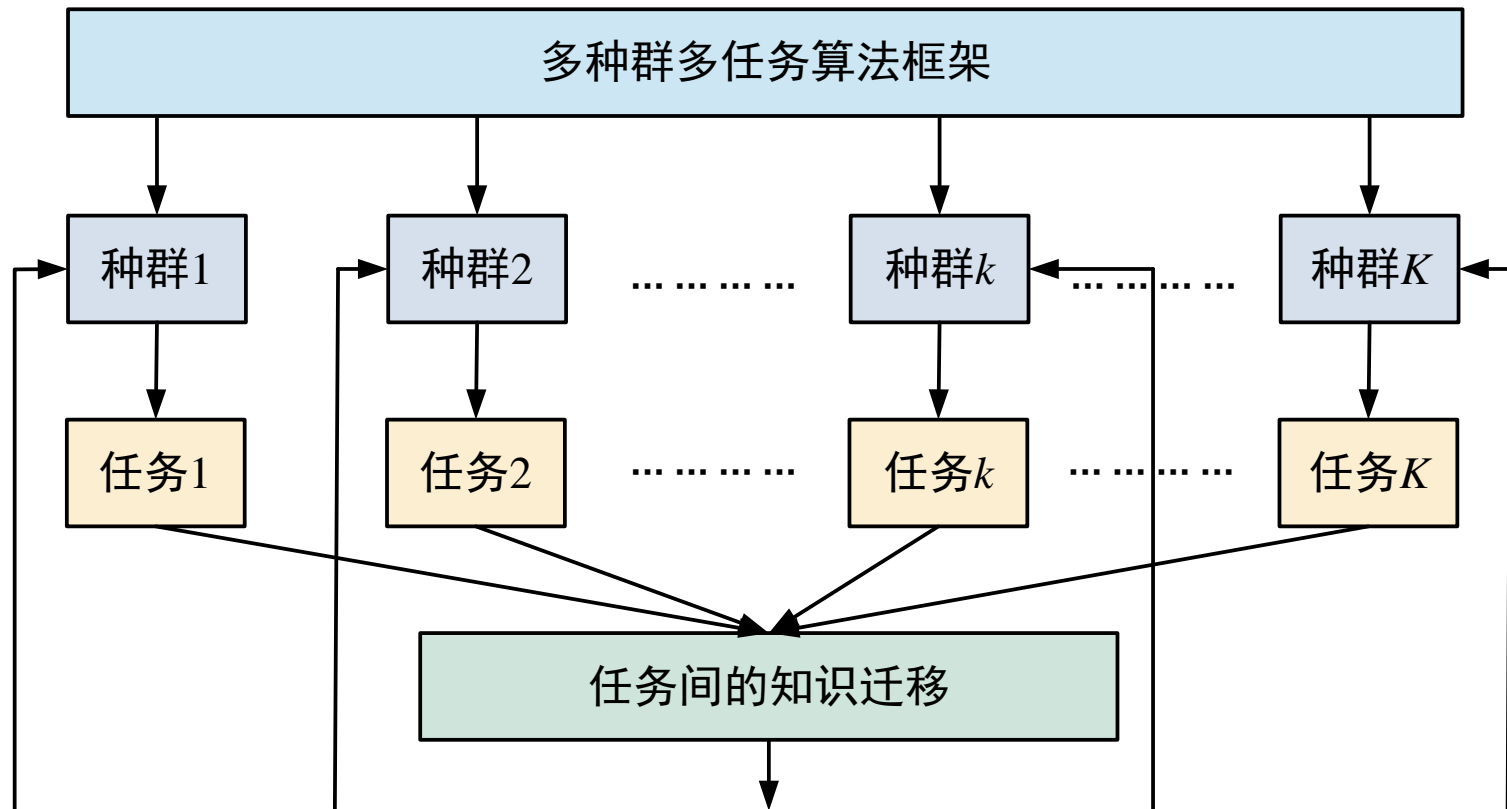
对优化任务精英解进行迁移的过程中增加迁移质量控制，保证知识迁移的正面效果

为了高效进行两种知识迁移，提出多种群多任务算法框架

2.2、MKTDE：算法框架

➤ 多种群多任务算法框架（Multiple population for multiple task, MPMT）

- 使用**多个种群**求解多个任务，通过**元知识迁移**进行协同
- 使用**统一虚拟搜索空间**，方便种群间进行**高效知识迁移**

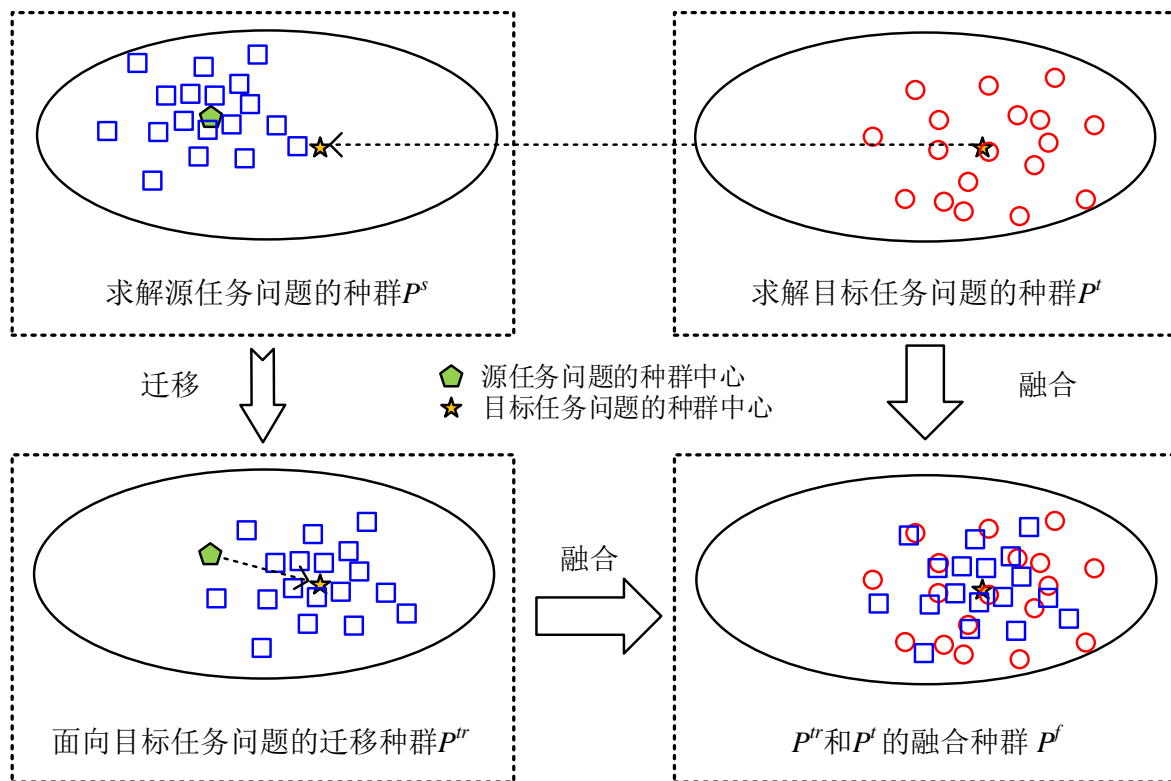


2.2、MKTDE: 元知识迁移

Meta-Knowledge Transfer (MKT)

差分进化算法基于**差分向量知识**对个体进行演化，产生高质量候选解

将种群进行迁移融合，得到其它种群的差分向量知识作为元知识，**促进演化**



元知识提取与迁移

分别计算源种群 P^s 和目标种群 P^t 的中心 c^s 和 c^t

将源种群 P^s 进行迁移，得到迁移种群 P^{tr}

$$P_i^{tr} = P_i^s - c^s + c^t$$

迁移种群 P^{tr} 与目标种群 P^t 合并，形成 P^f

$$P^f = P^{tr} \cup P^t$$

基于合并种群 P^f 产生的**差分向量知识**，对目标种群 P^t 的个体进行演化

变异 $B_i^f = P_{r1}^f + F \cdot (P_{fr1}^f - P_{fr2}^f)$

交叉
$$U_{i,j}^f = \begin{cases} B_{i,j}^f & \text{if } rand \leq CR \text{ or } j = j_{rand} \\ P_{i,j}^f & \text{otherwise} \end{cases}$$

2.2、MKTDE：精英解迁移

Elite Solution Transfer (EST)

算法：精英解迁移

1. 记目标任务种群 P^t 适应值最差的个体为 P_{wid}^t ;
2. 计算源任务种群中适应值最好的个体 X 在目标任务上的适应值，记为 fit ;
3. **If** P_{wid}^t 的适应值比 fit 差 **Then**
4. 用 X 替换 P_{wid}^t ;
5. **End If**

任务知识迁移

将源任务种群的精英解作为任务知识进行迁移，促进其它任务种群演化



迁移质量控制

源任务精英解在目标任务上的适应值若比待替换的解好，才进行迁移替换，否则停止迁移

在迁移后，保证目标任务种群的适应值不会变差，避免负面迁移

2.2、MKTDE：完整算法

完整的MKTDE

算法：完整的MKTDE算法

1. 在虚拟空间分别对任务1和任务2初始化 NP 个个体，形成种群 P^{T1} 和 P^{T2} ；
2. 评估初始化个体的适应值；
3. $FES = 2 \times NP$; //记录适应值评估消耗的次数
4. **While** $FES < MFES$ **Do** // $MFES$ 最大的适应值消耗次数
5. 进行任务间的元知识迁移；
6. 各任务的种群分别进行演化，产生新个体；
7. 评估新个体的适应值；
8. 记录适应值评估消耗的次数，并更新 FES ；
9. 通过选择操作更新种群及种群的最优解；
10. 进行任务间的精英解迁移；
11. 记录适应值评估消耗的次数，并更新 FES ；
12. **End While**
13. 将各任务搜索到的最优解转换回真实的搜索空间的解，并作为结果输出；

多种群协同演化

基于多种群多任务框架，采用统一的虚拟搜索空间，对多个任务进行演化



多知识迁移学习

每一代执行元知识迁移和精英解知识迁移，从而对多个任务进行协同求解

2.2、MKTDE：实验



实验测试函数与设置

➤ 最大适应值计算次数 1×10^5 次
(包括两个任务)

问题名	问题包含的任务	函数特点	任务间交叉程度	任务间相似程度
CI+HS	Griewank(T_1)	多峰、不可分	完全交叉 (Complete Intersection, CI)	1.0000
	Rastrigin(T_2)	多峰、不可分		
CI+MS	Ackley(T_1)	多峰、不可分	完全交叉	0.2261
	Rastrigin(T_2)	多峰、不可分		
CI+LS	Ackley(T_1)	多峰、不可分	完全交叉	0.0002
	Schwefel(T_2)	多峰、可分		
PI+HS	Rastrigin(T_1)	多峰、不可分	部分交叉(Partial Intersection, PI)	0.8670
	Sphere(T_2)	单峰、可分		
PI+MS	Ackley(T_1)	多峰、不可分	部分交叉	0.2154
	Rosenbrock(T_2)	多峰、不可分		
PI+LS	Ackley(T_1)	多峰、不可分	部分交叉	0.0725
	Weierstrass(T_2)	多峰、不可分		
NI+HS	Rosenbrock(T_1)	多峰、不可分	不交叉 (No Intersection, NI)	0.9434
	Rastrigin(T_2)	多峰、不可分		
NI+MS	Griewank(T_1)	多峰、不可分	没有交叉	0.3669
	Weierstrass(T_2)	多峰、不可分		
NI+LS	Rastrigin(T_1)	多峰、不可分	没有交叉	0.0016
	Schwefel(T_2)	多峰、可分		



2.2、MKTDE：实验

与前沿算法进行比较：

MKTDE显著优于对比算法

问题		MKTDE	MFEA-I	MFEA-II	MFEA-AKT	EMT-EGT	MFDE	MTDE	STDE
CI+HS	Score	-22.22	-4.72	-3.57	58.86	-21.74	-22.21	-22.21	37.80
CI+MS	Score	-23.13	-1.68	-1.84	49.37	-22.89	-23.12	-23.12	46.39
CI+LS	Score	-0.19	3.76	3.87	-48.51	13.06	13.78	3.55	10.67
PI+HS	Score	-11.50	-30.13	-30.24	78.25	-0.83	7.02	-24.04	11.46
PI+MS	Score	-10.96	-13.13	-13.77	60.81	-12.39	-13.41	-7.98	10.82
PI+LS	Score	-21.51	33.15	33.95	5.09	-22.12	-1.79	-22.94	-3.84
NI+HS	Score	-20.95	-25.19	-25.15	60.36	24.41	-22.36	-21.34	30.24
NI+MS	Score	-22.77	25.61	25.70	50.54	-16.34	-20.74	-19.81	-22.18
NI+LS	Score	-9.70	5.64	5.59	12.74	8.90	-5.04	-34.86	16.74
+/ \approx /-		NA	11/0/7	11/0/7	16/0/2	13/1/4	11/1/6	10/1/7	15/1/2
取得最优score的数量		3	1	2	1	0	0	2	0

2.2、MKTDE：实验

与前沿算法进行比较：

MKTDE显著优于对比算法

问题		MKTDE	MFEA-I	MFEA-II	MFEA-AKT	EMT-EGT	MFDE	MTDE	STDE
CI+HS	Score	-22.22	-4.72	-3.57	58.86	-21.74	-22.21	-22.21	37.80
CI+MS	Score	-23.13	-1.68	-1.84	49.37	-22.89	-23.12	-23.12	46.39
CI+LS	Score	-0.19	3.76	3.87	-48.51	13.06	13.78	3.55	10.67
PI+HS	Score	-11.50	-30.13	-30.24	78.25	-0.83	7.02	-24.04	11.46
PI+MS	Score	-10.96	-13.13	-13.77	60.81	-12.39	-13.41	-7.98	10.82
PI+LS	Score	-21.51	33.15	33.95	5.09	-22.12	-1.79	-22.94	-3.84
NI+HS	Score	-20.95	-25.19	-25.15	60.36	24.41	-22.36	-21.34	30.24
NI+MS	Score	-22.77	25.61	25.70	50.54	-16.34	-20.74	-19.81	-22.18
NI+LS	Score	-9.70	5.64	5.59	12.74	8.90	-5.04	-34.86	16.74
+/ \approx /-		NA	11/0/7	11/0/7	16/0/2	13/1/4	11/1/6	10/1/7	15/1/2
取得最优score的数量		3	1	2	1	0	0	2	0

2.2、MKTDE : 实验

组分分析:

- **MKTDE-w/o-MKT**: 不使用MKT
- **MKTDE-w/o-EST**: 不使用EST
- **MKTDE-w/o-MKT-EST**: 不使用MKT和EST

MKT和EST都能提高MKTDE的多解优化结果, 去掉其中任何一个都会使MKTDE的多任务求解能力下降

问题名	MKTDE	MKTDE-w/o-MKT	MKTDE-w/o-EST	MKTDE-w/o-MKT-EST
CI+HS	-27.16	-27.17	35.52	18.81
CI+MS	-20.32	-20.33	6.64	34.01
CI+LS	-22.18	22.18	-22.18	22.18
PI+HS	-7.33	8.20	-12.00	11.13
PI+MS	-7.10	-6.43	4.40	9.13
PI+LS	-9.75	1.75	-8.85	16.85
NI+HS	-21.72	-13.84	14.04	21.52
NI+MS	-18.83	7.68	-15.75	26.91
NI+LS	-27.57	27.49	-27.57	27.66
+/~/-	NA	10/2/6	13/4/1	15/1/2
取得最优score的数量	6	2	3	0

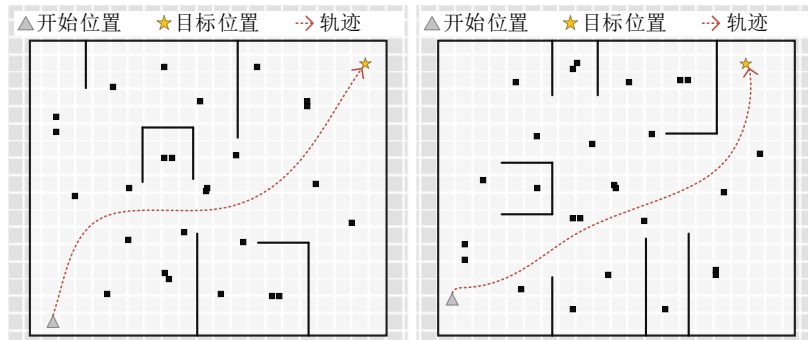
2.2、MKTDE：智能机器人应用



多智能机器人导航优化实例

$$f(traj) = c_l(traj) + c_{collision}(traj) + \lambda \cdot (dist(traj_{begin}, start) + dist(traj_{end}, goal)) + b$$

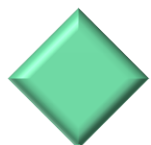
MKTDE显著优于对比算法



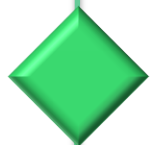
导航任务 1

导航任务 2

问题		MKTDE	MFEA-I	MFEA-II	MFEA-AKT	EMT-EGT	MFDE	MTDE	STDE
RN01	Score	-12.60	8.92	4.74	-10.38	-11.58	-10.10	-11.13	42.13
RN02	Score	-14.95	12.74	1.80	-7.05	-12.13	-8.50	-9.88	37.97
RN03	Score	-12.77	4.60	9.54	-9.05	-10.55	-8.38	-7.76	34.36
RN04	Score	-10.86	-1.51	-1.45	-6.01	-7.75	-3.49	-8.18	39.25
+/-		NA	8/0/0	8/0/0	7/1/0	8/0/0	8/0/0	8/0/0	8/0/0
取得最优score的数量		4	0	0	0	0	0	0	0

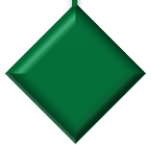


背景介绍：进化计算与机器学习



学习辅助的进化计算

- LEO: Learning-aided Evolution Optimization
 - 群体演化宏知识纵向学习辅助的进化优化
- KLEC: Knowledge Learning for Evolutionary Computation
 - 个体方向微知识平向学习辅助的进化优化
- MKTDE: Meta-Knowledge Transfer-based Differential Evolution
 - 跨越问题元知识横向学习辅助的进化优化



总结

3、总结

- 针对复杂优化问题，围绕学习辅助的进化计算方法，提出三种针对复杂优化问题的学习辅助进化计算方法，并解决智能机器人路径规划应用问题
 - LEO 通过群体演化宏知识学习来辅助进化优化 (**纵向学习**)
 - KLEC 通过个体方向微知识学习来辅助进化优化 (**平向学习**)
 - MKTDE 通过跨越问题元知识学习来辅助进化优化 (**横向学习**)
- 相关实验和对比验证了所提出算法具有优秀的性能
- 对学习辅助的进化优化算法进行了理论效率的分析
- **学习辅助的进化计算为求解大规模/超大规模、多任务等复杂优化问题提供了新思路**

参考资料

➤ 综述:

- Zhi-Hui Zhan (詹志辉), *et al.*, “A survey on evolutionary computation for complex continuous optimization,” *Artificial Intelligence Review*, vol. 55, no. 1, pp. 59-110, Jan. 2022.
- Zhi-Hui Zhan (詹志辉), *et al.*, “Evolutionary deep learning: A survey,” *Neurocomputing*, vol. 483, pp. 42-58, April 2022.
- J. Y. Li(学生), Zhi-Hui Zhan (詹志辉)(通讯作者), *et al.*, “Evolutionary computation for expensive optimization: A survey,” *Machine Intelligence Research*, vol. 19, no. 1, pp. 3-23, Jan. 2022.
- Z. G. Chen(学生), Zhi-Hui Zhan (詹志辉)(通讯作者), *et al.*, “Evolutionary computation for intelligent transportation in smart cities: A survey,” *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 83-102, May, 2022.

➤ 学习辅助进化计算:

- Zhi-Hui Zhan(詹志辉)(第一作者), *et al.*, “Learning-aided evolution for optimization,” *IEEE Transactions on Evolutionary Computation*, DOI:10.1109/TEVC.2022.3232776, Dec. 2022.
- Y. Jiang(学生), Zhi-Hui Zhan (詹志辉)(通讯作者), *et al.*, “Knowledge learning for evolutionary computation,” *IEEE Transactions on Evolutionary Computation*, DOI: 10.1109/TEVC.2023.3278132, May 2023.
- J. Y. Li(学生), Zhi-Hui Zhan (詹志辉)(通讯作者), *et al.*, “A meta-knowledge transfer-based differential evolution for multitask optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 719-734, Aug. 2022.

IEEE SSCI 2023 (IEEE计算智能国际会议)




SSCI 2023

HOME SYMPOSIA COMMITTEE AUTHOR PROGRAM ATTENDING REGISTRATION ABOUT

2023 IEEE Symposium Series on Computational Intelligence


December 5th – 8th 2023, Mexico City, Mexico



2023 IEEE Symposium Series on Computational Intelligence (SSCI 2023) is a flagship annual meeting of the IEEE Computational Intelligence Society. It provides a primary forum for multidisciplinary research in computational intelligence. SSCI 2023 will be held in Mexico City on December 5th-8th 2023. It is the first time the conference be held in Latin America, and in the largest Spanish-speaking city, Mexico City.

SSCI 2023 will be in-person conference. It will collocate more than 40 symposia, each symposium dedicates to a special topic within or related to computational intelligence. I hope that SSCI 2023 will be an exciting and memorable experience for all of you!

Wen Yu
CINVESTAV-IPN (National Polytechnic Institute)
Mexico City, Mexico
General Chair of SSCI 2023



投稿截止：8月15日

SSCI 2023

HOME SYMPOSIA COMMITTEE AUTHOR PROGRAM ATTENDING

Special session proposal: **March 15, 2023**
Paper submission: **June 30, 2023**
Paper acceptance: **August 31, 2023**
Conference dates: **December 5-8, 2023**

SSCI 2023

HOME SYMPOSIA COMMITTEE AUTHOR

Important Dates

- Proposals for special session: April 15, 2023
- Paper Submissions: **June 30, 2023** July 31, 2023
- Paper Acceptance: August 31, 2023
- Camera ready Paper: September 20, 2023
- Early Registration: September 20, 2023
- Conference dates: December 5-8, 2023

SSCI 2023

HOME SYMPOSIA COMMITTEE AUTHOR PROGRAM ATTENDING

Organizing Committee

General Chair

- Wen Yu, CINVESTAV-IPN, Mexico
yuw@ctrl.cinvestav.mx

Advisory Committee

- Carlos Coello Coello, CINVESTAV-IPN, Mexico
ccoello@cs.cinvestav.mx
- Jun Wang, City University of Hong Kong, Hong Kong
jwang.cs@cityu.edu.hk

Program Chairs

- Agung Julius, Rensselaer Polytechnic Institute, USA
agung@ecse.rpi.edu
- Dianhui Wang, La Trobe University, Australia, China University of Mining and Technology
dh.wang@latrobe.edu.au
- Zhi-Hui Zhan, South China University of Technology, China
cszhanzh@scut.edu.cn
- Xiaou Li, CINVESTAV-IPN, Mexico
lixo@cs.cinvestav.mx

Plenary Sessions Chair

- Yaochu Jin, Bielefeld University, Germany
yaochu.jin@uni-bielefeld.de

Finance Chair

- Bella Martinez, UPIITA-IPN, Mexico

In Mexico, December, 2023

<https://attend.ieee.org/ssci-2023/>



Thank you!

Q&A

詹志辉/Apollo

zhanapollo@163.com

