大规模多目标进化优化：算法与应用

安徽大学　田野

field910921@gmail.com

ECOLE 2022，郑州，2022年8月3日


目录
CONTENT

第 1 章 大规模多目标优化

问题定义

- 有多个优化目标、大量优化变量的问题，称为大规模多目标优化问题

目标向量

$$\min \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))$$

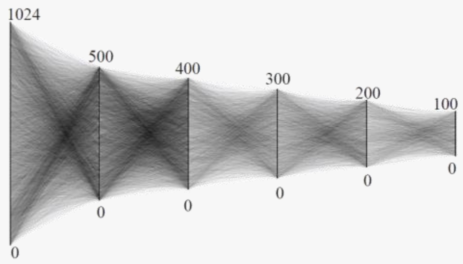
共 $M \geq 2$ 个目标函数

决策向量

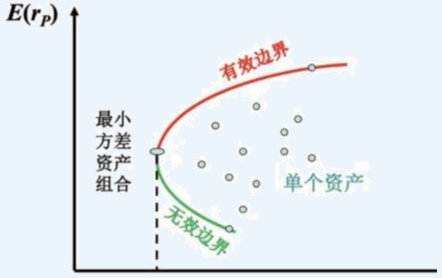
$$\text{s.t.} \ \mathbf{x} = (x_1, \dots, x_D)$$

共 $D \geq 100$ 个决策变量

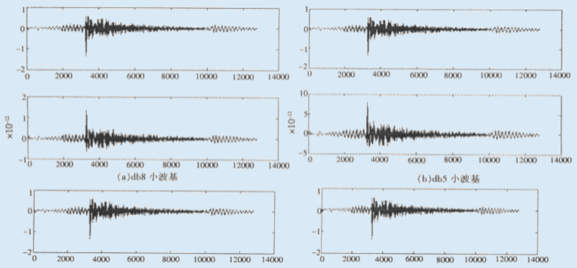
$$x_i \in [l_i, u_i]$$



神经网络训练
最小化误差
最小化复杂度



投资组合优化
最大化收益
最小化风险



稀疏信号重构
最小化误差
最大化稀疏度

问题定义

为什么要多目标

- 目标间具有矛盾，量级不同，难以加权

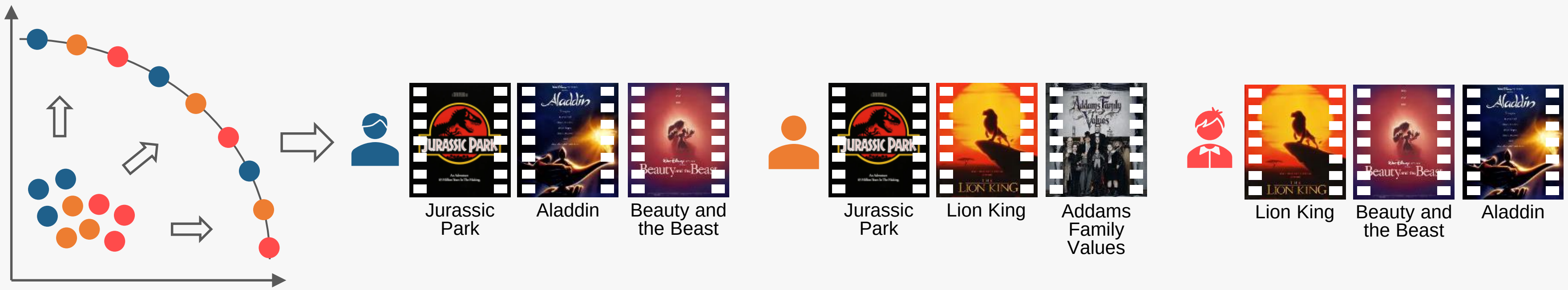
训练误差 vs 网络复杂度

预期收益 vs 预期风险

重构误差 vs 信号稀疏度

- 问题建模不精准，决策者需要多样化的解决方案供备选

个性化推荐系统

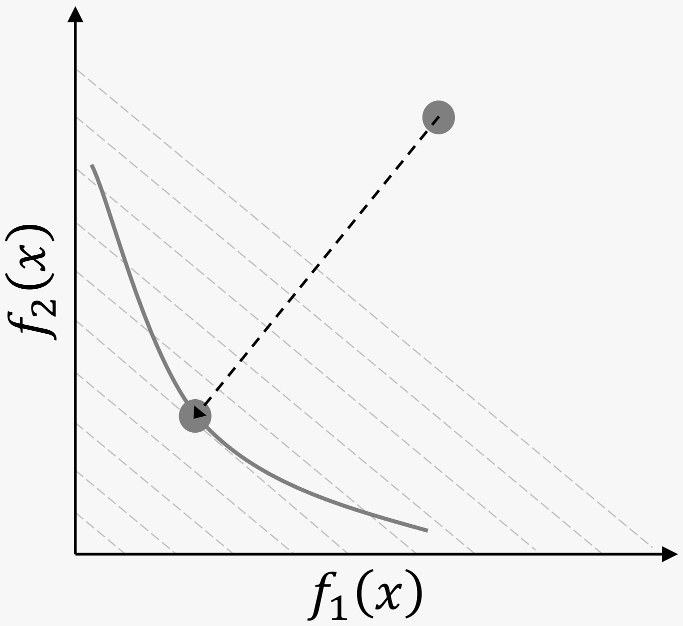


现有算法

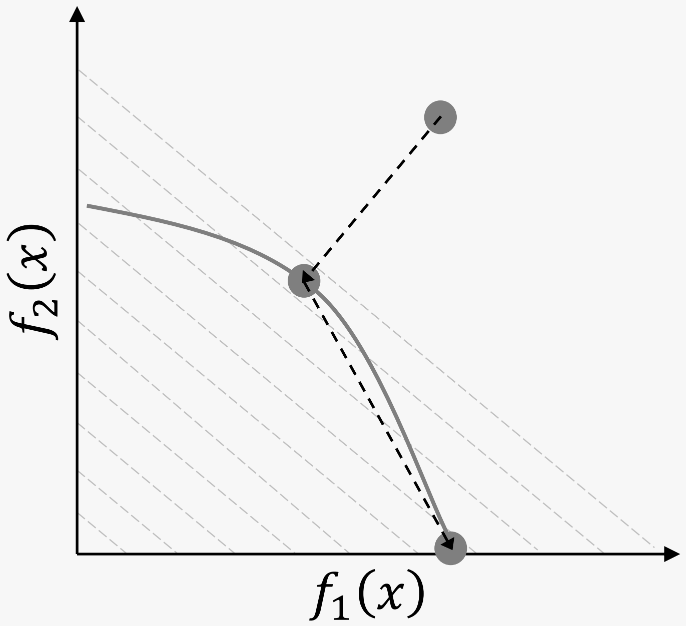
	优 势	大规模多目标上劣势
数学规划	大规模单目标收敛快	无法求解黑盒问题，多目标必须加权，只能得到一个解
贝叶斯优化	小规模问题评价少	难以对许多变量拟合模型，数据不足
强化学习	优化过程中学习到经验	无法求解黑盒问题，无法求解连续优化问题（无法定义状态）
进化算法	通用性强	大规模空间收敛速度太慢

现有算法

为什么不能加权



当最优前沿为凸时，最优解
位于权重向量与前沿的交点

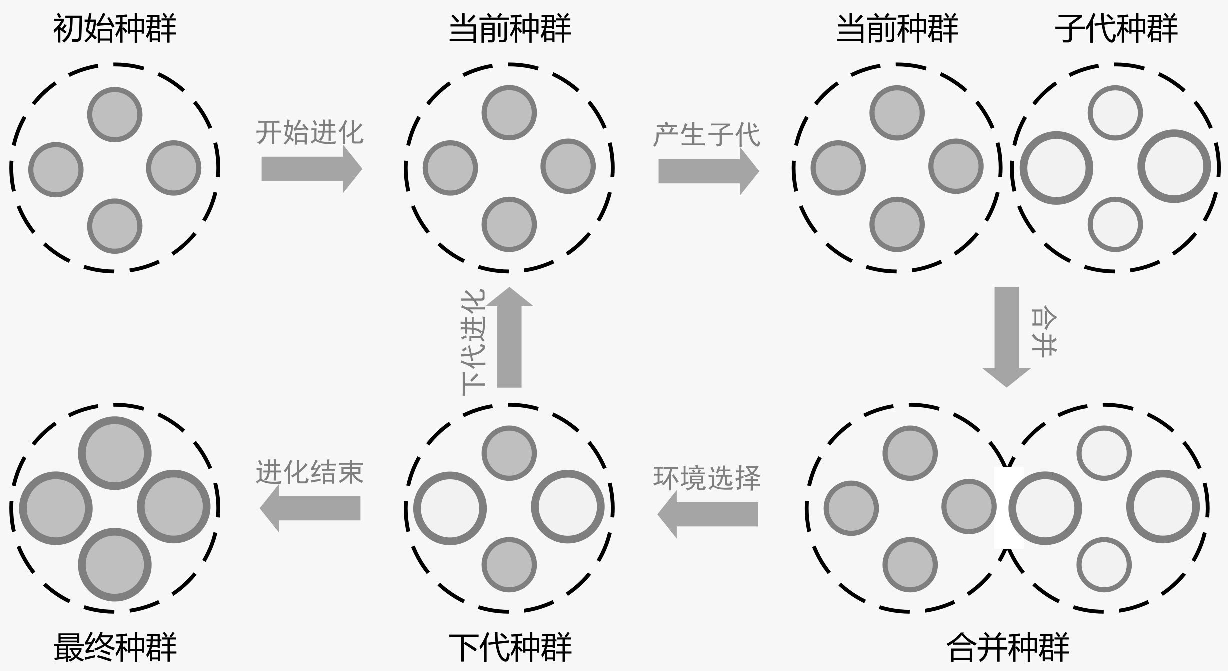


当最优前沿为凹时，
最优解始终位于边界

解决方案

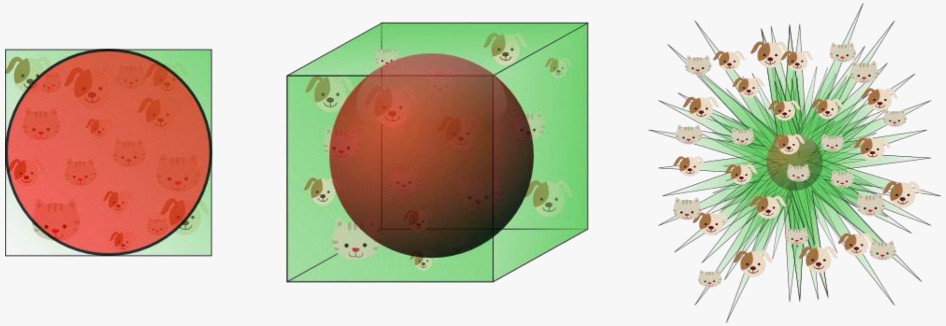
- 采用进化算法作为基本框架

✓ 无需目标加权　✓ 直接求解黑盒问题　✓ 得到一组多样化解　✓ 适用复杂前沿



解决方案

- 进化算法在大规模空间中会遭遇“维数灾难问题”，收敛过慢



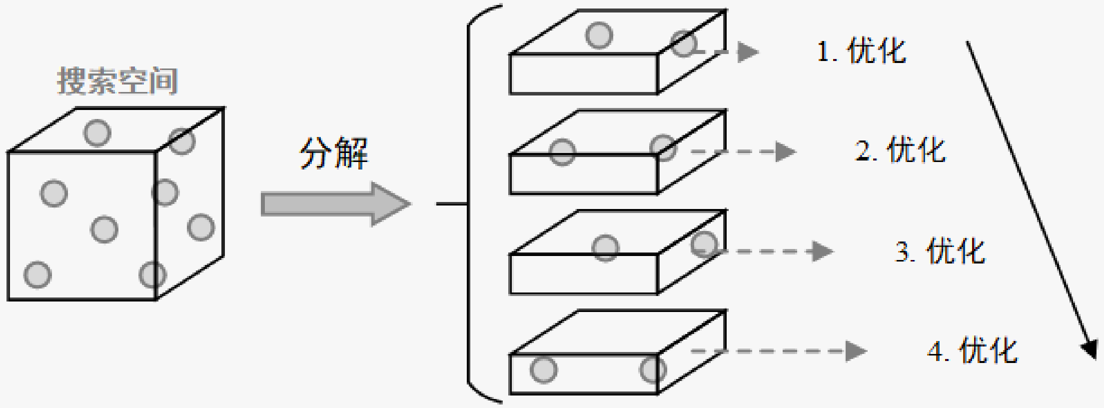
- 针对不同类型问题设计不同策略，提升进化算法收敛速度

变量聚类 | 问题重构 | 稀疏搜索 | 硬件加速 | 梯度辅助

第 2 章 大规模多目标进化算法

变量聚类

- 对大量决策变量进行分组并分别优化，分而治之

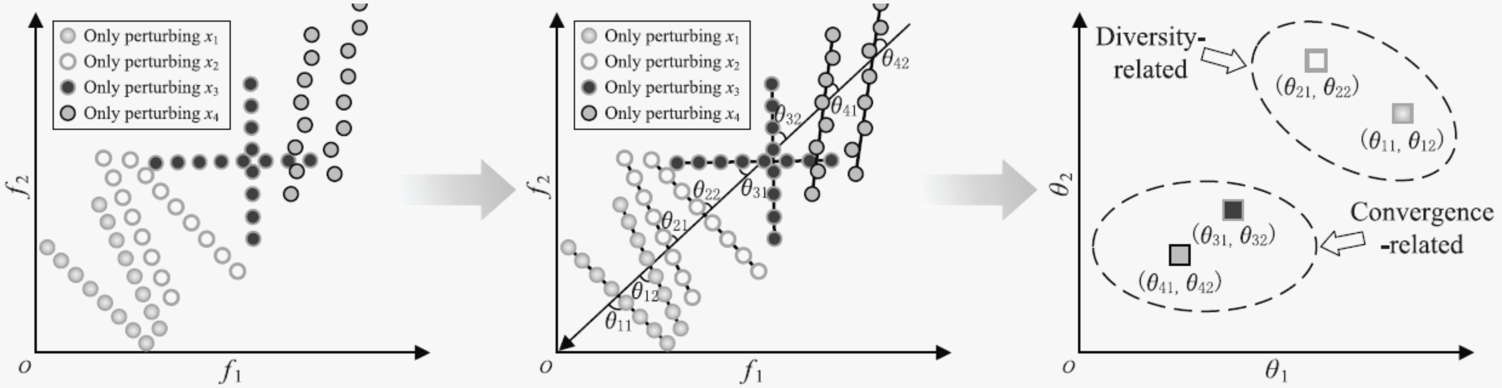


1000个变量的搜索空间		划分为100组后搜索空间
10^{1000}	→	$100 \times 10^{10} \approx 10^{12}$

变量聚类

如何划分变量

- 将变量聚类为收敛性相关和多样性相关，并分别优化



- 收敛性相关变量 ⟶ 收敛性优化
- 多样性相关变量 ⟶ 多样性优化

- 将收敛性相关变量进一步划分为不相关的多组，并分别优化

若变量 x_1 和 x_2 不相关，则一定存在 a_1, a_2, b_1, b_2 满足

$$\begin{cases} f(\mathbf{x})|_{x_i=a_1,x_j=b_1} > f(\mathbf{x})|_{x_i=a_2,x_j=b_1} \\ f(\mathbf{x})|_{x_i=a_1,x_j=b_2} < f(\mathbf{x})|_{x_i=a_2,x_j=b_2} \end{cases}$$

(1) X. Zhang, Y. Tian, R. Cheng*, and Y. Jin, A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization, *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 97-112.

变量聚类

- 基于变量聚类的进化算法LMEA，在<u>**变量相关性弱**</u>的大规模多目标优化问题上，可收敛至全局最优

算法主流程

Algorithm 1: Main Framework of LMEA

Input: N (population size), $nSel$ (number of selected solutions for decision variable clustering), $nPer$ (number of perturbations on each solution for decision variable clustering), $nCor$ (number of selected solutions for decision variable interaction analysis)

Output: P (final population)

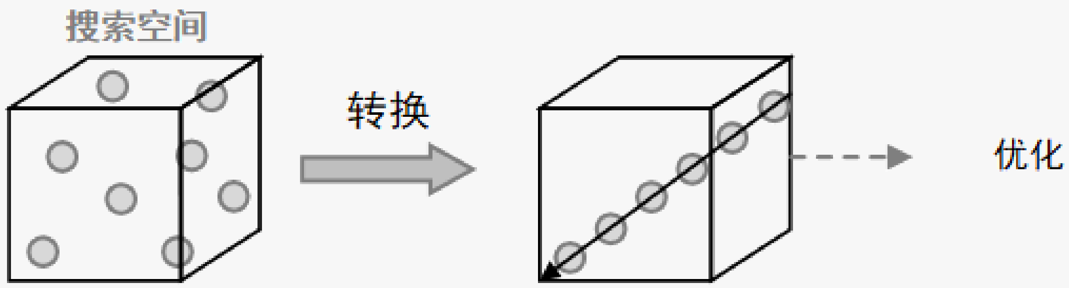
$P \leftarrow Initialize(N)$;
$[DV, CV] \leftarrow VariableClustering(P, nSel, nPer)$;
$subCVs \leftarrow InteractionAnalysis(P, CV, nCor)$;
while *termination criterion not fulfilled* **do**
  $P \leftarrow ConvergenceOptimization(P, subCVs)$;
  $P \leftarrow DiversityOptimization(P, DV)$;

对比实验结果

Problem	Obj.	Dec.	MOEA/D	NSGA-III	KnEA	MOEA/DVA	LMEA
DTLZ5	5	100	4.5161e-2(9.31e-7)–	1.4957e-1(2.58e-2)–	2.7185e-1(4.11e-2)–	2.0440e-1(5.06e-4)–	4.1162e-3(1.44e-4)
		500	4.5161e-2(1.04e-6)–	1.9413e-1(1.78e-2)–	3.1740e-1(6.11e-2)–	2.0469e-1(5.20e-8)–	4.0861e-3(1.48e-4)
		1000	4.5162e-2(3.32e-7)–	2.0606e-1(1.10e-2)–	3.8913e-1(6.77e-2)–	2.0461e-1(1.36e-4)–	4.0729e-3(9.90e-5)
	10	100	4.9994e-2(2.41e-4)–	3.1946e-1(2.03e-2)–	3.6432e-1(5.71e-2)–	1.8877e-1(1.87e-4)–	2.3954e-3(6.95e-5)
		500	5.0407e-2(4.16e-4)–	5.2642e-1(2.04e-2)–	3.6389e-1(5.43e-2)–	1.8866e-1(3.30e-4)–	2.2721e-3(4.47e-5)
		1000	5.0759e-2(2.12e-5)–	6.2093e-1(1.14e-2)–	4.1806e-1(5.07e-2)–	1.8880e-1(2.03e-4)–	2.0713e-3(6.98e-5)
DTLZ6	5	100	1.4970e-1(3.14e-2)–	2.5642e-1(2.29e-2)–	5.8811e-1(1.34e-1)–	1.8236e-1(2.43e-6)–	3.9943e-3(2.14e-4)
		500	1.3010e+0(1.04e-1)–	4.9939e-1(1.89e-2)–	7.2754e-1(1.43e-1)–	1.8236e-1(4.25e-7)–	4.5127e-3(1.22e-3)
		1000	2.7140e+0(1.97e-1)–	6.5774e-1(2.19e-2)–	1.5085e+0(4.53e-1)–	1.8236e-1(5.91e-7)–	3.9747e-3(2.29e-4)
	10	100	6.7510e-2(1.85e-2)–	7.2120e+0(1.35e+0)–	3.7560e+0(9.56e-1)–	1.6531e-1(4.09e-2)–	2.4477e-3(5.11e-4)
		500	1.1735e+0(2.52e-1)–	8.7171e+1(4.88e+0)–	6.3085e+0(2.18e+0)–	1.2750e-1(5.33e-2)–	3.0711e-3(7.20e-4)
		1000	2.6191e+0(5.73e-1)–	1.9202e+2(9.83e+0)–	4.8989e+0(2.54e+0)–	1.1844e-1(2.26e-2)–	3.7077e-3(1.66e-3)
WFG3	5	100	2.0705e+0(8.91e-2)–	7.7137e-1(3.83e-2)–	6.2696e-1(4.24e-1)–	2.3769e+0(7.55e-3)–	1.2581e-1(2.91e-2)
		500	2.2789e+0(6.03e-2)–	8.6982e-1(2.37e-2)–	2.2284e-1(2.56e-2)–	2.4699e+0(9.15e-3)–	1.1736e-1(3.58e-2)
		1000	2.3370e+0(8.10e-2)–	8.8753e-1(2.53e-2)–	4.6414e-1(1.49e-1)–	2.4410e+0(3.52e-2)–	1.2493e-1(2.41e-2)
	10	100	3.4569e+0(1.29e-1)–	3.0344e+0(5.71e-2)–	2.2907e+0(7.93e-1)–	3.4846e+0(2.45e-2)–	1.8542e-1(5.96e-2)
		500	3.8106e+0(8.24e-2)–	3.1112e+0(5.04e-2)–	1.6148e+0(5.53e-1)–	3.5264e+0(9.75e-2)–	4.8685e-1(5.49e-2)
		1000	3.9456e+0(7.23e-2)–	3.1454e+0(4.20e-2)–	1.9861e+0(1.27e+0)–	3.5070e+0(1.17e-1)–	6.9330e-1(1.16e-1)
UF9	3	100	2.9851e-1(1.58e-2)–	2.2030e-1(9.19e-2)–	5.3546e-1(1.39e-1)–	4.3517e-2(2.50e-6)+	5.7008e-2(8.91e-3)
		500	3.1975e-1(2.92e-2)–	3.1029e-1(7.27e-2)–	4.6017e-1(1.19e-1)–	4.3516e-2(9.76e-7)+	5.3626e-2(6.94e-3)
		1000	3.0557e-1(8.39e-2)–	3.7850e-1(4.21e-2)–	5.3607e-1(8.03e-2)–	4.3516e-2(7.00e-7)+	5.1231e-2(4.50e-3)
UF10	3	100	5.9354e-1(1.50e-1)–	3.3482e-1(8.13e-2)–	7.5510e-1(1.49e-1)–	1.1024e-1(2.92e-3)+	1.6632e-1(1.45e-2)
		500	6.3119e-1(1.92e-1)–	3.6779e-1(8.36e-2)–	1.3142e+0(8.69e-1)–	1.0158e-1(8.55e-4)+	1.5547e-1(4.99e-3)
		1000	5.6232e-1(2.48e-1)–	4.2148e-1(1.10e-1)–	9.1794e-1(1.35e-1)–	1.0277e-1(1.01e-3)+	1.6924e-1(9.48e-3)
+/–/≈			0/24/0	0/24/0	0/24/0	6/18/0	

问题重构

- 对大规模搜索空间直接进行降维，从而减小问题难度



1000个变量的搜索空间

$$10^{1000}$$

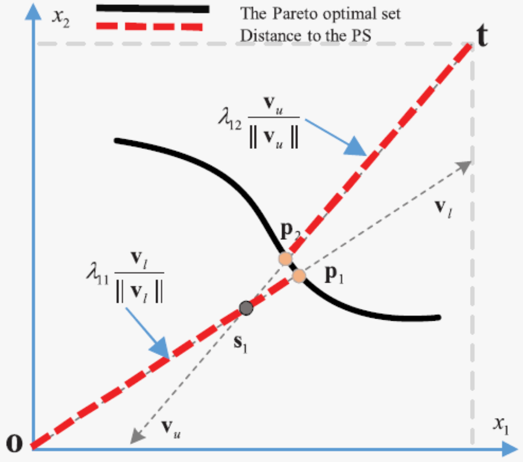
减小到 2 维搜索空间

$$10^{2}$$

问题重构

如何降维

- 对于一个当前最优解，仅在穿过其的两个直线方向上进行搜索



$$\mathbf{x}_i^l = \mathbf{o} + w_i^l \frac{\mathbf{x}_i - \mathbf{o}}{\|\mathbf{x}_i - \mathbf{o}\|}\|\mathbf{t} - \mathbf{o}\|$$

$$\mathbf{x}_i^u = \mathbf{t} - w_i^u \frac{\mathbf{t} - \mathbf{x}_i}{\|\mathbf{t} - \mathbf{x}_i\|}\|\mathbf{t} - \mathbf{o}\|$$

- 同时优化 k 个解 $\mathbf{x}$ 的HV值，则降维后仅优化 $2k$ 个权重 w

$$\max\ f(w_1^l, w_1^u, w_2^l, w_2^u, \ldots, w_k^l, w_k^u) = H(\mathbf{x}_1^l, \mathbf{x}_1^u, \mathbf{x}_2^l, \mathbf{x}_2^u, \ldots, \mathbf{x}_k^l, \mathbf{x}_k^u)$$

(2) C. He, L. Li, Y. Tian, X. Zhang, R. Cheng*, Y. Jin, and X. Yao, Accelerating large-scale multiobjective optimization via problem reformulation, *IEEE Transactions on Evolutionary Computation*, 2019, 23(6): 949-961.

问题重构

- 基于问题重构的进化框架LSMOF，在<u>评价次数很少时</u>可快速收敛至局部最优

算法主流程

Algorithm 1 Main Framework of the Proposed LSMOF

Input: Z (original LSMOP), FE_{max} (total FEs), Alg (embedded MOEA), N (population size for Alg), r (number of reference solutions), tr (threshold).
Output: P (final population).

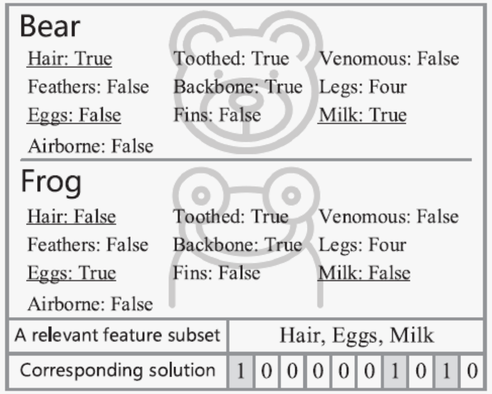
$P \leftarrow$ Initialization(N, Z)
/************First Stage************/
while $t \le tr \times FE_{max}$ **do**
  $Z' \leftarrow$ Problem_Reformulation(P, r, Z)
  $A, \Delta t \leftarrow$ Single_Objective_Optimization(Z')
  $P \leftarrow$ Environmental_Selection$(A \bigcup P, N)$
  $t \leftarrow t + \Delta t$
end while
/**********Second Stage**********/
$P \leftarrow$ Embedded_MOEA (P, N, Alg, Z)

对比实验结果

Problem	M	D	MOEA/DVA	WOF-NSGA-II	LS-NSGA-II
LSMOP1	2	200	8.66E+0(8.04E-1)−	6.30E-1(9.36E-2)−	5.78E-1(5.32E-2)
		500	1.91E+1(1.00E+0)−	6.58E-1(6.11E-2)−	6.14E-1(2.54E-2)
		1000	2.39E+1(7.84E-1)−	6.79E-1(4.22E-2)−	6.37E-1(1.97E-2)
	3	200	6.26E+0(4.62E-1)−	6.95E-1(1.32E-1)−	5.24E-1(1.35E-2)
		500	9.42E+0(2.89E-1)−	7.09E-1(8.36E-2)−	5.96E-1(1.08E-2)
		1000	1.08E+1(3.22E-1)−	8.01E-1(7.05E-2)−	6.33E-1(1.34E-2)
LSMOP2	2	200	1.51E-1(6.75E-4)−	7.46E-2(4.63E-4)−	3.85E-2(1.08E-3)
		500	7.27E-2(2.30E-4)−	3.30E-2(3.91E-4)−	2.32E-2(6.90E-4)
		1000	4.04E-2(3.87E-4)−	1.92E-2(3.40E-4)−	1.81E-2(5.41E-4)
	3	200	1.23E-1(2.61E-3)+	1.36E-1(3.84E-3)≈	1.38E-1(2.76E-3)
		500	7.89E-2(2.63E-3)+	8.54E-2(3.82E-3)≈	8.71E-2(3.29E-3)
		1000	6.48E-2(2.46E-3)+	7.00E-2(4.28E-3)≈	7.05E-2(3.08E-3)
LSMOP3	2	200	1.71E+1(1.30E+0)−	1.50E+0(6.88E-2)≈	1.54E+0(1.43E-3)
		500	2.87E+1(8.26E-1)−	1.57E+0(1.47E-3)−	1.57E+0(1.05E-3)
		1000	3.36E+1(6.07E-1)−	1.58E+0(1.61E-3)−	1.57E+0(2.28E-4)
	3	200	2.30E+1(3.53E+0)−	8.61E-1(3.38E-4)−	8.40E-1(2.51E-2)
		500	3.60E+1(2.95E+0)−	8.61E-1(1.30E-4)−	8.59E-1(3.26E-3)
		1000	4.02E+1(2.09E+0)−	8.61E-1(7.28E-4)≈	8.61E-1(7.03E-5)
LSMOP4	2	200	6.56E-1(9.76E-3)−	1.33E-1(1.51E-2)−	9.87E-2(1.69E-3)
		500	5.44E-1(1.90E-3)−	8.74E-2(6.83E-3)−	5.05E-2(1.14E-3)
		1000	4.61E-1(6.97E-4)−	5.99E-2(5.57E-3)−	3.20E-2(9.49E-4)
	3	200	3.26E-1(2.31E-3)−	3.15E-1(9.10E-3)−	2.92E-1(8.37E-3)
		500	1.94E-1(5.71E-4)+	2.14E-1(6.87E-3)≈	2.13E-1(4.72E-3)
		1000	1.20E-1(1.96E-4)+	1.39E-1(5.80E-3)≈	1.41E-1(3.63E-3)
LSMOP5	2	200	1.42E+1(6.21E-1)−	7.42E-1(1.14E-6)≈	7.42E-1(1.14E-6)
		500	2.09E+1(5.02E-1)−	7.42E-1(1.14E-6)≈	7.42E-1(1.14E-6)
		1000	2.41E+1(3.40E-1)−	7.42E-1(1.14E-6)≈	7.42E-1(1.14E-6)
	3	200	1.17E+1(9.27E-1)−	5.41E-1(1.02E-3)−	4.88E-1(5.13E-2)
		500	1.70E+1(6.15E-1)−	5.41E-1(4.66E-5)−	5.35E-1(1.23E-2)
		1000	1.91E+1(5.97E-1)−	5.41E-1(7.27E-5)≈	5.49E-1(2.83E-2)

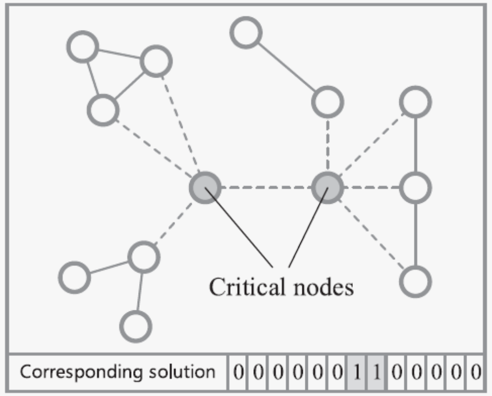
稀疏搜索

- 很多大规模多目标优化问题是稀疏的，即最优解中有大量为零的变量

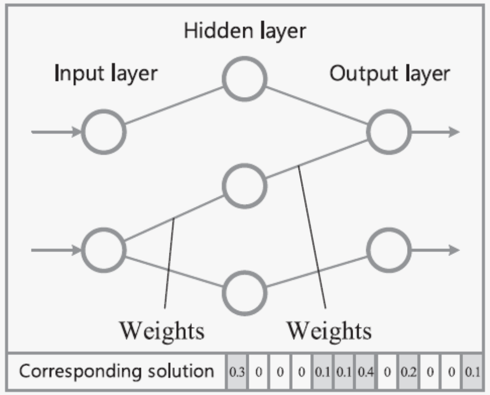


特征选择

模式挖掘



关键点识别

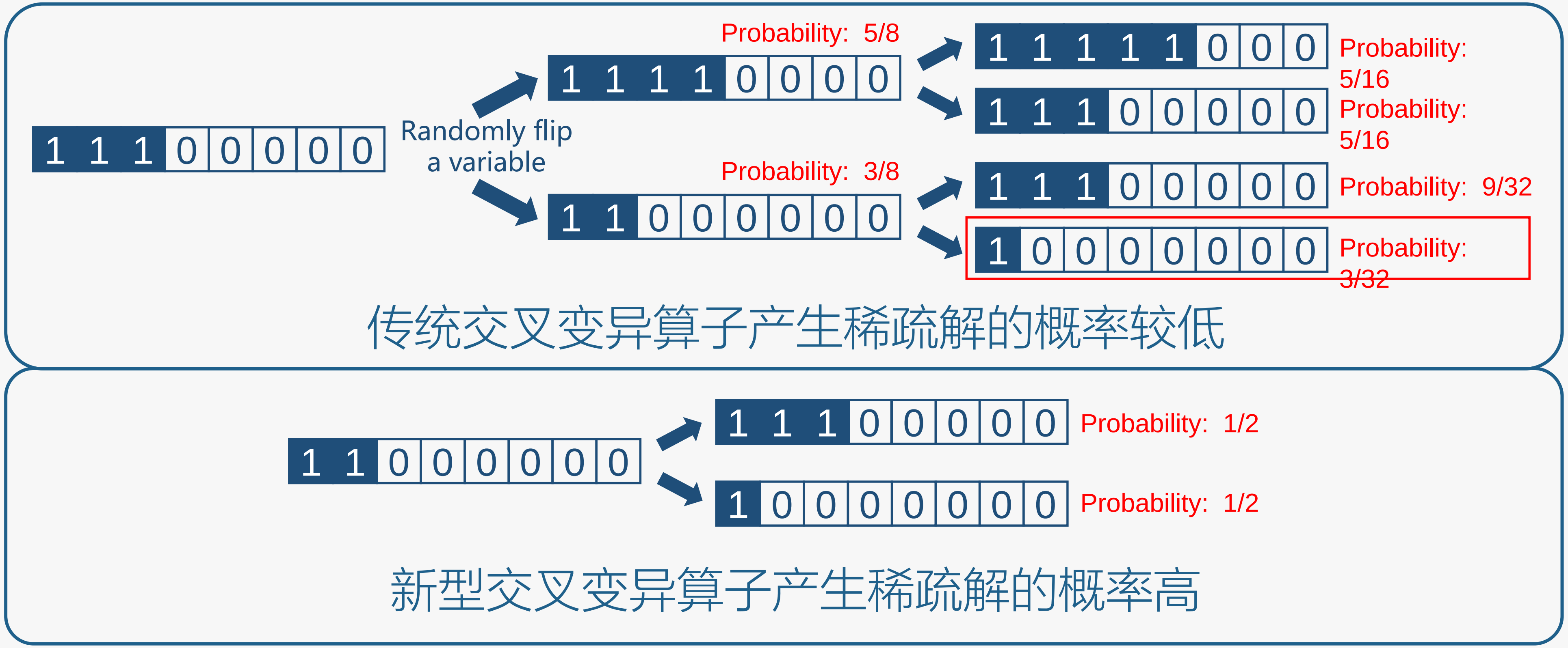


稀疏神经网络

- 若能快速识别其中为零的变量，则可大幅减小搜索空间、提升收敛速度

稀疏搜索

- 首个面向稀疏优化的进化算法SparseEA，提出新型交叉变异算子以更高效地搜索稀疏解

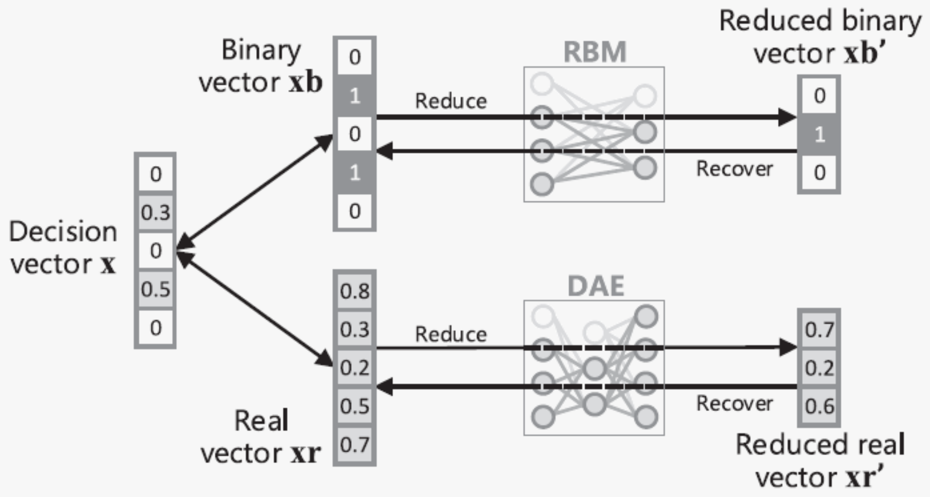


(3) Y. Tian, X. Zhang*, C. Wang, and Y. Jin, An evolutionary algorithm for large-scale sparse multiobjective optimization problems, *IEEE Transactions on Evolutionary Computation*, 2020, 24(2): 380-393.

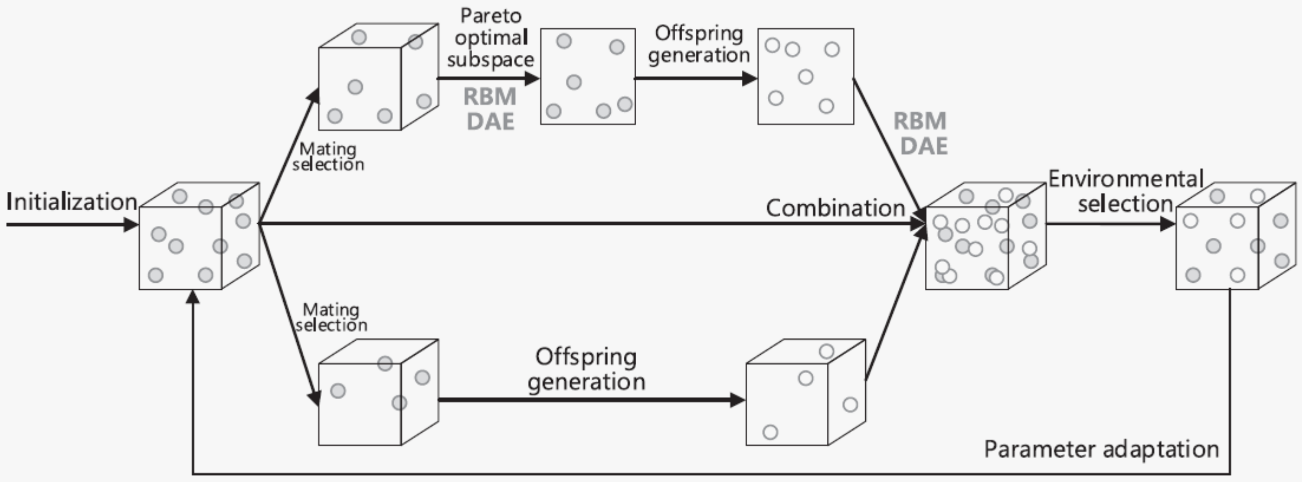
稀疏搜索

- 基于无监督神经网络的进化算法MOEA/PSL，提出无监督降维策略以压缩稀疏搜索空间、提升收敛速度

无监督降维策略



算法主流程

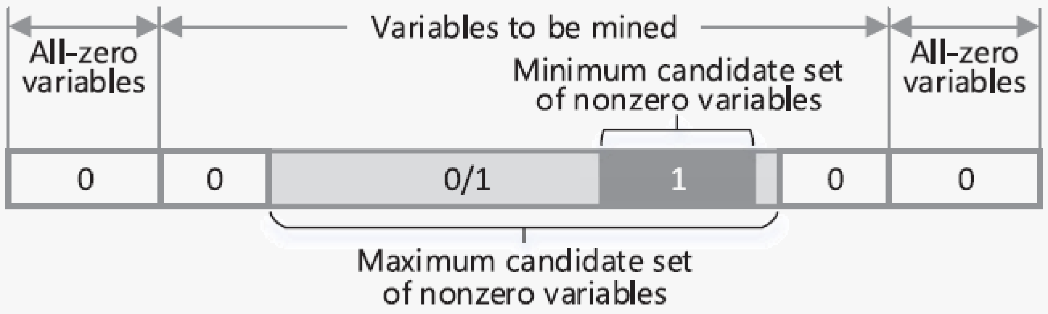


(4) Y. Tian, C. Lu, X. Zhang*, K. C. Tan, and Y. Jin, Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks, *IEEE Transactions on Cybernetics*, 2021, 51(6): 3115-3128.

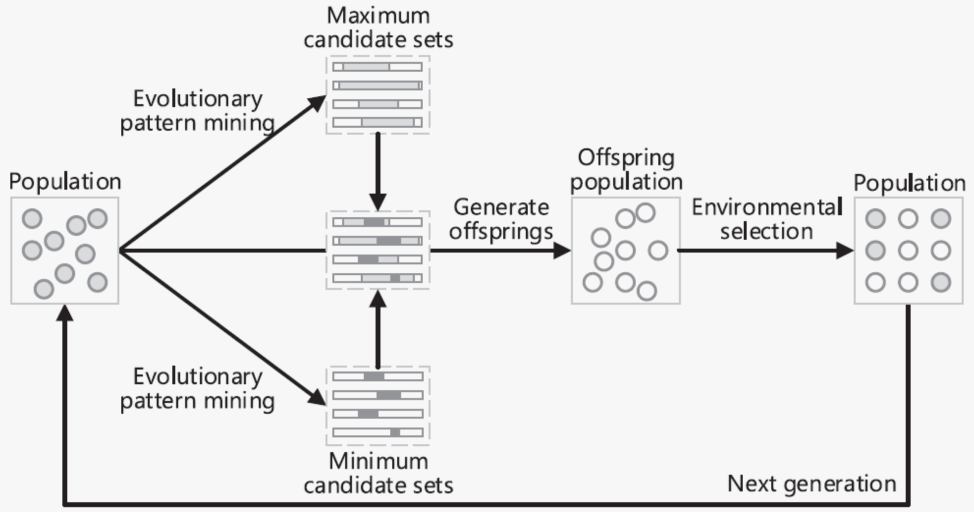
稀疏搜索

- 基于模式挖掘的进化算法PM-MOEA，提出进化模式挖掘策略以发现非稀疏搜索子空间、提升收敛速度

模式挖掘策略



算法主流程



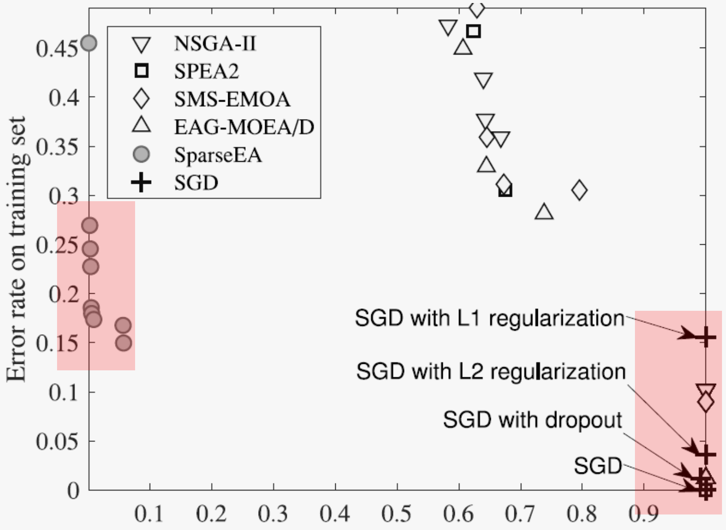
(5) Y. Tian, C. Lu, X. Zhang*, F. Cheng, and Y. Jin, A pattern mining based evolutionary algorithm for large-scale sparse multi-objective optimization problems, *IEEE Transactions on Cybernetics*, 2022, 52(7): 6784-6797.

- 实验结果表明，以上进化算法在<u>**稀疏大规模多目标优化问题**</u>上，可大幅提升优化性能

对比实验结果

Problem (D)	NSGA-II	MOEA/D-DE	MOEA/DVA	LMEA	WOF-SMPSO	LSMOF-NSGA-II	IM-MOEA	LMOCSO	SparseEA	MOEA/PSL	PM-MOEA
SMOP1 (100)	1.9337e-1	4.3614e-1	7.8851e-1	5.9369e-1	2.3918e-1	1.4151e-1	3.7767e-1	4.1068e-1	6.9883e-2	**4.8689e-2**	9.4315e-2
SMOP2 (100)	3.0427e-1	6.8635e-1	8.1987e-1	6.7602e-1	3.7089e-1	2.4525e-1	5.4250e-1	6.5158e-1	1.3044e-1	**5.5829e-2**	9.0096e-2
SMOP3 (100)	2.9980e-1	3.3553e-1	6.9653e-1	4.9054e-1	2.5190e-1	2.0252e-1	5.7921e-1	3.5256e-1	8.6080e-2	**7.0560e-2**	1.0548e-1
SMOP4 (100)	1.6997e-1	6.4059e-1	7.5967e-1	5.4760e-1	3.3175e-1	1.1802e-1	3.4479e-1	5.2913e-1	6.5248e-2	**4.9980e-2**	5.6740e-2
SMOP5 (100)	1.4063e-1	3.7059e-1	6.1990e-1	4.6552e-1	1.4168e-1	9.6309e-2	3.0165e-1	2.4447e-1	4.6993e-2	4.4203e-2	**3.7296e-2**
SMOP6 (100)	1.3167e-1	4.6350e-1	7.2521e-1	6.0050e-1	1.7961e-1	8.8620e-2	2.7609e-1	3.4374e-1	**4.4372e-2**	4.7910e-2	4.8082e-2
SMOP7 (100)	7.6009e-1	8.7693e-1	1.6762e+0	1.3955e+0	4.1530e-1	6.1877e-1	1.1281e+0	9.0127e-1	2.1786e-1	**1.0306e-1**	2.4001e-1
SMOP8 (100)	1.0803e+0	1.3295e+0	1.8521e+0	1.6576e+0	7.8605e-1	8.5502e-1	1.4568e+0	1.2191e+0	3.2522e-1	**1.5662e-1**	2.9235e-1
SMOP1 (500)	2.0078e-1	4.7565e-1	6.5680e-1	5.2181e-1	2.5840e-1	1.4826e-1	3.8710e-1	4.1421e-1	7.7962e-2	**6.8524e-2**	7.4613e-2
SMOP2 (500)	2.2361e-1	5.3837e-1	6.2014e-1	4.9237e-1	1.7649e-1	1.3007e-1	4.0178e-1	5.1778e-1	6.1991e-2	**4.6055e-2**	7.5849e-2
SMOP3 (500)	2.1226e-1	2.2364e-1	4.5775e-1	3.1473e-1	1.6350e-1	1.1170e-1	3.8878e-1	2.2364e-1	5.4376e-2	**5.0797e-2**	5.1477e-2
SMOP4 (500)	2.9639e-1	7.4486e-1	8.4093e-1	5.4238e-1	2.4601e-1	1.9151e-1	4.1929e-1	6.3857e-1	9.1078e-2	**6.0793e-2**	1.3729e-1
SMOP5 (500)	1.5555e-1	3.6846e-1	6.8601e-1	5.8986e-1	1.0433e-1	1.5748e-1	4.2196e-1	2.7542e-1	4.7540e-2	5.6343e-2	**4.0170e-2**
SMOP6 (500)	1.3524e-1	4.9684e-1	5.8008e-1	4.4997e-1	1.8761e-1	8.2919e-2	2.7406e-1	3.5664e-1	5.9739e-2	4.7825e-2	**3.3376e-2**
SMOP7 (500)	2.9905e-1	3.5295e-1	9.4878e-1	7.4620e-1	1.6603e-1	2.4426e-1	5.9310e-1	4.1848e-1	8.1813e-2	**5.2182e-2**	1.0454e-1
SMOP8 (500)	3.6262e-1	5.0656e-1	9.3498e-1	7.8791e-1	1.8598e-1	3.0491e-1	6.5754e-1	4.3061e-1	1.0165e-1	**7.5297e-2**	8.8172e-2
SMOP1 (1000)	1.5571e-1	3.5225e-1	5.2676e-1	4.3443e-1	1.9660e-1	1.0731e-1	3.1717e-1	3.3554e-1	**4.8398e-2**	5.3307e-2	5.3906e-2
SMOP2 (1000)	2.5495e-1	5.2097e-1	5.9670e-1	4.9781e-1	2.0615e-1	1.5315e-1	4.0175e-1	5.3552e-1	9.2543e-2	7.0190e-2	**4.0488e-2**
SMOP3 (1000)	2.5065e-1	2.5065e-1	4.6366e-1	3.4697e-1	1.9917e-1	1.4570e-1	4.0951e-1	2.5065e-1	7.5598e-2	6.2881e-2	**5.5213e-2**
SMOP4 (1000)	2.9679e-1	7.9097e-1	8.7980e-1	5.8962e-1	2.4536e-1	1.9047e-1	4.0549e-1	6.9149e-1	1.0773e-1	**5.3168e-2**	1.4350e-1
SMOP5 (1000)	1.8111e-1	3.9310e-1	5.8350e-1	5.9136e-1	1.2368e-1	1.4881e-1	4.1826e-1	2.8160e-1	4.8329e-2	7.0232e-2	**3.6738e-2**
SMOP6 (1000)	1.3829e-1	4.9495e-1	5.5872e-1	5.1353e-1	1.9575e-1	8.9246e-2	2.7406e-1	3.5636e-1	5.7827e-2	5.0686e-2	**3.5731e-2**
SMOP7 (1000)	3.0360e-1	3.5633e-1	9.0563e-1	7.2155e-1	1.7128e-1	2.4907e-1	5.8356e-1	4.1170e-1	6.9131e-2	**6.6191e-2**	1.1651e-1
SMOP8 (1000)	3.6972e-1	5.6807e-1	9.0280e-1	7.3794e-1	1.7119e-1	3.0418e-1	7.2364e-1	4.2391e-1	8.5294e-2	**5.5697e-2**	7.7375e-2
SMOP1 (5000)	3.0581e-1	4.3757e-1	7.1662e-1	6.5268e-1	2.5393e-1	2.0094e-1	4.3473e-1	4.3757e-1	1.1666e-1	8.4255e-2	**7.2000e-2**
SMOP2 (5000)	2.3548e-1	3.9400e-1	4.1265e-1	5.3216e-1	1.3689e-1	1.8393e-1	3.8831e-1	3.9517e-1	6.6315e-2	**5.1404e-2**	6.6043e-2
SMOP3 (5000)	2.0494e-1	2.0494e-1	4.0782e-1	3.3496e-1	1.5823e-1	1.6087e-1	3.9645e-1	2.0494e-1	8.5894e-2	**4.6221e-2**	6.6826e-2
SMOP4 (5000)	2.9513e-1	6.3529e-1	7.4225e-1	6.3870e-1	2.0108e-1	2.4033e-1	3.9630e-1	5.7324e-1	8.9054e-2	**7.9108e-2**	1.4608e-1
SMOP5 (5000)	1.8998e-1	3.9632e-1	4.8896e-1	6.4584e-1	1.3207e-1	1.3989e-1	4.4364e-1	2.8988e-1	4.8379e-2	8.3544e-2	**4.6288e-2**
SMOP6 (5000)	2.1177e-1	4.6564e-1	5.0253e-1	7.2834e-1	1.4962e-1	1.2542e-1	2.8417e-1	3.6945e-1	6.1343e-2	6.5937e-2	**3.5237e-2**
SMOP7 (5000)	3.7257e-1	2.9600e-1	1.0383e+0	9.1123e-1	1.5326e-1	2.3453e-1	5.0184e-1	3.1529e-1	1.1192e-1	**7.0545e-2**	1.1669e-1
SMOP8 (5000)	2.7640e-1	4.3124e-1	1.0415e+0	9.3653e-1	1.7614e-1	2.2673e-1	5.5521e-1	3.2799e-1	1.1516e-1	7.5387e-2	**4.4007e-2**
Average ranking	5.7500	8.3125	10.6875	9.4688	4.8438	4.4688	7.9375	7.7500	2.4375	1.5313	2.0313

同时最小化训练误差和L0正则化，可有效防止过拟合



硬件加速

- 对于超大规模问题(一百万变量)，不仅要考虑算法的收敛速度，更要考虑算法的执行效率
- 首个超大规模进化算法SLMEA中，提出快速聚类的策略对所有变量进行分组，该策略具有线性时间复杂度

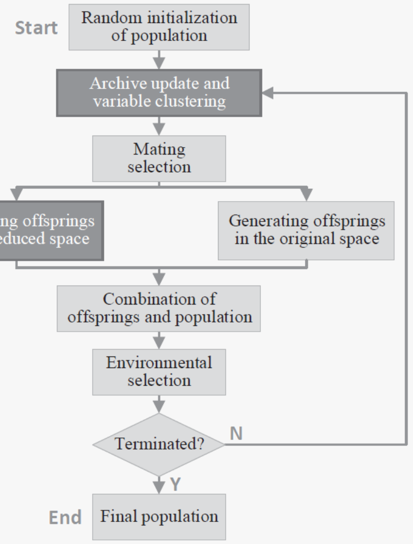
Algorithm 3: $Clustering(A, K)$

Input: A (current archive), K (number of groups)
Output: $Group$ (groups of variables)

$B \leftarrow$ A $|A| \times D$ matrix containing the binary vectors of all solutions in A ;   //D denotes the number of variables
$Sparsity \leftarrow \mathrm{sum}(B)$; //sum of each column of B
$c \leftarrow \mathrm{argmin}_b\,|Sparsity_b - 0.5|$;
$B' \leftarrow \mathrm{repmat}(B_{\cdot c}, D)$; //Repeat the c-th column of B for D columns
$Sim \leftarrow \mathrm{sum}\left(\frac{(1-B)\cdot B'+B\cdot(1-B')}{(1-B)\cdot B'+B\cdot(1-B')+B\cdot B'}\right)$;
$rank \leftarrow \mathrm{sort}(Sim)$; //Rank of variables in Sim
$Group_1 \leftarrow \{b|Sparsity_b = 1\}$;
$Group_2 \leftarrow \{b|Sparsity_b = 0\}$;
$rank \leftarrow rank \setminus Group_1 \setminus Group_2$;
$l \leftarrow \left\lceil \frac{|rank|}{K-2} \right\rceil$;
for $i = 3$ *to* K **do**
  $Group_i \leftarrow \{rank_{1+(i-3)l}, \cdots, rank_{\min(|rank|,(i-2)l)}\}$;
return $Group$;

	$Group_1$		$Group_3$ — $Group_K$											$Group_2$		
Binary variable	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Solution 1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0
2	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0
3	1	1	0	1	1	0	0	0	1	0	1	0	1	0	0	0
4	1	1	1	0	1	1	1	1	0	1	0	0	0	0	0	0
5	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0
6	1	1	1	1	1	0	1	1	0	0	0	1	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0
9	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0
Sparsity	1	1	0.9	0.8	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.1	0	0	0

Reference variable



- 对每组变量赋予相同的值并优化，可达到大幅减小搜索空间的目的

(6) Y. Tian, Y. Feng, X. Zhang*, and C. Sun, A fast clustering based evolutionary algorithm for super-large-scale sparse multi-objective optimization, *IEEE/CAA Journal of Automatica Sinica*, 2022, 9(4): 1-16.

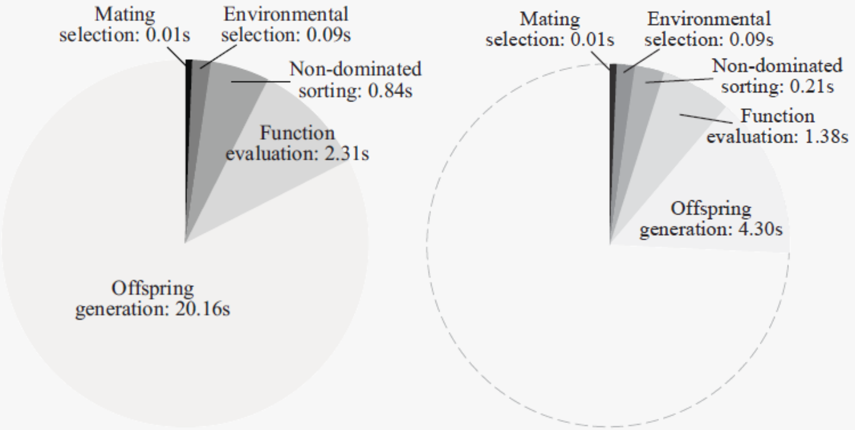
硬件加速

- SLMEA减少复杂逻辑判断、并将所有操作矩阵化，从而可以支持GPU加速，在<u>**稀疏超大规模优化问题**</u>上效率提升显著

对比实验结果

Problem	*D*	NSGA-II	CCGDE3	LMOCSO	WOF-SMPSO	SparseEA	MOEA/PSL	SLMEA
SMOP1	10 000	8.5255e-1 (2.11e-2) –	1.1139e+0 (4.97e-2) –	7.1073e-1 (1.53e-2) –	2.2035e-1 (3.09e-2)–	3.8454e-1 (1.23e-2) –	2.2836e-2 (2.54e-3)–	1.7755e-2 (5.56e-3)
	100 000	1.3335e+0 (4.90e-3) –	1.1403e+0 (1.04e-2) –	7.2615e-1 (1.19e-2) –	3.2690e-1 (1.90e-2)–	6.5167e-1 (7.39e-3) –	3.0330e-2 (4.64e-3)+	3.8269e-2 (1.07e-2)
	1 000 000	1.6088e+0 (9.99e-4) –	1.1664e+0 (0.00e+0) –	7.3840e-1 (0.00e+0) –	2.8202e-1 (6.78e-3)–	8.1868e-1 (4.55e-4) –	4.8356e-2 (8.46e-3)–	3.3165e-2 (4.57e-3)
SMOP2	10 000	1.6723e+0 (6.57e-3) –	2.1214e+0 (6.69e-2) –	2.0303e+0 (4.06e-3) –	3.4487e-1 (1.34e-1)–	8.1988e-1 (1.14e-2) –	1.0195e-1 (1.39e-3)–	8.5880e-2 (3.38e-2)
	100 000	2.0483e+0 (5.26e-3) –	2.1093e+0 (3.35e-3) –	2.0365e+0 (1.24e-2) –	1.0149e+0 (2.34e-1)–	1.0585e+0 (2.90e-3) –	1.0525e-1 (5.84e-3)+	1.2244e-1 (1.09e-2)
	1 000 000	2.2175e+0 (1.01e-3) –	2.1489e+0 (0.00e+0) –	2.0289e+0 (0.00e+0) –	1.0516e+0 (1.07e-1)–	1.1651e+0 (0.00e+0) –	1.2154e-1 (8.55e-3)+	1.4765e-1 (3.54e-3)
SMOP3	10 000	2.0604e+0 (1.82e-2) –	2.1404e+0 (3.97e-2) –	1.7561e+0 (1.44e-2) –	7.0353e-1 (2.06e-3)–	2.5800e+0 (1.41e-2) –	4.7620e-2 (7.74e-3)–	3.1226e-2 (1.65e-2)
	100 000	2.4274e+0 (3.41e-3) –	2.1163e+0 (3.13e-3) –	1.7635e+0 (3.95e-3) –	7.0114e-1 (2.24e-4)–	2.8156e+0 (3.18e-3) –	4.8219e-1 (8.67e-2)–	3.4858e-2 (1.11e-2)
	1 000 000	2.5806e+0 (3.95e-4) –	2.1367e+0 (0.00e+0) –	1.7673e+0 (0.00e+0) –	7.0095e-1 (1.02e-6)–	2.9037e+0 (3.47e-4) –	5.1837e-2 (8.83e-3)–	2.6745e-2 (1.49e-3)
SMOP4	10 000	8.2280e-1 (3.57e-3) –	1.0483e+0 (4.56e-2) –	1.0378e+0 (1.60e-2) –	2.8248e-2 (3.98e-2)–	2.6116e-1 (4.86e-3) –	4.7853e-3 (6.10e-5)+	5.3671e-3 (7.43e-4)
	100 000	1.0069e+0 (3.79e-3) –	1.0606e+0 (1.37e-3) –	1.0321e+0 (1.54e-2) –	3.4081e-1 (5.82e-2)–	3.7765e-1 (2.23e-3) –	4.9729e-3 (2.32e-4)+	5.9332e-3 (1.47e-3)
	1 000 000	1.0912e+0 (8.23e-4) –	1.0829e+0 (0.00e+0) –	1.0447e+0 (0.00e+0) –	3.5733e-1 (0.00e+0)–	4.2600e-1 (3.24e-5) –	9.7073e-3 (5.81e-4)–	9.3905e-3 (4.72e-4)
SMOP5	10 000	6.0886e-1 (4.25e-3) –	6.8227e-1 (2.28e-2) –	4.5992e-1 (5.44e-4) –	3.5439e-1 (2.99e-3)–	2.2960e-1 (4.32e-3) –	8.3291e-3 (1.94e-4)–	7.4829e-3 (1.55e-3)
	100 000	9.1117e-1 (3.27e-3) –	6.8136e-1 (3.42e-3) –	4.6135e-1 (3.70e-4) –	3.6432e-1 (2.40e-3)–	3.8880e-1 (4.04e-3) –	9.4841e-3 (2.90e-4)–	6.5888e-3 (1.39e-3)
	1 000 000	1.0934e+0 (9.99e-4) –	7.0471e-1 (0.00e+0) –	4.6270e-1 (0.00e+0) –	9.7012e-2 (0.00e+0)–	4.9411e-1 (4.01e-4) –	1.6188e-2 (3.59e-4)–	1.5529e-2 (4.70e-3)
SMOP6	10 000	2.5603e-1 (3.54e-3) –	3.5305e-1 (2.27e-2) –	2.2085e-1 (1.68e-3) –	5.8457e-2 (1.27e-2)–	1.0191e-1 (2.31e-3) –	1.1692e-2 (5.95e-4)–	7.3531e-3 (1.35e-3)
	100 000	4.1137e-1 (2.87e-3) –	3.4831e-1 (2.71e-3) –	2.2544e-1 (2.19e-3) –	9.7341e-2 (4.84e-4)–	1.8529e-1 (1.42e-3) –	1.3794e-2 (3.96e-4)–	5.2252e-3 (7.76e-4)
	1 000 000	4.9367e-1 (5.62e-4) –	3.6337e-1 (0.00e+0) –	2.3002e-1 (0.00e+0) –	1.9822e-1 (2.80e-3)–	2.3308e-1 (0.00e+0) –	2.2386e-2 (9.96e-4)+	2.7621e-2 (3.67e-3)
SMOP7	10 000	1.6126e+0 (8.31e-2) –	1.7815e+0 (5.99e-2) –	8.3812e-1 (6.25e-2) –	8.1614e-2 (8.87e-3)–	9.3992e-1 (5.28e-3) –	1.2639e-1 (1.41e-2)–	4.8563e-2 (4.25e-2)
	100 000	2.4672e+0 (4.29e-2) –	1.8551e+0 (2.73e-2) –	8.4147e-1 (2.25e-2) –	2.1584e-1 (1.02e-2)–	1.5216e+0 (1.24e-2)–	2.4075e-1 (6.64e-3)–	1.0216e-2 (3.69e-3)
	1 000 000	3.0333e+0 (5.11e-3) –	1.9036e+0 (0.00e+0) –	8.6488e-1 (0.00e+0) –	5.4422e-1 (1.77e-2) –	1.8954e+0 (1.61e-3) –	1.0446e-1 (4.86e-3)–	5.3954e-2 (6.17e-3)
SMOP8	10 000	3.0912e+0 (3.35e-2) –	3.6289e+0 (4.72e-2) –	3.0709e+0 (1.05e-1) –	5.7141e-1 (2.35e-2)–	2.1975e+0 (2.09e-2) –	3.4412e-1 (6.35e-2)–	3.0189e-1 (3.36e-2)
	100 000	3.5233e+0 (8.99e-3) –	3.6259e+0 (3.90e-3) –	3.1002e+0 (4.92e-2) –	6.4938e-1 (1.31e-1)–	2.7164e+0 (7.09e-3) –	3.8808e-1 (8.60e-2)–	3.3533e-1 (2.06e-2)
	1 000 000	3.7057e+0 (9.92e-4) –	3.6743e+0 (0.00e+0) –	3.2062e+0 (0.00e+0) –	4.2363e-1 (0.00e+0)–	2.8873e+0 (4.24e-3) –	4.5537e-1 (7.97e-3)–	4.5356e-1 (7.22e-3)
+/ – / ≈		0/24/0	0/24/0	0/24/0	0/24/0	0/24/0	6/18/0	

基于GPU加速的进化算法可大幅缩短运行时间



梯度辅助

- 在梯度已知的情况下，数学规划方法的收敛速度远超进化算法

根据梯度获得雅可比矩阵

$$J_{\mathbf{f}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_D} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_M(\mathbf{x})}{\partial x_1} & \frac{\partial f_M(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_M(\mathbf{x})}{\partial x_D} \end{bmatrix}$$

对于黑盒函数，利用有限差分估计梯度

$$\frac{\partial f_i(\mathbf{x})}{\partial x_j} \approx \frac{f_i(\mathbf{x}_{j+\epsilon}) - f_i(\mathbf{x})}{\epsilon}$$

- 利用数学规划方法辅助进化算法，可大幅提升其收敛速度
  - 数学规划 → 保证收敛性
  - 进化算法 → 保证多样性

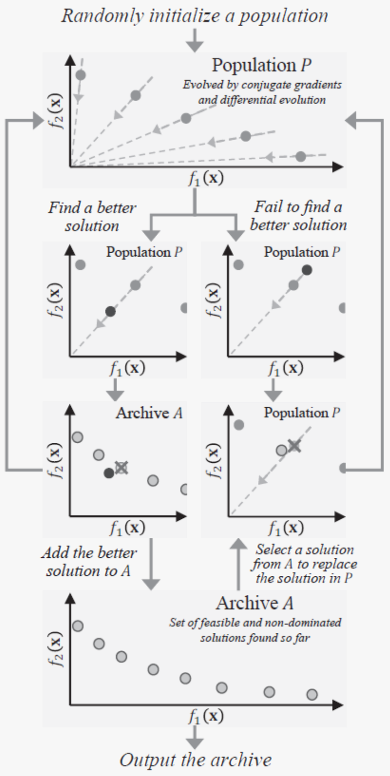
梯度辅助

如何避免陷入局部最优

- MOCGDE建立数学规划和进化算法的联合搜索模式，利用共轭梯度法和差分进化算法产生子代

$$\mathbf{s}=\begin{cases}-\mathbf{g}, & \text{if } \bmod(k,D)=0\\ -\mathbf{g}+\frac{\mathbf{g}\mathbf{g}^T}{\mathbf{g}_0\mathbf{g}_0^T}\mathbf{s}_0, & \text{otherwise}\end{cases}$$

$$y_i = x_i + (1-d_i)\times 0.5^m \times s_i + d_i \times 0.5^m \times (x_i' - x_i'')$$



如何保证种群多样性

- MOCGDE使用小种群进化+大文档存储的结构，在快速收敛的同时保证种群多样性

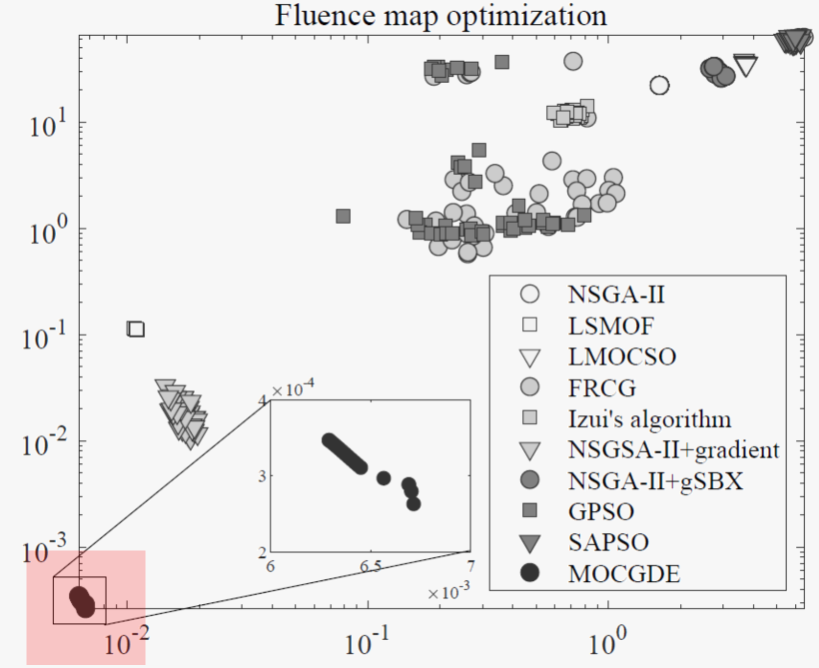
(7) Y. Tian, H. Chen, H. Ma, X. Zhang*, K. C. Tan, and Y. Jin, Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization, *IEEE/CAA Journal of Automatica Sinica*, 2022.

- MOCGDE在**梯度容易计算**的大规模多目标优化问题上，收敛速度提升显著，并能保证多样性

对比实验结果

Problem (M, D)	NSGA-II	LSMOF	LMOCSO	FRCG	Izui's algorithm	MOSD	NSGA-II+ gradient	NSGA-II+ gSBX	MO-EGS	GPSO	SAPSO	MOCGDE
ZDT1 (2,1000)	5.6637e-1 (8.74e-2) −	9.9424e-3 (4.19e-4) −	2.1318e+0 (2.01e-1) −	3.1898e-1 (9.89e-2) −	1.2327e+0 (1.84e-1) −	2.6338e+0 (2.03e-2) −	2.0851e+0 (1.18e-1) −	2.0556e+0 (1.08e-1) −	2.2685e+0 (1.02e-1) −	2.3511e-1 (9.80e-2) −	2.5302e+0 (5.57e-2) −	7.5508e-3 (8.04e-5)
ZDT1 (2,10000)	2.0267e+0 (3.85e-2) −	1.0069e-2 (3.78e-4) −	2.7842e+0 (2.32e-2) −	2.6646e+0 (4.60e-1) −	1.1731e+0 (1.38e-1) −	2.9045e+0 (2.97e-2) −	2.7995e+0 (1.41e-2) −	2.6146e+0 (9.34e-2) −	2.7047e+0 (1.52e-2) −	2.1754e+0 (9.88e-1) −	2.8482e+0 (1.43e-2) −	8.0435e-3 (5.27e-4)
ZDT2 (2,1000)	8.1034e-1 (8.38e-2) −	9.8913e-3 (4.60e-4) −	3.1488e+0 (1.48e-1) −	1.1360e+0 (1.16e-1) −	2.2737e+0 (2.85e-1) −	4.0901e+0 (5.00e-2) −	3.5233e+0 (1.47e-1) −	3.3725e+0 (1.19e-1) −	3.9070e+0 (5.87e-2) −	4.5658e-1 (1.32e-1) −	4.2654e+0 (1.13e-1) −	7.6381e-3 (7.67e-5)
ZDT2 (2,10000)	3.6064e+0 (5.30e-2) −	1.0476e-2 (7.57e-4) −	3.7016e+0 (2.27e-1) −	4.4804e+0 (1.03e-1) −	2.0827e+0 (8.76e-2) −	4.6193e+0 (1.85e-2) −	4.6103e+0 (1.52e-2) −	4.2438e+0 (9.90e-2) −	4.4373e+0 (1.08e-2) −	6.0949e-1 (1.17e-16) −	4.6395e+0 (1.53e-2) −	7.7498e-3 (9.62e-5)
ZDT3 (2,1000)	3.6440e-1 (1.03e-1) −	1.8567e-1 (9.70e-2) −	1.6639e+0 (1.24e-1) −	9.2358e-2 (3.00e-2) −	9.8576e-1 (1.25e-1) −	2.2105e+0 (1.33e-1) −	2.3023e+0 (6.01e-2) −	1.6430e+0 (6.15e-2) −	1.9113e+0 (9.73e-2) −	1.7496e+0 (2.48e-1) −	2.4705e+0 (2.17e-1) −	3.7836e-2 (8.56e-2)
ZDT3 (2,10000)	1.4953e+0 (2.58e-2) −	4.0317e-1 (1.61e-1) −	1.8707e+0 (1.53e-1) −	3.9005e-1 (2.08e-1) −	7.4045e-1 (1.89e-2) −	2.3483e+0 (3.62e-2) −	2.7768e+0 (1.87e-1) −	2.0834e+0 (8.80e-2) −	2.1625e+0 (2.03e-2) −	2.3067e+0 (1.29e-2) −	2.4369e+0 (1.34e-1) −	2.8915e-2 (3.42e-2)
ZDT4 (2,1000)	3.3308e+3 (8.20e+2) ≈	6.6082e+0 (1.08e+1) +	2.0813e+4 (1.66e+3) −	8.2175e+3 (2.42e+2) −	2.4818e+4 (3.83e-12) −	1.5727e+4 (1.62e+3) −	7.1683e+3 (3.81e+3) ≈	1.1975e+4 (4.50e+2) −	1.6412e+4 (2.19e+2) −	1.2755e+4 (9.10e+2) −	1.1222e+3 (6.22e+1) +	4.6546e+3 (3.67e+3)
ZDT4 (2,10000)	1.0087e+5 (1.49e+4) −	5.1586e+1 (1.31e+2) +	1.8482e+5 (5.91e+4) −	8.2703e+4 (4.62e+2) −	2.0777e+5 (1.13e+5) −	1.7282e+5 (1.27e+3) −	9.4186e+4 (2.62e+4) −	1.2327e+5 (2.60e+3) −	1.7454e+5 (4.37e+2) −	1.3836e+5 (6.70e+3) −	1.3476e+4 (2.56e+2) +	6.4416e+4 (2.74e+4)
ZDT6 (2,1000)	4.8456e+0 (5.15e-1) −	5.3309e-1 (9.90e-2) −	7.2895e+0 (1.44e-1) −	6.4301e-1 (3.88e-2) −	6.6209e+0 (1.97e-1) −	7.6365e+0 (2.91e-2) −	7.5490e+0 (2.22e-2) −	7.3556e+0 (4.26e-2) −	7.0886e+0 (7.17e-2) −	6.0954e-1 (8.94e-2) −	7.6660e+0 (2.04e-2) −	2.3628e-1 (1.14e-1)
ZDT6 (2,10000)	7.4259e+0 (3.65e-2) −	9.4675e-1 (2.50e-1) −	7.5330e+0 (1.12e-1) −	3.5280e+0 (3.32e+0) −	6.4634e+0 (7.23e-2) −	7.8740e+0 (9.59e-3) −	7.8629e+0 (1.05e-2) −	7.7257e+0 (5.00e-2) −	7.6858e+0 (1.38e-2) −	5.4056e-1 (1.76e-1) −	7.8820e+0 (6.25e-3) −	2.4182e-1 (1.64e-1)
DTLZ1 (2,1000)	3.7988e+3 (3.94e+2) −	5.6073e+0 (5.43e+0) +	7.3815e+3 (1.70e+3) −	4.9581e+3 (2.07e+2) −	2.0264e+4 (7.59e+2) −	9.2265e+3 (3.41e+3) −	1.2205e+4 (8.82e+2) −	1.2652e+4 (1.72e+3) −	8.8768e+3 (3.02e+1) −	9.9829e+3 (5.26e+3) −	3.1072e+4 (1.55e+3) −	1.3559e+3 (6.68e+2)
DTLZ1 (2,10000)	1.9020e+5 (3.51e+4) −	1.0742e+2 (1.19e+2) +	6.7214e+4 (2.10e+4) ≈	4.9490e+4 (1.41e+3) +	1.9548e+5 (1.25e+4) −	2.1457e+5 (2.32e+4) −	1.2493e+5 (1.77e+1) −	1.1991e+5 (5.61e+3) −	8.9079e+4 (3.98e+2) −	1.2487e+5 (7.26e+0) −	3.0633e+5 (2.19e+4) −	6.5565e+4 (2.40e+4)
DTLZ2 (2,1000)	2.1762e-1 (1.99e-2) −	1.0435e-2 (2.95e-4) ≈	4.0718e+0 (4.73e-1) −	4.7916e-1 (1.84e-1) −	1.2753e+1 (5.63e+0) −	4.9523e+1 (4.62e+0) −	2.2528e-1 (1.20e-1) −	3.2679e+1 (4.78e+0) −	7.6976e+1 (1.43e+0) −	3.4242e-1 (2.01e-14) −	2.6925e+1 (8.84e+0) −	9.8765e-3 (9.45e-4)
DTLZ2 (2,10000)	4.4721e+2 (1.19e+1) −	2.7091e-2 (2.35e-2) ≈	5.0478e+1 (5.67e+0) −	5.4392e+1 (6.57e+1) −	1.9489e+2 (9.14e+0) −	8.0547e+2 (2.97e+0) −	3.1680e-1 (5.99e-2) −	6.1874e+2 (3.33e+1) −	8.1454e+2 (2.42e+0) −	6.0668e-1 (1.91e-1) −	6.8788e+2 (1.93e+1) −	2.4951e-2 (9.40e-3)
DTLZ3 (2,1000)	9.9840e+3 (1.03e+3) −	1.0921e+1 (1.30e+1) +	2.0966e+4 (5.71e+3) −	1.8348e+4 (1.35e+4) −	4.9136e+4 (7.06e+3) −	2.9055e+4 (1.11e+4) −	2.4962e+4 (2.69e+0) −	3.5092e+4 (4.16e+3) −	2.4911e+4 (1.08e+1) −	1.9962e+4 (1.05e+4) −	8.3494e+4 (6.40e+3) −	3.3015e+3 (1.24e+3)
DTLZ3 (2,10000)	5.1175e+5 (1.08e+5) −	3.2402e+2 (4.23e+2) +	1.6028e+5 (4.92e+4) −	1.6179e+5 (8.83e+4) −	5.3206e+5 (4.77e+4) −	6.1801e+5 (7.90e+4) −	2.4984e+5 (1.44e+1) −	3.2938e+5 (1.43e+4) −	2.4986e+5 (1.87e+1) −	2.4975e+5 (1.22e+2) −	8.3735e+5 (9.38e+4) −	5.4209e+4 (3.47e+4)
DTLZ4 (2,1000)	7.8412e-1 (2.63e-1) −	5.2321e-1 (3.54e-1) ≈	1.2267e+1 (6.98e+0) −	7.0328e-1 (1.23e-1) −	3.1764e+1 (1.83e+1) −	6.0628e+1 (1.91e+1) −	5.0228e-1 (2.06e-1) ≈	4.3319e+1 (1.36e+1) −	7.7259e+1 (1.10e+0) −	7.4209e-1 (8.97e-7) −	7.4176e+1 (9.59e+0) −	4.2235e-1 (1.69e-1)
DTLZ4 (2,10000)	4.8683e+2 (6.96e+1) −	6.0058e-1 (3.11e-1) −	2.7314e+2 (6.74e+1) −	7.4209e-1 (1.27e-11) −	1.8237e+1 (3.72e+1) −	7.1392e+2 (4.76e+1) −	4.6209e-1 (1.93e-1) ≈	3.5793e+2 (5.86e+1) −	8.0913e+2 (4.94e+0) −	7.0956e-1 (1.03e-1) −	5.6554e+2 (1.71e+2) −	3.4898e-1 (1.73e-1)
DTLZ5 (2,1000)	3.1729e-1 (1.02e-1) −	1.0878e-2 (3.81e-4) ≈	4.3023e+0 (4.94e-1) −	5.8920e-1 (1.78e-1) −	1.1419e+1 (4.98e+0) −	4.9382e+1 (3.48e+0) −	8.0330e-2 (9.70e-2) −	3.1904e+1 (4.21e+0) −	7.5208e+1 (1.93e+0) −	3.4242e-1 (7.85e-13) −	3.0954e+1 (1.46e+1) −	1.0215e-2 (1.04e-3)
DTLZ5 (2,10000)	4.4325e+2 (1.89e+1) −	1.2747e-2 (3.46e-3) ≈	5.5592e+1 (7.42e+0) −	3.4543e+1 (7.24e+1) −	1.9391e+2 (8.90e+0) −	7.8374e+2 (8.50e+0) −	3.4090e-1 (4.80e-3) −	6.1941e+2 (3.37e+1) −	8.1774e+2 (2.83e+0) −	4.6010e-1 (1.41e-1) −	6.8848e+2 (2.73e+1) −	1.1000e-2 (1.11e-3)
DTLZ6 (2,1000)	6.3535e+2 (7.98e+0) −	1.1391e+2 (9.82e+0) −	3.1397e+2 (3.10e+1) −	1.0000e+0 (1.17e-16) −	8.3147e+2 (1.91e+0) −	8.9463e+2 (9.83e-1) −	4.1233e+1 (2.62e+1) −	7.4530e+2 (6.27e+1) −	4.3209e+2 (4.11e+0) −	7.0921e+1 (6.11e+1) −	5.7244e+2 (1.84e+1) −	4.1444e-1 (1.19e-4)
DTLZ6 (2,10000)	8.5178e+3 (2.33e+1) −	3.3109e+3 (3.08e+2) −	3.9434e+3 (6.27e+2) −	7.0774e+3 (3.73e+3) −	8.8872e+3 (4.86e+0) −	9.0474e+3 (6.37e+0) −	6.8601e+3 (4.97e+2) −	8.9293e+3 (5.14e+1) −	4.7925e+3 (1.21e+1) −	5.7574e+3 (5.00e+2) −	8.2072e+3 (2.70e+1) −	4.1443e-1 (2.40e-4)
DTLZ7 (2,1000)	1.0665e+0 (1.13e-1) −	4.0072e-1 (1.37e-1) −	5.2279e+0 (5.27e-1) −	2.6633e-1 (1.18e-1) −	3.3289e+0 (6.05e-1) −	7.7334e+0 (1.17e-1) −	4.4660e+0 (4.94e-1) −	5.5639e+0 (3.83e-1) −	6.1115e+0 (8.17e-2) −	3.1983e-1 (1.16e-1) −	7.6237e+0 (3.11e-1) −	9.4681e-3 (2.79e-4)
DTLZ7 (2,10000)	5.7861e+0 (8.86e-2) −	4.0174e-1 (1.37e-1) −	6.4588e+0 (3.53e-1) −	5.3949e-1 (1.96e-1) −	3.1057e+0 (1.64e-1) −	7.9608e+0 (3.76e-2) −	7.8697e+0 (9.40e-2) −	7.1688e+0 (2.05e-1) −	7.3912e+0 (7.62e-2) −	6.8268e-1 (1.69e-1) −	7.9696e+0 (4.10e-2) −	1.0086e-2 (9.22e-4)
+/ − / ≈	0/23/1	6/13/5	0/23/1	1/23/0	0/24/0	0/24/0	0/21/3	0/24/0	0/24/0	0/24/0	2/22/0	

在大规模连续优化问题上性能提升显著

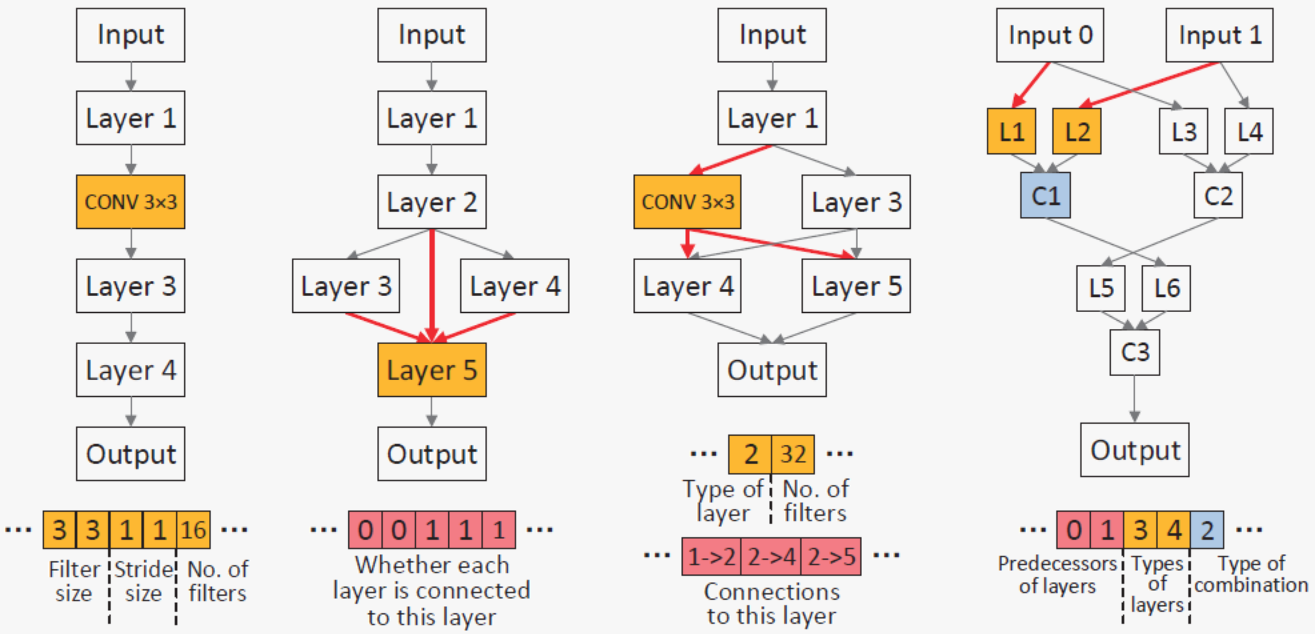


第 3 章

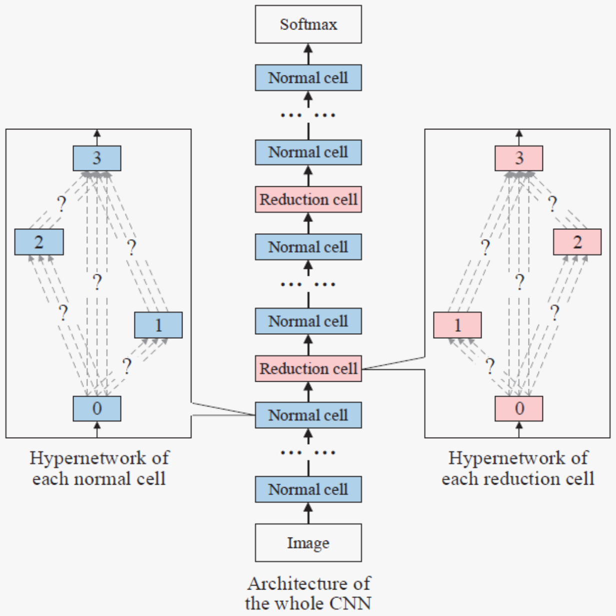
大规模多目标优化应用

神经结构搜索

- 神经网络结构搜索是一个大规模且耗时的优化问题



四类主流的神经网络结构编码方式

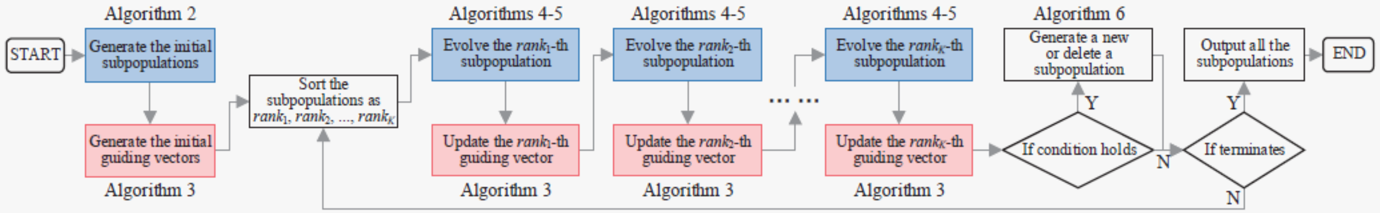


基于one-shot和block的编码方式

神经结构搜索

- 建立面向多模优化的稀疏大规模多目标进化算法MP-MMEA，采用one-shot编码，同时最小化验证误差与网络复杂度

$$\text{Minimize } f_1(A) = f_{valid\_error}(A)$$
$$f_2(A) = f_{complexity}(A)$$



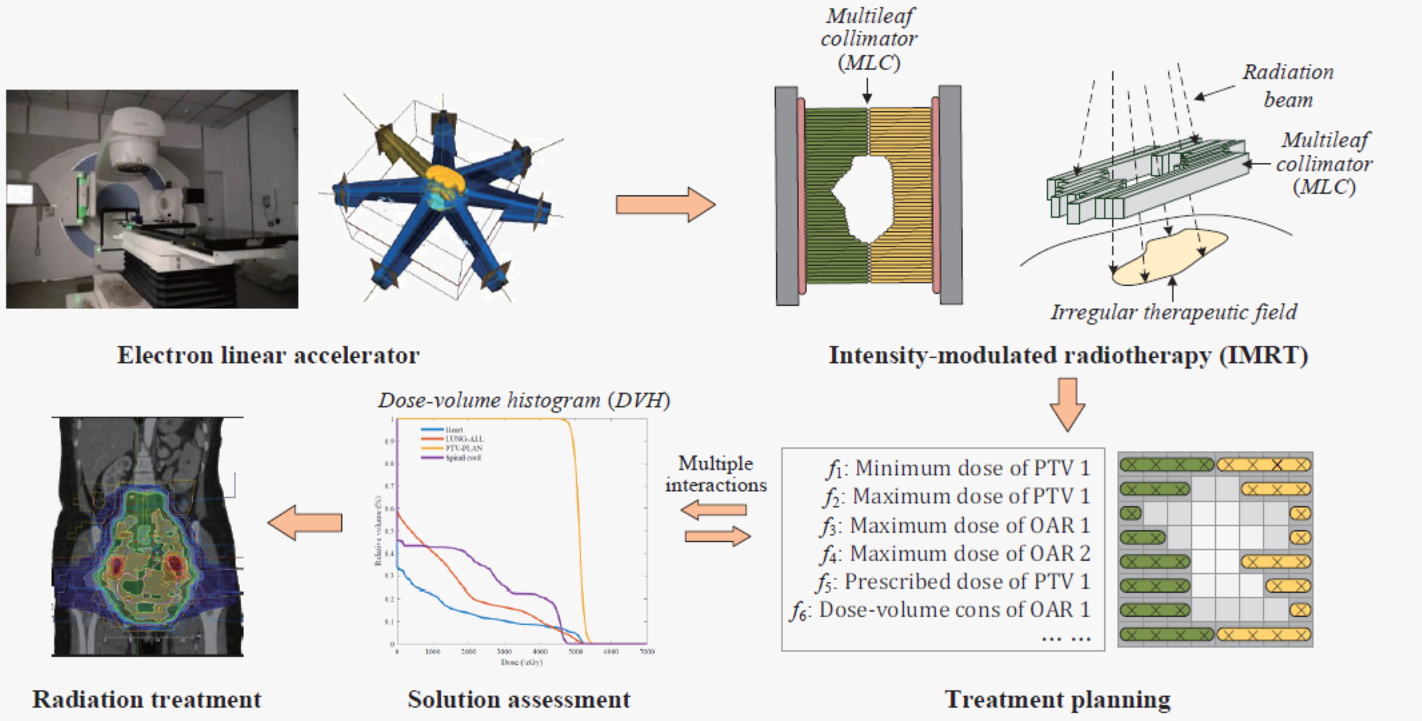
- 将得到的多样化网络结构集成，进一步提升性能

Ensemble method	Baseline 1 [47]	Baseline 2 [48]	MO_Ring_PSO_SCD	SparseEA	MP-MMEA
Best single CNN	89.47%	93.99%	93.68%	91.10%	93.91%
Unweighted average	90.19%	94.55%	94.34%	92.71%	95.03%
Majority vote	N/A	94.33%	94.11%	92.52%	94.79%
Weighted average	90.23%	N/A	94.33%	92.71%	95.03%
Rank based weight average	90.32%	N/A	94.50%	92.51%	94.97%

(8) Y. Tian, R. Liu, X. Zhang*, H. Ma, K. C. Tan, and Y. Jin, A multi-population evolutionary algorithm for solving large-scale multi-modal multi-objective optimization problems, *IEEE Transactions on Evolutionary Computation*, 2021, 25(3): 405-418.

调强放疗计划

- 调强放疗计划是一个大规模超多目标的组合优化问题



- 目标：
  - 最大化癌细胞杀伤率
  - 最小化正常器官损伤率

- 变量：
  - 多叶准直器叶片位置（射野形状）
  - 射线强度、角度

调强放疗计划

$$\text{Minimize } f(x) = (f_1(x), \ldots, f_5(x))$$

$$f_1(x) = \sum_{V \in PTV} MinDose(x, V, d_{min})$$

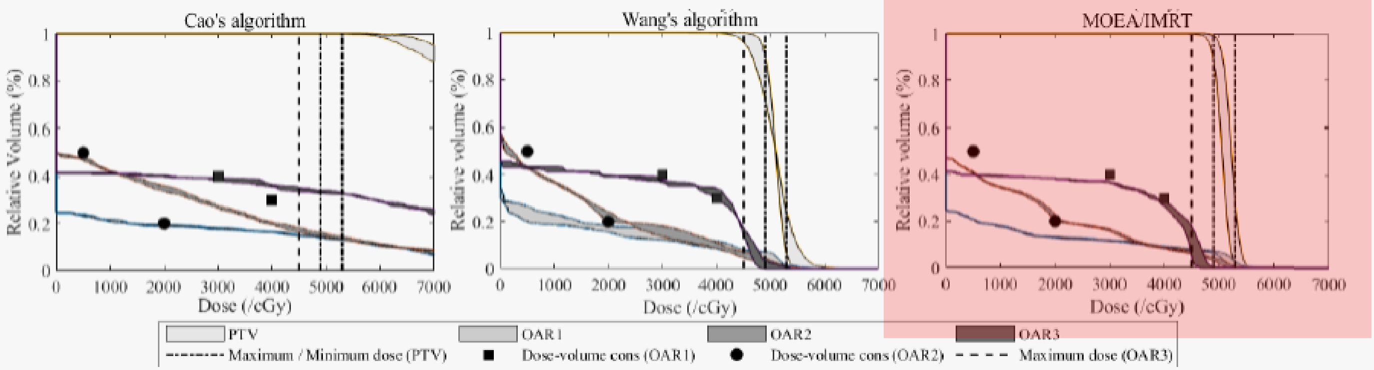
$$f_2(x) = \sum_{V \in PTV} MaxDose(x, V, d_{max}) + \sum_{V \in PTV} DVC(x, V, d_{dvc}, r)$$

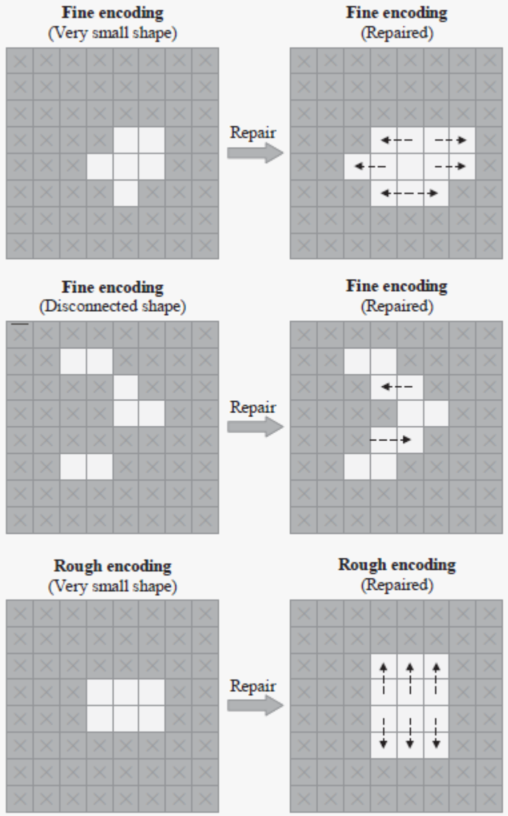
$$f_3(x) = \sum_{V \in OAR} MaxDose(x, V, d_{max})$$

$$f_4(x) = \sum_{V \in OAR} DVC(x, V, d_{dvc}, r)$$

$$f_5(x) = \sum_{V \in PTV} PreDose(x, V, d_{pre})$$

- 建立超多目标优化模型，并设计大规模超多目标进化算法MOEA/IMRT，采用双精度编码+修正的搜索策略，以发现多样化的高效放疗方案
- 该算法可以得到比传统数学规划+贪心策略更加多样化的放疗方案，大幅降低了建模成本与人工调参耗时





(9) Y. Tian, Y. Feng, C. Wang, R. Cao, X. Zhang*, X. Pei, K. C. Tan, and Y. Jin, A large-scale combinatorial many-objective evolutionary algorithm for intensity-modulated radiotherapy planning, *IEEE Transactions on Evolutionary Computation*, 2022.

未来工作

- 大规模搜索空间中的 维数灾难 问题不可避免，但可针对问题特性来设计相应的策略以提升收敛性

- **稀疏**大规模多目标优化问题
- **动态**大规模多目标优化问题
- **昂贵**大规模多目标优化问题
- **鲁棒**大规模多目标优化问题
- **约束**大规模多目标优化问题
- **多模**大规模多目标优化问题

…… ……

(10) Y. Tian, L. Si, X. Zhang*, R. Cheng, C. He, K. C. Tan, and Y. Jin, Evolutionary large-scale multi-objective optimization: A survey, *ACM Computing Surveys*, 2022, 54(8): 174.

未来工作

- 设计全新的算子，从最底层提升在连续优化上的收敛速度

①

空间变换不变性算子推导

Theorem 1 (Translation invariant operator). *A continuously differentiable variation operator $h(x_{1d}, x_{2d}, \ldots)$ is translation invariant if and only if it has the following form:*

$$h(x_{1d}, x_{2d}, \ldots) = x_{1d} + \psi(x_{2d} - x_{1d}, x_{3d} - x_{2d}, \ldots) \ , \quad (34)$$

where ψ can be any continuously differentiable function.

Secondly, a scale invariant operator should satisfy

$$h(a\mathbf{w}) = a \cdot h(\mathbf{w}) \ , \quad (35)$$

according to (33), a translation and scale invariant operator should satisfy

$$\begin{aligned} &aw_1 + \psi(aw_2 - aw_1, aw_3 - aw_2, \ldots) \\ &= aw_1 + a \cdot \psi(w_2 - w_1, w_3 - w_2, \ldots) \ . \end{aligned} \quad (36)$$

Let $\mathbf{v} = (v_1, v_2, \ldots) = (w_2 - w_1, w_3 - w_2, \ldots)$, we have

$$\psi(a\mathbf{v}) = a \cdot \psi(\mathbf{v}), \quad (37)$$

and the first order Taylor series expansion at an arbitrary point $\mathbf{v}_0$ is

$$\begin{aligned} &\psi(\mathbf{v}_0) + (a\mathbf{v} - \mathbf{v}_0)\nabla\psi(\mathbf{v}_0') \\ &= a\psi(\mathbf{v}_0) + a(\mathbf{v} - \mathbf{v}_0)\nabla\psi(\mathbf{v}_0'') \ , \end{aligned} \quad (38)$$

where $\mathbf{v}_0'$ and $\mathbf{v}_0''$ are unknown points determined by $\mathbf{v}_0$. Since (38) holds for any a, we only consider the components excluding a in (38), that is,

$$\psi(\mathbf{v}_0) = \mathbf{v}_0\nabla\psi(\mathbf{v}_0') \quad (39)$$

must hold for any $\mathbf{v}_0$ and $\mathbf{v}_0'$, which means that

$$v_1\frac{\partial\psi}{\partial v_1} + v_2\frac{\partial\psi}{\partial v_2} + \cdots = \psi \ . \quad (40)$$

Let $g(\mathbf{v}, \psi) = 0$, the following homogeneous partial differential equation can be obtained:

$$\psi\frac{\partial g}{\partial \psi} + v_1\frac{\partial g}{\partial v_1} + v_2\frac{\partial g}{\partial v_2} + \cdots = 0 \ , \quad (41)$$

and thus the following first integrals can be obtained:

$$\begin{cases} \ln\psi - \ln v_1 = c_1 \\ \ln v_1 - \ln v_2 = c_2 \\ \ln v_2 - \ln v_3 = c_3 \\ \quad \cdots \quad \cdots \end{cases}, \quad (42)$$

Therefore, the solution of (41) is

$$g = g(\ln\psi - \ln v_1, \ln v_1 - \ln v_2, \ln v_2 - \ln v_3, \ldots) \ . \quad (43)$$

Since $g = 0$, there must exists a function φ such that

$$\ln\psi - \ln v_1 = \varphi(\ln v_1 - \ln v_2, \ln v_2 - \ln v_3, \ldots) \ , \quad (44)$$

hence the generic form of $\psi(\mathbf{v})$ can be determined:

$$\ln\psi(\mathbf{v}) = \ln v_1 + \varphi\left(\ln\frac{v_1}{v_2}, \ln\frac{v_2}{v_3}, \ldots\right) \ . \quad (45)$$

Let $\varphi(u_1, u_2, \ldots) = \ln\phi(e^{\frac{1}{u_1}}, e^{\frac{1}{u_2}}, \ldots)$, then

$$\psi(\mathbf{v}) = v_1\phi\left(\frac{v_2}{v_1}, \frac{v_3}{v_2}, \ldots\right) \ . \quad (46)$$

According to (33), we have

$$h(\mathbf{w}) = w_1 + (w_2 - w_1)\phi\left(\frac{w_3 - w_2}{w_2 - w_1}, \frac{w_4 - w_3}{w_3 - w_2}, \ldots\right) . \quad (47)$$

Moreover, it is obvious that (47) satisfies both (23) and (35), hence (47) is a sufficient and necessary condition of (23) and (35), and the following theorem can be given:

Theorem 2 (Translation and scale invariant operator). *A continuously differentiable variation operator $h(x_{1d}, x_{2d}, \ldots)$ is translation and scale invariant if and only if it has the following form:*

$$h(x_{1d}, x_{2d}, \ldots) = x_{1d} + (x_{2d} - x_{1d})\phi\left(\frac{x_{3d} - x_{2d}}{x_{2d} - x_{1d}}, \frac{x_{4d} - x_{3d}}{x_{3d} - x_{2d}}, \ldots\right) , \quad (48)$$

where ϕ can be any continuously differentiable function.

Thirdly, a rotation invariant operator should satisfy

$$h\left(\sum_{i=1}^{D} m_{id}x_{1i}, \sum_{i=1}^{D} m_{id}x_{2i}, \ldots\right) = \sum_{i=1}^{D} m_{id} \cdot h(x_{1i}, x_{2i}, \ldots) \ , \quad (49)$$

where $m_{id} \in M$ and $d = 1, \ldots, D$. Let $\phi(\mathbf{w}) = \varphi(w_1, \prod_{i=1}^{2} w_i, \prod_{i=1}^{3} w_i, \ldots)$, then (47) is equivalent to

$$h(\mathbf{w}) = w_1 + (w_2 - w_1)\varphi\left(\frac{w_3 - w_2}{w_2 - w_1}, \frac{w_4 - w_3}{w_2 - w_1}, \ldots\right) . \quad (50)$$

According to (49), a translation, scale, and rotation invariant operator should satisfy

$$\begin{aligned} &\sum_{i=1}^{D} m_{id}x_{1i} + \left(\sum_{i=1}^{D} m_{id}x_{2i} - \sum_{i=1}^{D} m_{id}x_{1i}\right) \\ &\cdot\varphi\left(\frac{\sum_{i=1}^{D} m_{id}x_{3i} - \sum_{i=1}^{D} m_{id}x_{2i}}{\sum_{i=1}^{D} m_{id}x_{2i} - \sum_{i=1}^{D} m_{id}x_{1i}}, \ldots\right) \\ &= \sum_{i=1}^{D} m_{id}\left[x_{1i} + (x_{2i} - x_{1i})\varphi\left(\frac{x_{3i} - x_{2i}}{x_{2i} - x_{1i}}, \ldots\right)\right] , \end{aligned} \quad (51)$$

which is equivalent to

$$\begin{aligned} &\sum_{i=1}^{D} m_{id}(x_{2i} - x_{1i}) \cdot \varphi\left(\frac{\sum_{i=1}^{D} m_{id}(x_{3i} - x_{2i})}{\sum_{i=1}^{D} m_{id}(x_{2i} - x_{1i})}, \ldots\right) \\ &= \sum_{i=1}^{D} m_{id}(x_{2i} - x_{1i})\varphi\left(\frac{m_{id}(x_{3i} - x_{2i})}{m_{id}(x_{2i} - x_{1i})}, \ldots\right) . \end{aligned} \quad (52)$$

Let $\mathbf{u}_i = (u_{i1}, u_{i2}, \ldots) = (m_{id}(x_{2i} - x_{1i}), m_{id}(x_{3i} - x_{2i}), \ldots)$, we have

$$\sum_{i=1}^{D} u_{i1} \cdot \varphi\left(\frac{\sum_{i=1}^{D} u_{i2}}{\sum_{i=1}^{D} u_{i1}}, \ldots\right) = \sum_{i=1}^{D} u_{i1}\varphi\left(\frac{u_{i2}}{u_{i1}}, \ldots\right) , \quad (53)$$

②

空间变换不变性算子的一般形式

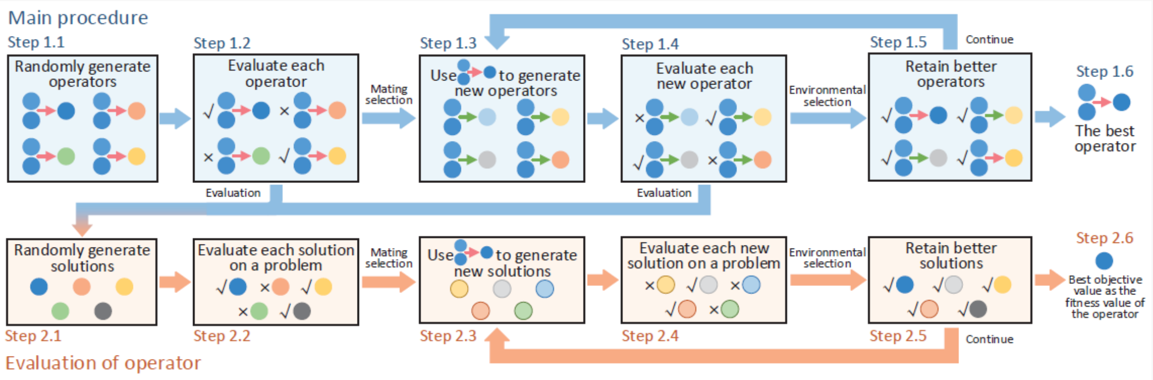
Theorem 3 (Translation, scale, and rotation invariant operator). *A continuously differentiable variation operator $h(x_{1d}, x_{2d}, \ldots)$ is translation, scale, and rotation invariant if and only if it has the following form:*

$$h(x_{1d}, x_{2d}, \ldots) = r_1x_{1d} + r_2x_{2d} + r_3x_{3d} + \cdots \ , \quad (61)$$

where $r_1, r_2, r_3, \ldots$ can be any real constants satisfying $r_1 + r_2 + r_3 + \cdots = 1$.

③

自动化算子参数搜索



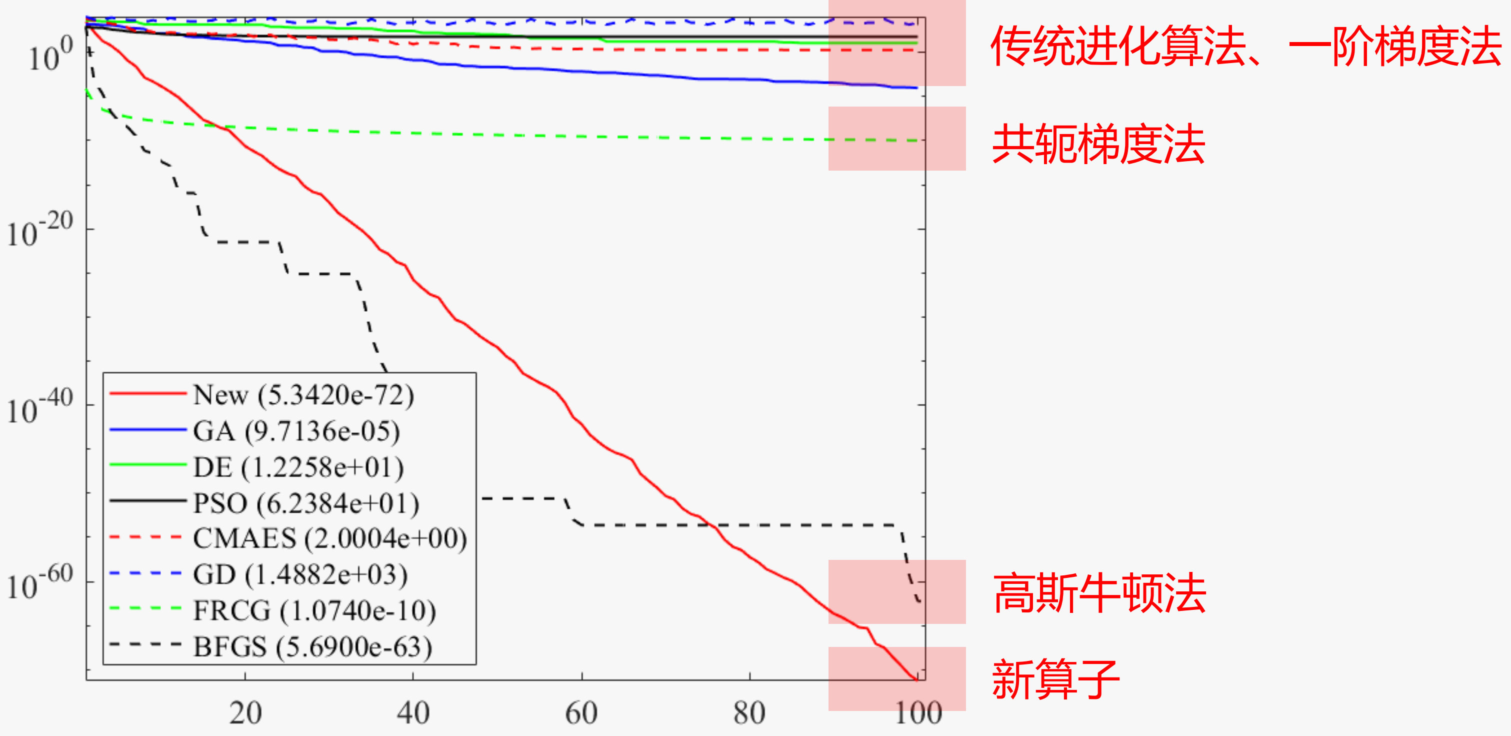
(11) Y. Tian, X. Zhang*, C. He, K. C. Tan, and Y. Jin, Principled design of translation, scale, and rotation invariant variation operators for metaheuristics, *Chinese Journal of Electronics*, 2022.

未来工作

- 新算子+遗传算法框架（**<u>纯黑盒</u>**），在**<u>单峰问题</u>**上较之**<u>二阶梯度法</u>**有性能优势

$$\min \ f(\mathbf{x}) = \sum_{i=1}^{20} x_i^4$$

$$\text{s.t.} \ -\frac{10}{i} \le x_i \le \frac{10}{i}$$



大规模多目标进化优化：算法与应用

安徽大学　田野

field910921@gmail.com

ECOLE 2022，郑州，2022年8月3日

下载本PPT、论文及代码

欢迎使用PlatEMO与GOpt

PlatEMO