> 早期算法

GA遗传算法：离散优化问题

Simple GA (SGA)

Representation	Binary representation
Recombination	One-point crossover
Mutation	Bit-wise mutation
Parent selection	Fitness proportional selection – implemented by Roulette Wheel
Survivor selection	Generational, i.e, age-based replacement with $\lambda = \mu$

第一件事：先选表示方式

用01串筛选特征，每一个待选对应一个位

ES演化策略：连续优化问题

> Evolutionary Strategies

Representation	Real-valued representation
Recombination	Discrete or arithmetic
Mutation	Gaussian perturbation
Parent selection	Uniform random
Survivor selection	Fitness-based replacement by (μ, λ) or $(\mu + \lambda)$
Speciality	Self-adaptation of mutation step sizes

- mutation：多元高斯分布

Discrete recombination: Same as for binary representation, e.g., m-point crossover and uniform crossover

Arithmetic recombination: Create offspring "between" parents

$$z_i = (1 - \alpha)x_i + \alpha y_i, \text{ where } \alpha \in [0,1]$$

适用于解决黑盒优化问题

EP：也是连续域上有穷优化

> Evolutionary Programming

- 每个个体确认产生一个子代
- 不进行recombination
- round-robin：每个与随机k个比较，选win最多的

Now typically applied to optimization in continuous domains, and almost merged with ES

Representation	Real-valued representation
Recombination	None
Mutation	Gaussian perturbation
Parent selection	Deterministic (each parent generates one offspring)
Survivor selection	Round-robin tournament
Speciality	Self-adaptation of mutation step sizes

difference

GP

> Genetic Programming

- 树表示
- 子树交换，树随机变换
- age-based

Representation	Tree representation
Recombination	Exchange of subtrees
Mutation	Random change in trees
Parent selection	Fitness proportional
Survivor selection	Generational replacement

- GP执行交叉或变异

生成树

确定初始树最大深度dmax

- Full method (each branch has depth = d_{max}):
  - nodes at depth < d_{max} are randomly chosen from F
  - nodes at depth d_{max} are randomly chosen from T
- Grow method (each branch has depth ≤ d_{max}):
  - nodes at depth < d_{max} are randomly chosen from $F \cup T$
  - nodes at depth d_{max} are randomly chosen from T

- 采样方式不同
- grow方法可能深度为0