The Origins

what is the name of the other four ”fathers” of Evolutionary Computing?

Alan Turing, John von Neumann, Darwin, Paul & work

Motivation

记住个体 基于 evolution of intelligence

启发式 problem solving

直接开发解决问题 分析问题时间减少 问题复杂度增加

ROBUST PROBLEM SOLVING technology 鲁棒性解决

灵感来自 nature， 大自然有答案

达尔文进化 适者生存

环境资源有限 选择不可避免 最有效争夺资源的获得繁衍机会

Phenotypic traits 表型：对环境的物理反应 fitness 由遗传和后天开发决定 个体独有

可继承 高了更容易繁衍

“If you have variation, heredity, and selection, then you must get evolution”

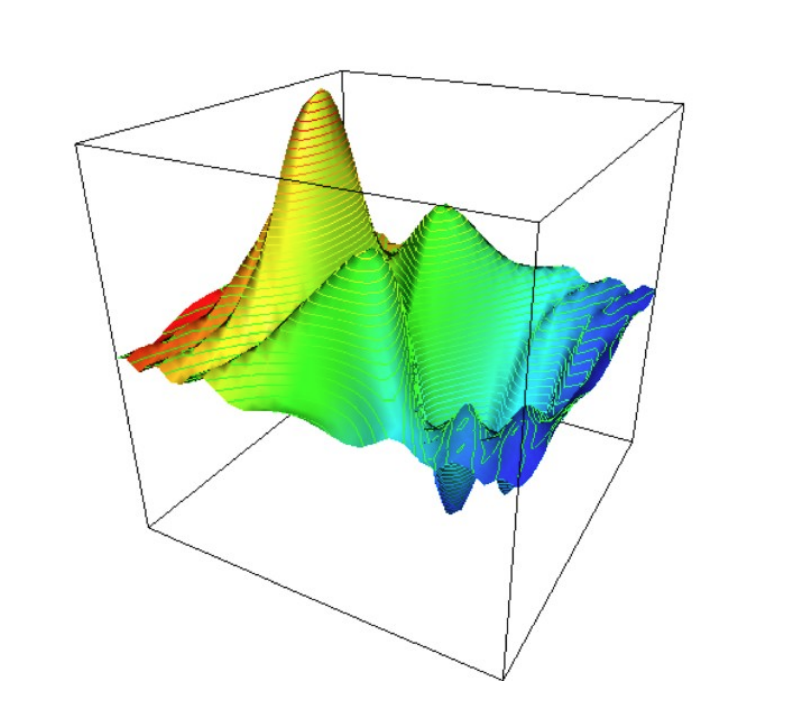
有变异 遗传 和选择 就有 进化

总结 种群由不同个体组成

个体是 “units of selection” 因为选择出来的遗传组合适应性更好

种群是 “unit of evolution” 因为 随机变异 导致 种群存在

Adaptive landscape metaphor 假设 适应度 存在于 3D中，高度对应fitness，XY对应基因，个体是空间中一个点，种群是点云



选择 推动 点云 向高处 移动

Genetic drift 基因漂变 蹦走了 离开当前位置 或 local optimal

Genetics

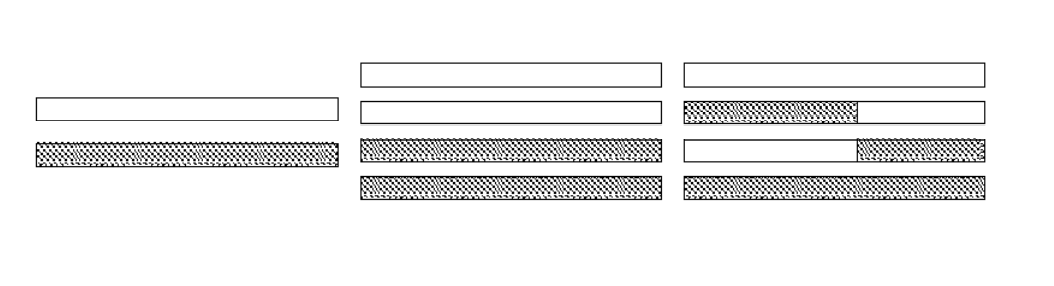
构建生物体 所需的 编码

Genotype 内 决定了 phenotype 外

Mapping genotype 到 phenotype很难 影响方式 1对多 多对1 都有

但是 Small changes in the genotype lead to small changes in the organism 小变 对 小变

Gene 在 DNA 编码为 染色体 双链 个体中全部的gene叫做 genome

23 对 染色体 一半一半的配对 部分cross over

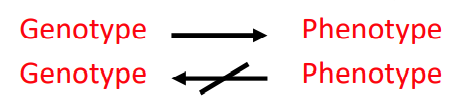
某些时候 一些gene会 mutate 不从父母任何一方 继承

大多时候 完蛋 小概率 没事儿 极小概率 牛逼了

尽管 所有cell基因一样 根据 ontogenesis 后天发育 也会不同

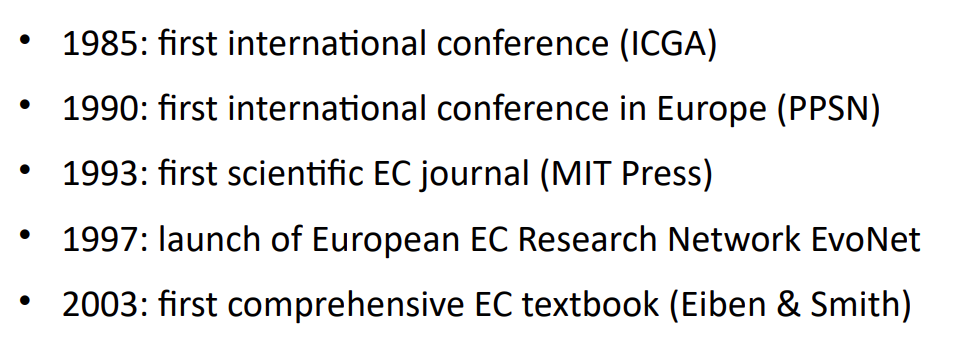
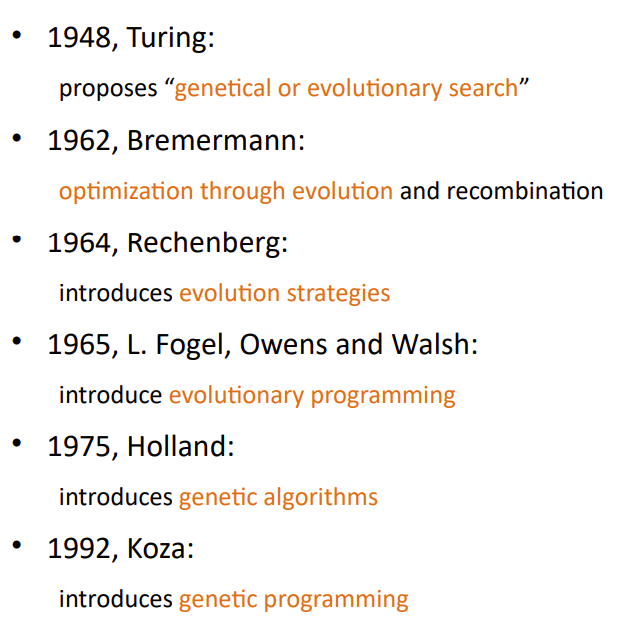
地球上 所有 东西 氨基酸 都一样 For all natural life on earth, the genetic code is the same

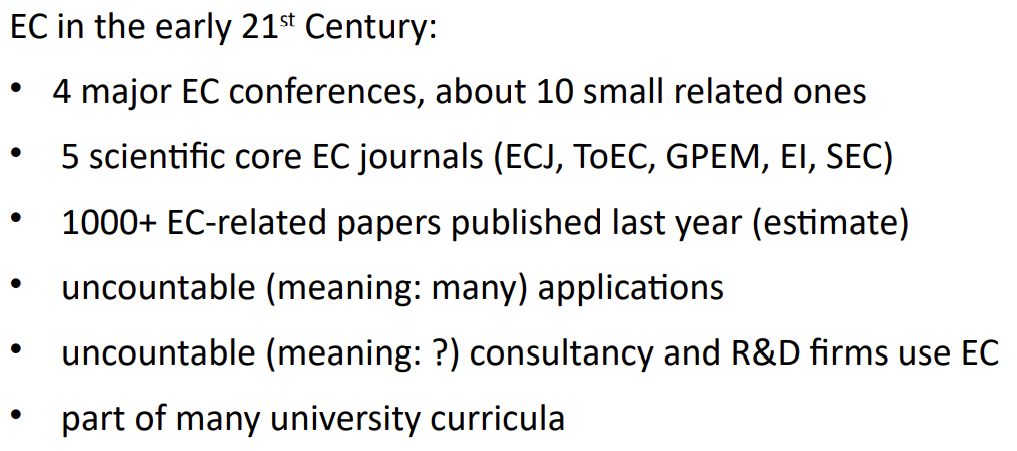
DNA transcription成 RNA translation成 Proteln



拉马克主义 认为的后天特性 可以遗传 事实上不对 But we can use it in artificial evolution

历史





重点

Several historical ‘dialects 几个历史时间点

principal algorithm: Evolutionary Algorithm 基础算法 EA

field: Evolutionary Computing (or Computation) EC

Not necessarily biological plausibility 非必要生物知识

进化过程 模型

Population of individuals 个体有 fitness

Reproduction / variation operators：mutation crossover

选择 higher fitness survival of the fittest mating of the fittest

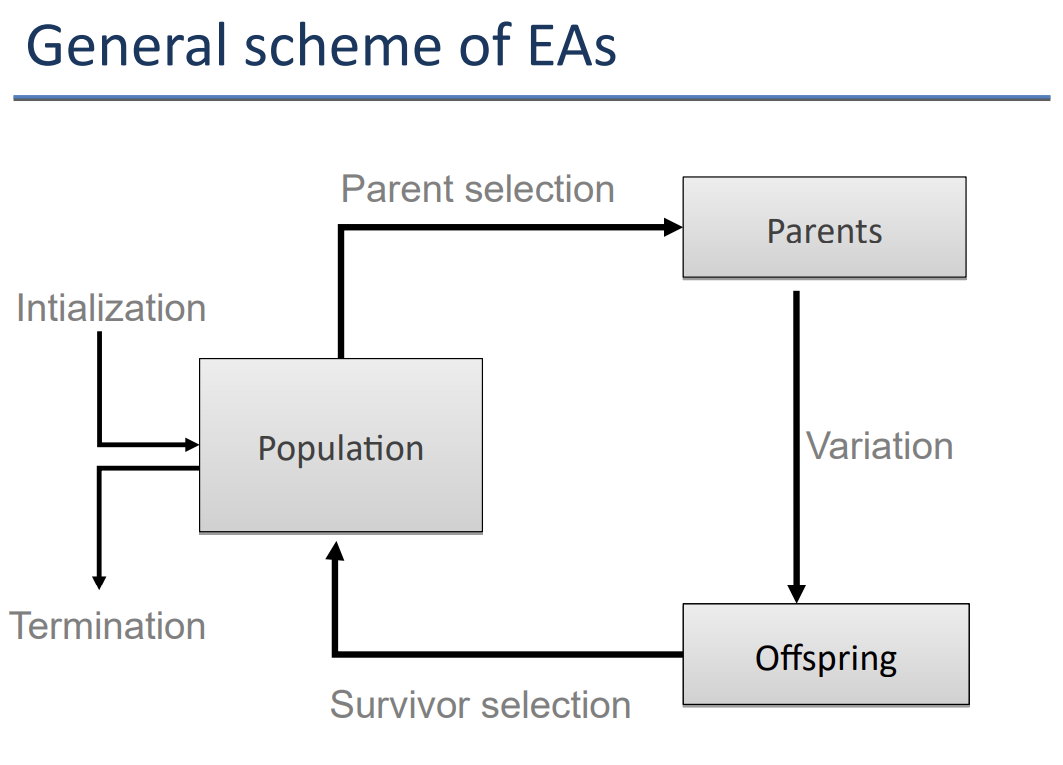
种群的fitness 随时间增加

两种竞争力

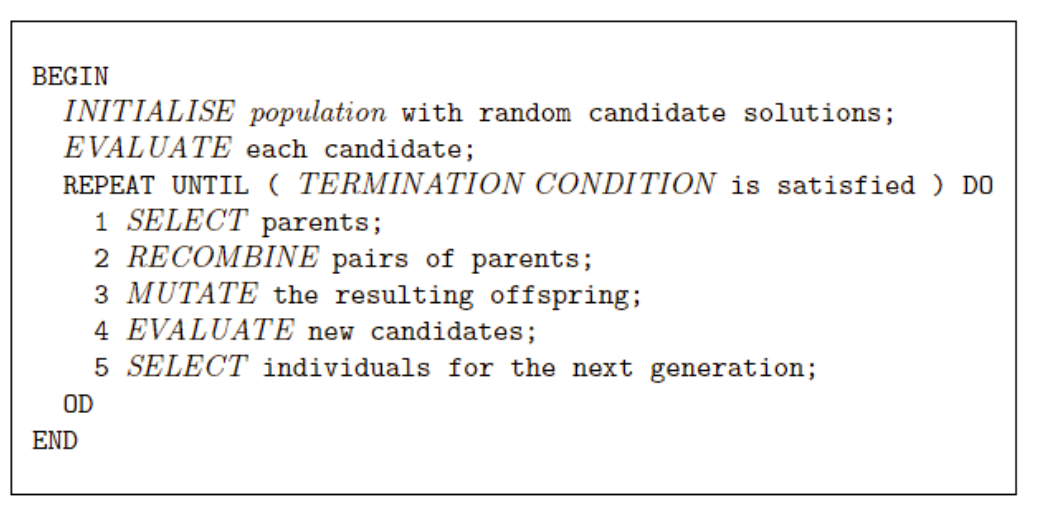
增加population 通过 变异 增加 diversity 更加 novelty

减少 population 通过 选择 增加 diversity 更加 quality

Scheme of EA



伪代码



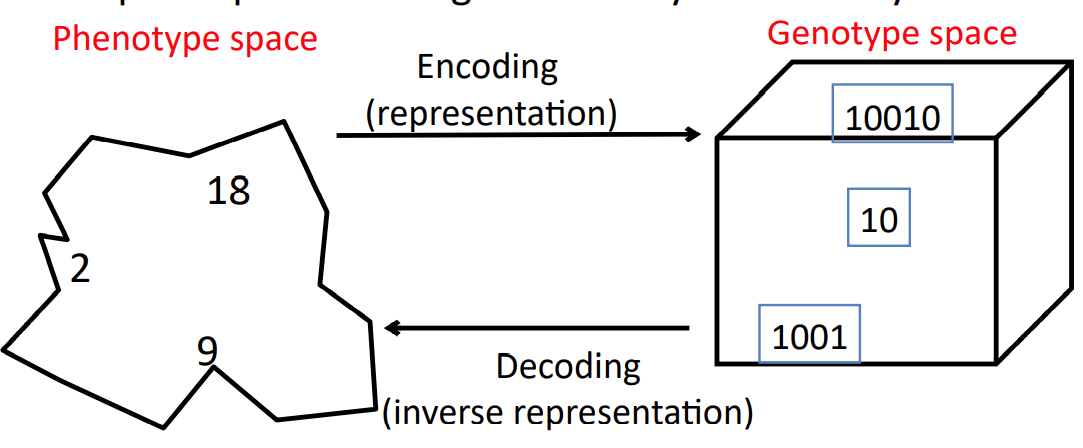
**Representation**

作用 提供由可操作运算符构成的解，可以变异

两种存在 phenotype 表型 对象 由初始上下文定义

Genotype 代码 表示 构成上述对象

2种映射 encoding 表型到基因 不一定 1v1

 decoding 基因到表型 1v1

**Evaluation / fitness function**

作用：表示要解决的任务 即要适应的环境 能够选择（可以比较）

Objective function 或 quality function

给每个表型分配一个值 用于 选择 一般 要 fitness最大（看情况）

**Population**

作用 hold 问题的候选解 作为 个体

种群是个体的多集，同一个体可能存在多个

Population is the basic unit of evolution, i.e., the population is evolving, not the individuals

进化的基础，在种群上选择，在个体上变异

复杂的EA 还有 人口结构的决定

种群的 diversity 指 不同的fitness value 数目

**Selection**

作用 识别个体 要么成为父代 要么活着

让种群通向higher fitness

通常是 概率 随机的

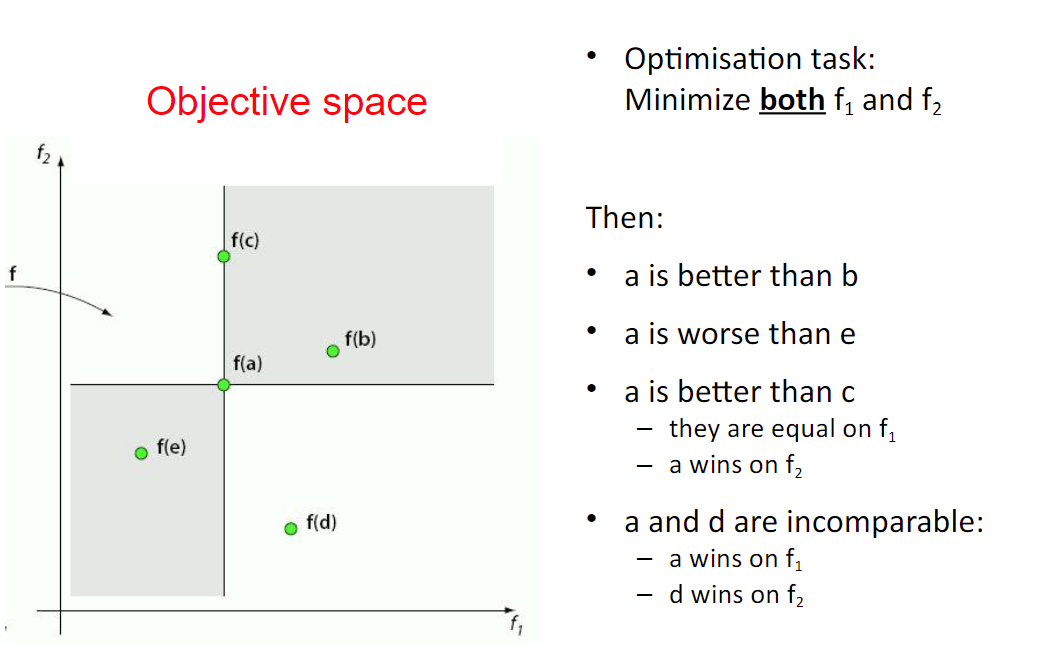
多目标优化

pareto优化，EC方法：进化域和多样性保持

一般问题会有N个冲突的目标 强度vs重量

两个目标：找到一组好解，找到解中的可应用best solution

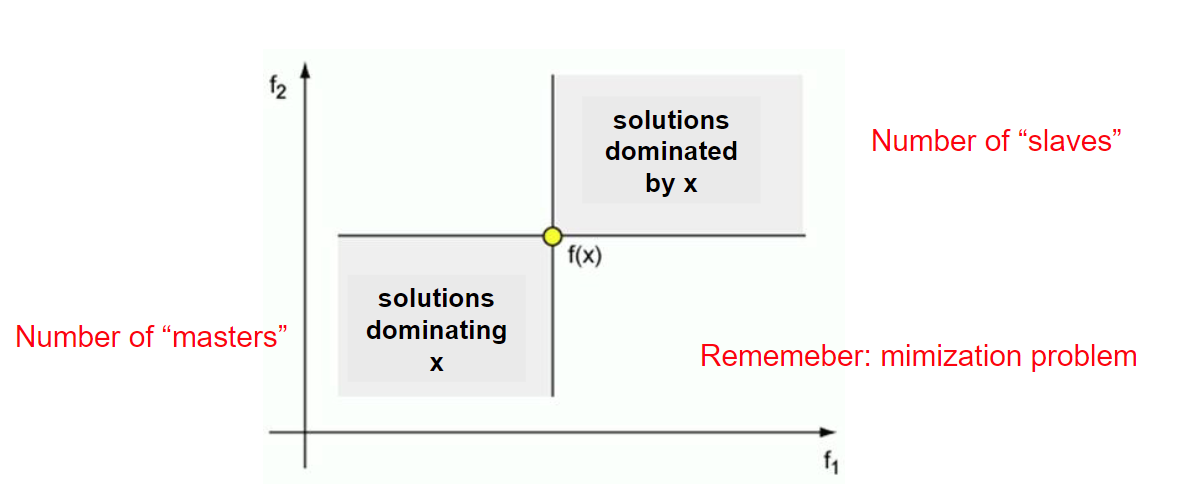
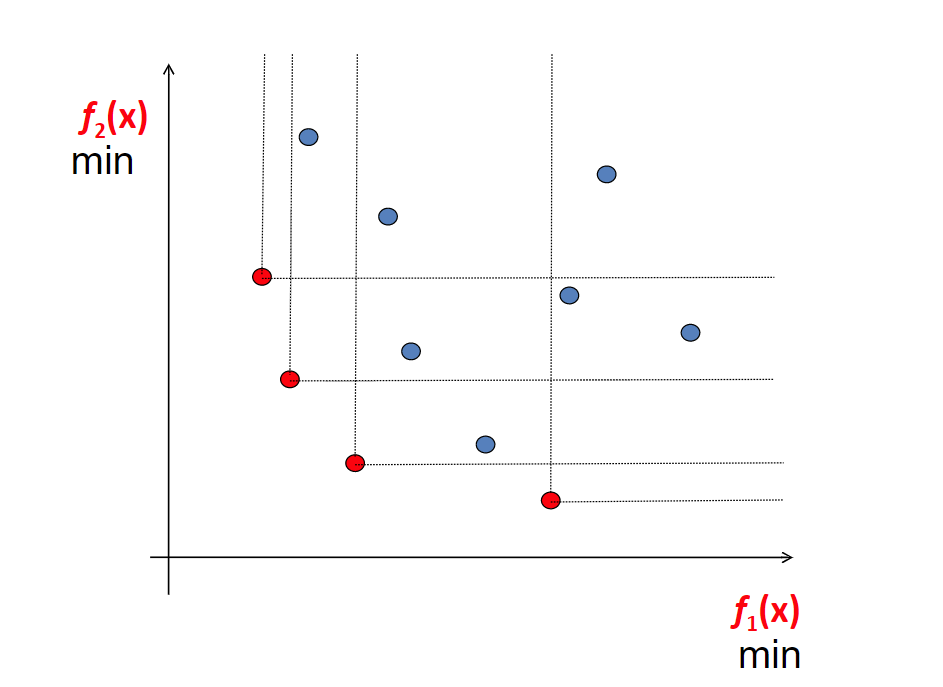
比较 solutions



决策域：产生解的地方 目标域：决策域中解映射过来的目标 （和单目标没区别）

Dominance relation 占优关系 说 x 占优 y

X better than y 在至少一个obj x 在所有 obj 上 都不比y差

Pareto 优化 在obj space

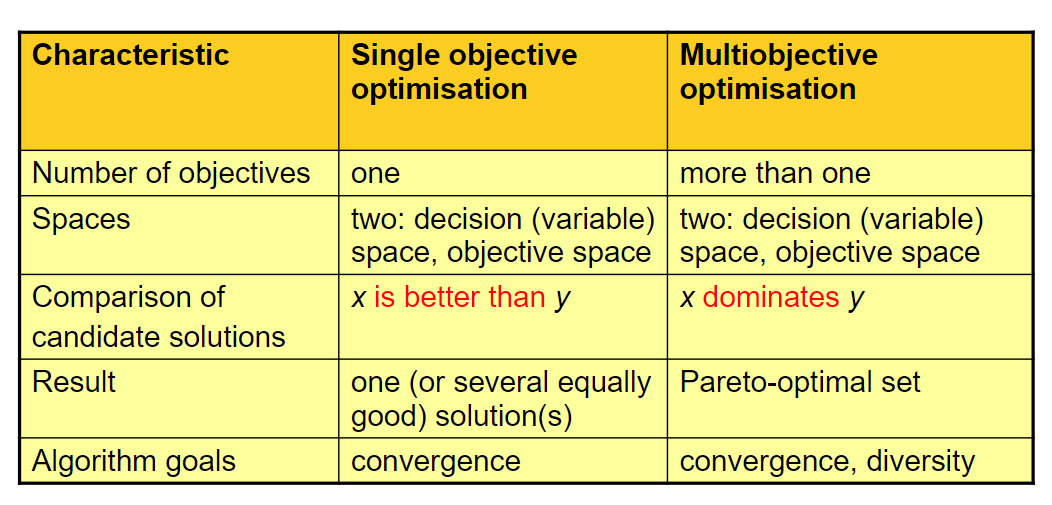
在一组solutions中，如果solution X在这组解中 是 non-dominated 则是 pareto solution

Pareto-optimal set：在solution space中的 一组 非占优solutions

Pareto-optimal front：在 obj space中， pareto set的 image

多目标优化器的目标 找 approximation set，

convergence 尽可能接近pareto front diversity 要分布



两种 多目标优化 办法：

Preference-based 又叫 聚合办法 和单目标一样，传统

将问题 转为 单目标，然后用原来的办法，转换过程使用更高级的信息：如偏好 来调整weights 是一种 单一 trade off的solution

Real multiobjective 使用 新的多目标技术，更好解决问题

Use a multiobjective optimisation method 多 trade off solution 但是在选择时候依然需要更高级的info

EC 办法：main rationale：因为EA是population based，他需要返回一组 trade off解，以上两种都行

优点：因为基于种群，EA可以 simultaneously 同时找一组 pareto 解

不用在搜索时候关心weights 不需要pareto front的相关info

EC requirements：1. Define fitness 通常基于占优来设定

用聚合办法调整weights 不同种群 不同标准

2. Maintain diversity 用小trick保持多样性

小技巧 fitness sharing 给fitness增加个值根据最近邻居 将space化成box并计算占用 所有的都依赖于 gene和表型的 distance metric

3. Remember all good points 用archive

用 elitist algorithm 精英算法 共同维护非占优点的记录

总结

Multi-objective problems occur very frequently

•EAs are very good in solving multi-objective problems

•MOEAs are one of the most successful EC subareas

•Number of objectives is typically low in practice and research, current State of the Art is about 2 or 3, more than 3 are seen as “many-objective” problems

约束处理 constraint handling

进化机器人

Entities with or without agency 的 区别贼大

Agent：Active, has agency, exhibits behavior “Living” matter

没有agent：Passive, has no agency, no behavior “Dead” matter

EA的亚种，更贴近自然：

个体是**active/autonomous entities**，基于sensor inputs 有自己的决策 并 反馈环境，有 behavior，能干点啥

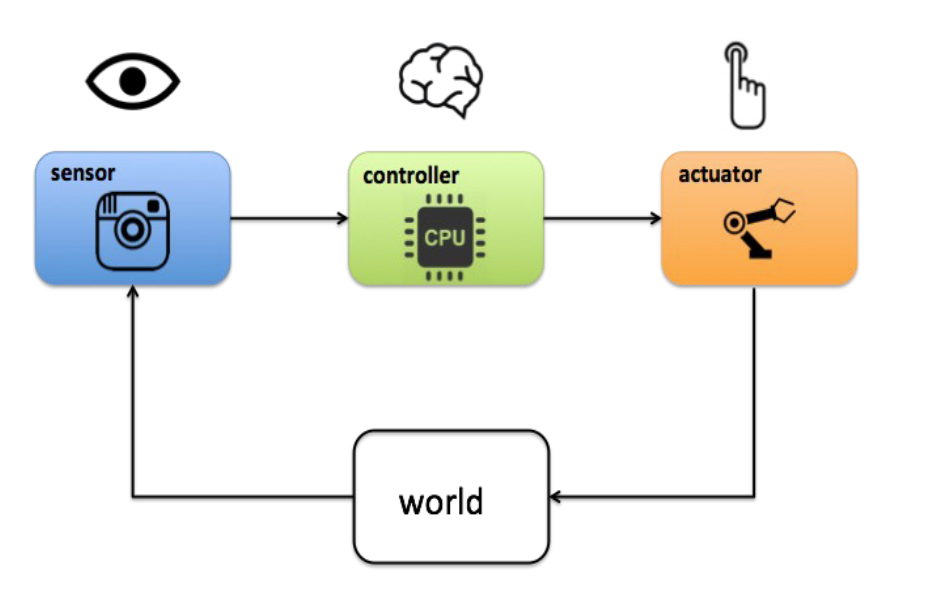
个体 在 **environment 中 operate**，不是只在fitness空间调优

**Evaluation 和 selection 是基于 agnet 做了什么，**不是他们是啥：

Fitness评估要在环境中模拟，time costs；且依靠很多 factors，增加不确定性；search space贼大；选择机制因为没有numeric评价，可以开放进化

繁衍可以异步 **Reproduction can be asynchronous**：

非异步：不用同时做all offsprings，population大小可变

 有这仨 就是 robot

Hardware – morphology – body Software – controller – brain Sensory

在环境 中 operate

给agent 写代码 hard 原因：

复杂 动态 噪音 的环境 复杂的 sensor mapping 复杂 task

**最关键的：user的影响是local的，目标行为却是 global**

Controller 许多种

NN 行为树rules C++ 机器码 都是控制器

problem description

Arena 舞台 如 space：empty obstacle等

Task 任务 避障 速度等

机器人的表型：固定结构和可调权重的NN控制器 基因型：权重vector

Fitnesss 表现的质量

在vector上操作 reproduction和selection，可以使用任何实数优化EA

在线进化 vs 线下进化

Offline 在部署前就跑好 controller提前安好固定在机器人上 都用这个95%

要注意 simulation evaluation和 reality的gap

Online 在执行阶段才跑 一边执行任务一边跑算法 只有文章

Neuro evolution 神经进化

AI: 自然 brain的简单抽象 多层

常见method supervised learning 事先know answers基于 答案 获得 反向传播

不知道 answers 怎么办？ 例如 control，有很多控制办法 但不知道 对不对

NE：不同于单纯基于可监督答案来调节 weights，还 evolve network种群

主要变量：

基于fixed topology只调节weights

也包括 网络的topology

还包括 网络的激活函数

常用于：control application

好处：通过 explicit diversity maintenance 防止 local optimal

Creative 且 有趣的 solution，不只是perfect solutions

可用于 supervised 和 非监督 微监督学习任务

坏处：黑盒 explain ability弱，computationally expensive 与其他学习办法比

简单的 neuro evolution 只包含weights

将NN学习问题变为vector 优化，固定topology，常为 全连接

高级的：让 topology也 evolvable，甚至包括 activation functions

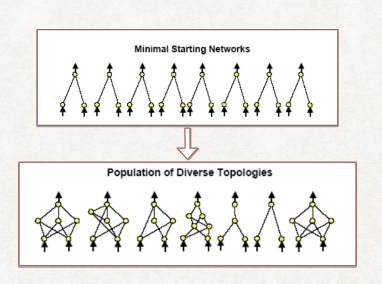
Topology and Weight Evolving Artificial Neural Network (TWEAN

NEAT: NEUROEVOLUTION OF AUGMENTING TOPOLOGIES

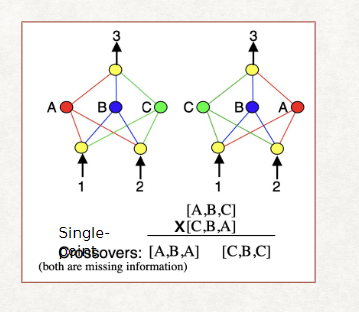
前提：过于big的network 优化成本 expensive

保护 innovation - 在 speciation 物种生成的时候，给 innovation一个机会 来展示他们的 potential

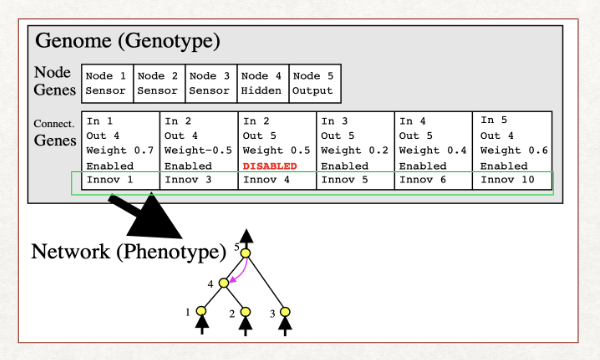
Minimality：种群从 最小 topology开始，根据繁衍增加复杂度，但复杂度也要maintains



Competing conventions problem 竞争规则问题：不止一种办法来表示NN权值优化问题

不知道哪个基因是哪个，不好crossover

那就使用 unique historical markers来标记基因



使用 innovation numbers，来align 2 genes 并 crossover

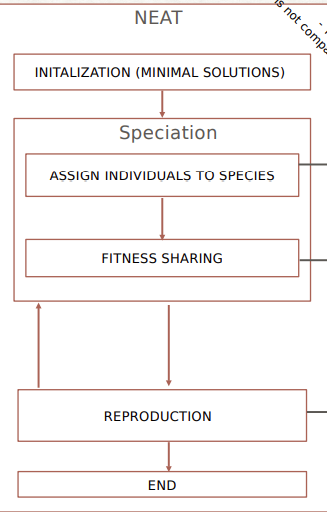
基因的选择是 random的，disjoint gene 在中间层不对付的 和 excess gene 在尾部不匹配的 直接从more fit parent 那里继承

Mutation: 不同的 突变会发生，每种都有其概率，在一次mutation中有多种types

Add node, remove node, add connection, remove connection, mutate weight….

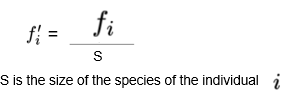
Speciation: protecting innovation and maintaining diversity 保持创新和多样性

Innovation是disruptive的，一个single extra node会让NN表现得完全不同

可能会让fitness 下降，但改了weights会更好，通过 having different species 保护 innovation，个体 只在 其 species中进行compete，在同一species中的个体是相似的

Maintain 一个 species list，每个species表示上一代的一个随机genotype，按顺序排放，

当前代的一个genotype和对应的representative species比较，评估measure是 compatibility distance，如果一个新的species和任何存在的都不一样，则另立门户，其作为representative

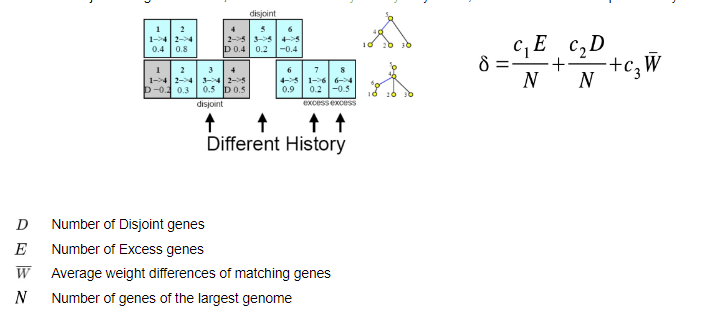


个体的fitness根据其species大小而改变，specie大了，fitness低，防止high fitness的species 太大。

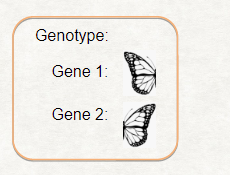
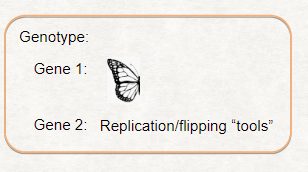
对每个 species：

丢了最差的，随机父母产生N个offspring，N和species的fitness成正比，fitness越高，N越大，子代替换父代。

计算 compatibility distance：两个基因中 不同的元素越多 则 进化史越不同 less compatible



Encoding type：直接编码 一个gene一个cell，结构冗余

间接编码 可重用性强 （描述都有啥的工具）

通常gene space很大，必须highly compressed representation才可以

包含 weights topology 和 activation function的neat：

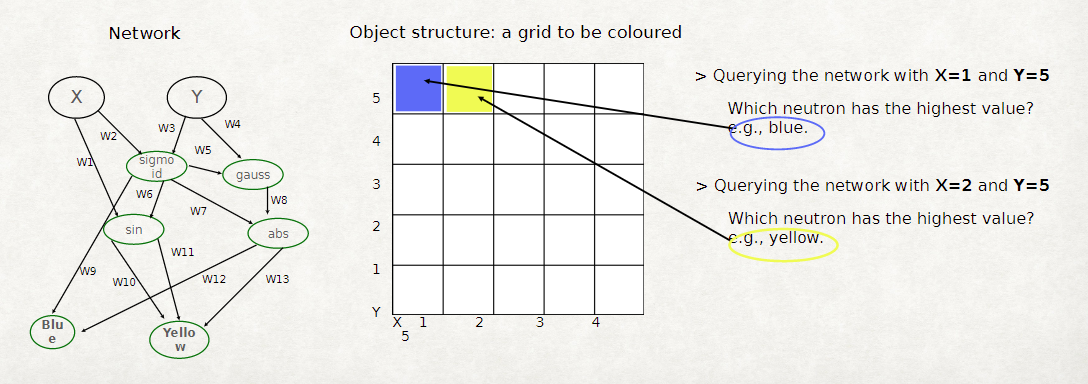
一个 developmental encoding：一个能生成要的的NN，优点smaller search space。

网络是根据问题geometry 和 context提出的function：CPPN-NEAT 和 NEAT一样，担保函激活函数 COMPOSITIONAL PATTERN PRODUCING NETWORKS （CPPN）

激活函数的 composition 成分 被 创建 with regularities

优点：object被赋予了 regularities

要 query一个network，需要提供依据问题产生的inputs 并 obtaining outputs

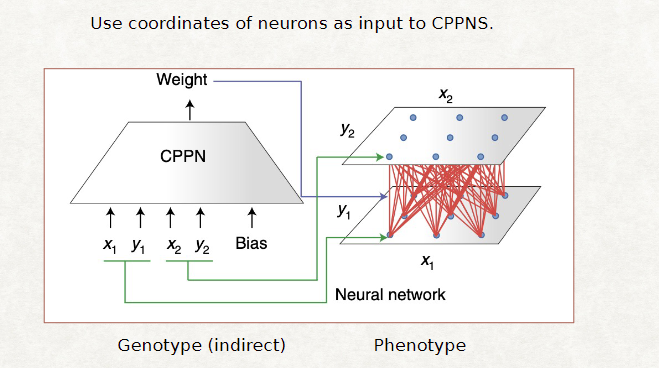


提供对应的XY值来 搜索 network，看哪个 neutron 有最高的fitness，则说明这个激活函数们好

CPPN 应用：画画，造型，音乐识别

HyperNeat 超神经进化！！！

Creating a neural network with a CPPN: making a neural network with another neural network.

用NN创造另一个NN 

因为 substrate resolution没有上界，所以 链接概念 有 无限的 resolution

总结

Neuroevolution can be done only on the weights, but also on topology and activation functions.

> Neuroevolution can be used for both supervised and (semi or un) supervised problems.

> NEAT can evolve networks fully, and its main principles are minimality and protection of innovation through speciation.

> A CPPN is a developmental representation that can encode regularities and can be used to generate “objects”, like drawings, shapes, other neural networks, et