



《前沿智能算法及应用》
课 程 论 文

题目：遗传算法综述

学 号	2212266	姓 名	张恒硕
专业班级	智能科学与技术 2 班	手 机	13940235405
出生日期	2004. 2. 11	提交日期	2024. 9. 9
成绩评定： 分 任课教师签名：			
2024 年 月 日			

遗传算法综述

摘要：遗传算法(Genetic Algorithms, GA)是模拟生物进化过程的一种算法，体现了交配、变异等遗传学行为，以“自然选择”原理寻找问题的全局最优解。本文从该算法的起源、发展、原理、实现、改进、应用等角度，对其进行了较为详细的描述，反应了该算法的内在原理与外延应用。

关键词：遗传算法 起源与发展 原理与实现 应用

一、遗传算法概述

遗传算法是进化计算的一个分支，为模拟自然界生物进化过程的随机搜索算法。其思想源于达尔文进化论的“自然选择”“物竞天择”和“优胜劣汰”的进化规律。通过模拟生物进化、自然选择中的复制(replication)、交叉(crossover)和变异(mutation)等遗传学行为，从任一初始种群出发，通过随机选择、交叉和变异操作，产生一代又一代更适合环境的个体，使群体繁衍进化，最后收敛到一群最适应环境的个体，从而求得问题的优质解。其最早由美国密歇根大学教授约翰·霍兰(John Holland)提出，现已广泛应用于各大领域的优化问题。

二、遗传算法的起源与发展

以下对截止到世纪初的关于遗传算法的研究进行了阶段性说明，具体可以参考早期的综述文章^[1]。

2.1 遗传算法的起源和早期发展(1960 年代—1970 年代)

1967 年，美国密歇根大学的计算机科学家约翰·霍兰和他的学生巴格利(Bagley)提出了遗传算法的概念。他在《适应性系统的理论》("Adaptation in Natural

and Artificial Systems")一书中描述了这一算法，并介绍了基于自然选择和遗传机制的优化方法。随后的几年中，霍兰及其学生开始研究遗传算法的实际应用，并对算法进行了初步的改进和实验。

2.2 遗传算法的推广和应用(1980 年代—1990 年代初)

1980 年代，遗传算法逐渐被推广到各种应用领域，研究者开始探索其在工程设计、经济学、控制系统等问题上的应用。大卫·戈德堡(David E. Goldberg)于 1985 年出版了经典著作《遗传算法与优化和机器学习》("Genetic Algorithms in Search, Optimization, and Machine Learning")，该书系统地介绍了遗传算法的基本原理和应用，成为遗传算法领域的重要参考文献。

1989 年，霍兰和他的学生们首次在遗传算法中引入了基因表达模型(Gene Expression Programming, GEP)，进一步扩展了遗传算法的应用范围。

2.3 遗传算法的成熟与多样化(1990 年代中期—2000 年代初)

1990 年代，遗传算法的应用范围扩展到更多领域，如机器学习、优化问题以及复杂系统建模等。

1997 年，首届 IEEE 国际遗传算法会议(GECCO)召开，为全球的遗传算法研究人员提供了重要的交流平台，促进了该领域的国际化发展。

1990 年代中期，多目标遗传算法(如 NSGA-II)被提出，用于解决涉及多个目标函数的优化问题，这些算法在处理复杂优化问题上表现出了优越性。

三、遗传算法的流程结构

3.1 遗传算法的原理

遗传算法是借鉴了生物遗传进化的原理而设计的，以下对相关的生物学原理进行解释，再将其与遗传算法的数学概念相对应。

3.1.1 遗传学概念与数学概念

1. 群体与种群：群体指在一定空间范围内，同时生活着的同种生物个体的总和。

这个群体内的个体之间可以相互交配，产生后代。种群则强调生物种群作为一个整体在自然界中的存在和动态变化，是生物进化的基本单位，具有特定的遗传结构。

- 个体(individuals)：搜索空间的一组可行解，可以用染色体表示。
- 种群(Population)：当前问题的可行解集合，即个体集合。

2. 染色体：细胞核内携带遗传信息的物质，由 DNA 和蛋白质组成。在细胞分裂时，染色体能够被精确地复制并分配到子细胞中，从而确保遗传信息的连续性和稳定性。基因型是组成染色体的一组基因的集合，在自然界中表征繁殖、突变。

- 基因型(Genotype)：可行解的编码串，其上每位代表一个基因。

3. 基因：染色体上线性排列的控制生物性状的基本遗传单位。每个基因都携带着特定的遗传信息，这些信息通过遗传密码来指导、控制蛋白质的合成进而影响生物性状。

- 基因(gene)：染色体的编码单元。

4. 适应能力：生物体在特定环境中生存和繁殖的能力。适应能力强的生物体更有可能在竞争中占据优势，从而将其遗传信息传递给后代。适应能力的强弱

取决于生物体的遗传组成以及它们与环境的相互作用。

- **适应度函数(Fitness function):** 染色体所表征的形状对应的适应值，也是问题的目标函数。适应度得分更高的个体代表更好的解，其更有可能被选择进行繁殖，使其性状在下一代中得到表现。随着遗传算法的进行，解的质量会提高，适应度会增加，直到找到令人满意的解。
5. **交配:** 生物繁殖过程中的一个重要环节，雌雄双方的生殖细胞结合形成受精卵，开始新生命，如下图 1 所示。交配不仅有助于遗传信息的传递，还能通过基因重组产生新的遗传变异，为生物进化提供原材料。

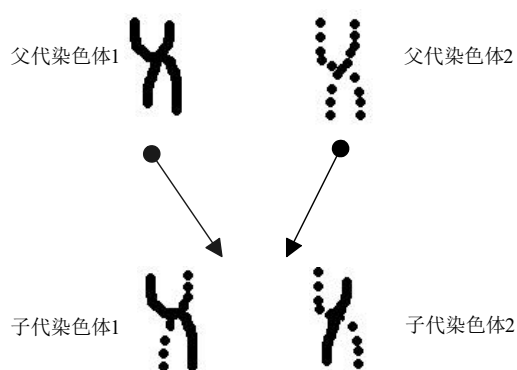


图 1 染色体交配生物学示意图

- **交叉/重组(Crossover) :** 染色体交换部分基因得到新染色体。在创造一对新个体时，通常将从当前代中选择的双亲样本的部分染色体交叉互换，以创建代表后代的两个新染色体。下图图 2 展示了单点交叉的操作。

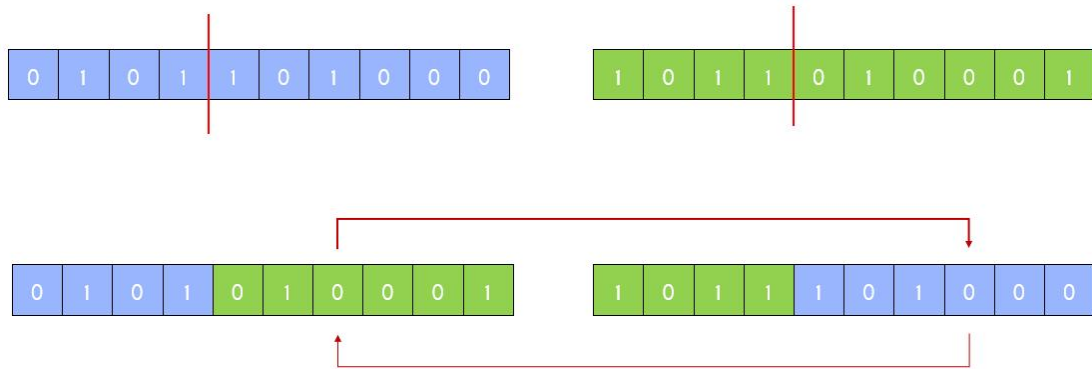


图 2 染色体交叉（单点交叉）

6. 变异：生物体在遗传组成上发生的可遗传的变化。变异是生物进化的原材料之一，它使得生物种群中出现不同的基因型和表型，从而增加生物体对环境的适应能力。变异可以自发产生，也可以由环境因素诱发。
- 突变(Mutation)：染色体某些基因的数值发生改变。突变操作可以定期随机更新种群，将新模式引入染色体，并鼓励在解空间的未知区域进行探索。下图图 3 展示了单点变异的操作。



图 3 染色体突变（单点变异）

7. 进化：生物种群在长时间内基因频率发生变化，使生物体更好地适应环境，是自然选择、遗传漂变、基因流和突变等多种机制共同作用的结果。进化是生物多样性的来源之一，也是生物界普遍存在的现象。
- 选择(Selection)：在新种群中选择能够继续繁衍、延续遗传信息的个体。在计算出种群中每个个体的适应度后，选择确定种群中用于繁殖产生下一代的

个体。具有较高适应值的个体更有可能被选中，并将遗传物质传递给下一代。低适应度值的个体仍然有机会被选择，但概率较低，可以保证不会完全摒弃其遗传物质。

3.1.2 自然选择与遗传算法

自然选择学说是英国生物学家达尔文在《物种起源》一书中提出的关于生物进化机理的重要学说，其基本内容如下：

1. 过度繁殖：地球上的生物普遍具有很强的繁殖能力，能够产生大量后代。这是自然选择发生的前提条件，因为大量后代中会出现各种遗传变异，为自然选择提供了丰富的材料。
 - 大量个体上的不同性状表征了其染色体上大量基因的不同，这说明遗传算法适用于多变量或变化范围大的优化问题。
2. 生存斗争：自然界中生物赖以生存的生活条件（食物、生存空间等）是有限的，因此生物之间、生物与无机环境之间必须进行生存斗争。这是自然选择的动力，只有适应环境的生物才能在斗争中生存下来并繁殖后代。
 - 这是目标函数评估个体适应度的依据。同时，这也启发了遗传算法的迭代思路：在种群的一次次演变中，适应度值高的个体更可能将其染色体上的遗传信息流传下来。
3. 遗传：生物个体能够保持亲本的遗传性状，使得生物种群中的某些特征能够稳定地传递下去。
 - 这为子代从亲代双方那里继承遗传信息提供了依据，在种群迭代的进化过程

中，提供了大量的确定因素。

4. 变异：生物个体在遗传过程中会发生变异，产生新的性状。这些变异可能是有利的，也可能是不利的。因为这种不确定性，变异后的个体可能无法比变异前的个体更加适应环境，这种方向的变异遗传下去的概率将变小。
 - 这为子代与亲代双方间可能产生的不同提供了依据。在种群迭代的进化过程中，提供了少量的不确定因素。
5. 适者生存：在生存斗争中，具有有利变异的个体能够生存下来并繁殖后代，而具有不利变异的个体则逐渐被淘汰。这体现了“物竞天择，适者生存”的原理，是自然选择学说的核心原理，解释了生物进化的方向和动力。
 - 这为适应度值与染色体基因排布的相关性提供了依据。
6. 物种的起源和进化：通过长期的自然选择过程，适应环境的生物种群会逐渐积累有利变异，形成新的物种。
 - 这为多次迭代最终获得优质解提供了依据。自然界中，生物的基因的变化性是不可用数字描述的，因此很难承认一个物种已经进化到了最优的阶段；但对于基因数量有限的问题情境来说，是可以找到最优解的。

3.2 遗传算法的流程结构

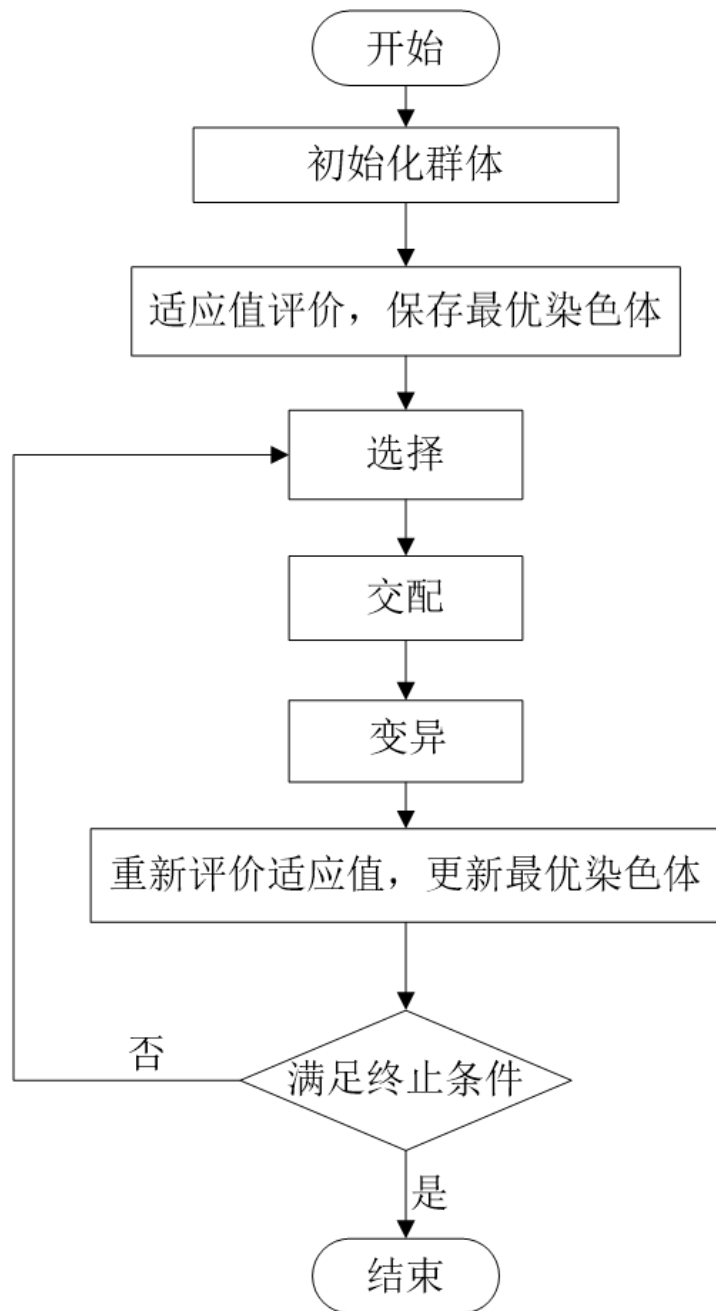


图 4 遗传算法流程图

- 对可行解空间进行编码，确定长度为 n 的染色体。
- 种群初始化：对染色体的 n 维进行随机初始化。
- 适应度函数：根据优化目标确定适应度函数，用来评估不同染色体表征的性状的优劣。规定适应值越大的染色体越优。

- 进化：在进化过程中，重复下述选择、交配、变异过程。
 - 选择：轮盘赌选择。在一个种群当中，一个个体被保留到下一代的可能性由其适应度值占比决定。
 - 交配：单点交叉。在染色体交配阶段，有两个随机参数，一个是交配概率 P_c ，一个是交配位置。前者决定染色体是否进行交配，后者则决定参与交配的两个染色体交换的位置。确定交配的两个染色体将交换交配位置以后的基因。
 - 变异：单点变异。染色体变异阶段的参数为变异概率 P_m ，其决定每个基因是否变异。确定变异的基因将向反方向变异。
- 进化终止：迭代进化，直到适应度函数达到阈值或者结果收敛。

四、遗传算法的实现与实践

本部分针对遗传算法进行了简单的实践，在 `python` 中对库中的遗传算法函数进行解析，进而形成简单的代码，并进行实际应用。

4.1 遗传算法的函数解析

`deap`(Distributed Evolutionary Algorithms in Python)，是进化算法框架，以下是调用其实现的简单的遗传算法的代码，其可对照 3.3 遗传算法的流程结构中所介绍的流程。

```
import random  
  
from deap import base, creator, tools, algorithms
```

```

# 设置遗传算法参数

creator.create("FitnessMax", base.Fitness, weights=(1.0,)) # 创建最大化适应度类

creator.create("Individual", list, fitness=creator.FitnessMax) # 创建个体类

toolbox = base.Toolbox()

toolbox.register("attr_bool", random.randint, 0, 1) # 二进制编码

toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_bool, n=10)

# 个体：10 个基因

toolbox.register("population", tools.initRepeat, list, toolbox.individual) # 种群

toolbox.register("evaluate", evalFunc) # 适应度函数：自定义的 evalFunc 函数

toolbox.register("mate", tools.cxTwoPoint) # 交配：单点交叉

toolbox.register("mutate", tools.mutFlipBit, indpb=0.05) # 变异：每个基因变异的概率为
0.05

toolbox.register("select", tools.selTournament, tournsize=3) # 选择：锦标赛选择，选取 3
个个体

# 应用遗传算法

population = toolbox.population(n=300) # 初始种群：300 个个体

NGEN = 40 # 进化次数

for gen in range(NGEN):

    offspring = algorithms.varAnd(population, toolbox, cxpb=0.5, mutpb=0.2) # 生成
后代，交配概率 0.5，变异概率 0.2

    fits = toolbox.map(toolbox.evaluate, offspring) # 计算适应度

```

```

for fit, ind in zip(fits, offspring):

    ind.fitness.values = fit

population = toolbox.select(offspring, k=len(population)) # 形成新种群

```

4.2 遗传算法的具体实现

以下是自己实现的遗传算法，题设背景为第 1 讲 遗传算法 ppt 中的作业题。

```

import numpy as np

n = 24 # 基因数目 (二进制编码长度)

population = 100 # 种群大小

Pc = 0.7 # 交配率

Pm = 0.07 # 变异率

generations = 200 # 迭代代数

X = [-3.0, 12.1] # x 取值范围

Y = [4.1, 5.8] # y 取值范围

# 二进制编码转换为实数域 x 和 y

def translateDNA(pop):

    x_pop = pop[:, 1::2]

    y_pop = pop[:, ::2]

    x = x_pop.dot(2 ** np.arange(n)[::-1]) / float(2 ** n - 1) * (X[1] - X[0]) + X[0]

    y = y_pop.dot(2 ** np.arange(n)[::-1]) / float(2 ** n - 1) * (Y[1] - Y[0]) + Y[0]

```

```
return x, y
```

```
# 种群适应度
```

```
def get_fitness(pop):
```

```
    x, y = translateDNA(pop)
```

```
    # 适应度函数
```

```
    pred = 21.5 + x * np.sin(4 * np.pi * x) + y * np.sin(20 * np.pi * y)
```

```
    return (pred - np.min(pred)) + 1e-3
```

```
# 交配、变异
```

```
def crossover_and_mutation(pop):
```

```
    new_pop = []
```

```
    for father in pop:
```

```
        child = father.copy()
```

```
        # 交配
```

```
        if np.random.rand() < Pc:
```

```
            mother = pop[np.random.randint(population)]
```

```
            cross_points = np.random.randint(low=0, high=n * 2)
```

```
            child[cross_points:] = mother[cross_points:]
```

```
        # 变异
```

```
        if np.random.rand() < Pm:
```

```
            mutate_point = np.random.randint(0, n * 2)
```

```

        child[mutate_point] = child[mutate_point] ^ 1

    new_pop.append(child)

    return new_pop

# 选择：轮盘赌

def select(pop, fitness):

    idx = np.random.choice(np.arange(population), size=population, replace=True,
p=fitness / fitness.sum())

    return pop[idx]

if __name__ == "__main__":

    pop = np.random.randint(2, size=(population, n * 2))

    for _ in range(generations):

        x, y = translateDNA(pop)

        pop = np.array(crossover_and_mutation(pop))

        fitness = get_fitness(pop)

        pop = select(pop, fitness)

    fitness = get_fitness(pop)

    max_fitness_index = np.argmax(fitness)

    print("max_fitness:", fitness[max_fitness_index])

    x, y = translateDNA(pop)

```

```
print("最优的基因型: ", pop[max_fitness_index])

print("(x, y):", (x[max_fitness_index], y[max_fitness_index]))
```

4.3 遗传算法的简单应用

上述代码的题设为：

$$f(x,y) = 21.5 + x \sin 4\pi x + y \sin 20\pi y$$

$$-3.0 \leq x \leq 12.1$$

$$4.1 \leq y \leq 5.8$$

求 $\max(f(x,y))$ 。

参数设定为种群数量 $N=100$ ，交叉概率 $pc=0.7$ ，变异概率 $pm=0.07$ ，最大迭代次数 $G_{max}=2000$ 。

结果如下：

- `max_fitness: 3.4181480338009913`
- 最优的基因型: [1 1 0 1 1 1 0 1 1 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 0 1 1 1 0 1 1 0 1 1
0 0 0 1 0 0 1 1 0 1 1 0 1 0]
- `(x, y): (11.624133337982496, 5.224069632534363)`
- `max(f(x,y))=38.33859034098016`

五、遗传算法的改进

遗传算法自提出以来，已经超过半个世纪，在这漫长的发展历程中，算法得到了各方面的改进。以下分别列举：

5.1 方法选择

5.1.1 编码方法

- 二进制编码：最常见的编码方式，使用 0、1 来构成个体基因型。简单方便，便于交叉变异操作，但不适合连续函数的优化问题，局部搜索能力差。
- 实数编码：基因值用某一范围内的一个浮点数来表示，个体的编码长度等于决策变量数。计算精确度高，适合解决高维、连续优化问题。
- 符号编码：使用符号来表示个体的基因型，符号的顺序不同表示不同的意义。
- 整数编码：使用整数作为编码符号，常用于序列排序问题、调度问题等。
- 排列编码（序列编码）：将有限集合内的元素进行排列作为个体的基因型，适用于旅行商问题等。

5.1.2 选择方法

- 轮盘赌选择(Roulette-wheel Selection)：又称比例选择方法，是基于适应度的随机选择方法，也是遗传算法最常用的选择方法。个体被选择的概率等于其适应度值与所有个体适应度值之和的比，适应度越高的个体被选中的概率越大。这种方法具有较强的随机性，既能保证较优个体有较大的被选中机会，也能给较差个体一定的生存空间，有助于维持种群的多样性。其公式如下：

$$p_i = \frac{f_i}{\sum_{j=0}^{\text{population}} f_j}$$

其中， p_i 表示第 i 个个体被选择的可能， f_i 表示第 i 个个体的适应度值。

- 锦标赛选择(Tournament Selection)：锦标赛选择法是一种确定性的选择方法。

在每次锦标赛中，从种群中随机选择一定数目的个体进行比较，适应度较高的个体作为获胜者。通过多场锦标赛可以构建出下一代种群。锦标赛选择能够较好地保留种群中的优秀个体，同时避免了对个体适应度的直接依赖，有助于算法在搜索空间中保持探索能力。

- **精英选择(Elitism Selection):** 在选择过程中，确保当前种群中适应度最高的(几个)个体能够直接复制到下一代中，这有助于防止最优解在进化过程中丢失。
- **随机遍历抽样(Stochastic Universal Sampling):** 根据每个个体的选择概率在 $[0,1]$ 区间内生成一系列等间距的点。通过比较这些点与个体的累积概率，来选择进入下一代的个体。这种方法比轮盘赌选择具有更低的选择误差。
- **玻尔兹曼选择(Boltzmann Selection):** 玻尔兹曼选择结合了轮盘赌选择和温度参数。在选择过程中，个体被选择的概率不仅与其适应度值有关，还受到一个逐渐降低的温度参数的影响。随着温度的降低，适应度较高的个体被选中的概率逐渐增加。

5.1.3 交配方法

- **单点交叉(Single-point Crossover):** 在两个配对的染色体中设置一个交叉点，在该点交换两个个体的部分染色体，生成两个新个体。该方法实现简单，容易保持个体的完整性。
- **多点交叉(Multi-point Crossover):** 在染色体中设置多个交叉点，交换这些交叉点之间的部分染色体，相比单点交叉能产生更多样化的后代，但也更容易破坏个体中的优良模式。

- **均匀交叉(Uniform Crossover):** 在两个配对的个体中，每个基因都以相同的交叉概率进行交换，从而形成两个新的个体。这种方法可以确保每个基因都有被交换的机会，增加了后代的多样性。
- **算术交叉(Arithmetic Crossover):** 在实数编码的遗传算法中，通过两个个体的线性组合来产生新的个体。
- **有序交叉(Ordered Crossover):** 有序交叉会保留父代个体的某些基因顺序，专门用于处理有序列表的编码方式，如旅行商问题中的城市顺序。
- **部分匹配交叉(Partially Matched Crossover, PMX):** 选择交叉点后交换点间基因，同时保持这些基因的相对顺序不变。
- **循环交叉(Cycle Crossover, CX):** 选择交叉点，通过交换交叉点上的基因，在两个父代个体间构建一个循环链，再根据循环链的顺序，将父代个体的基因依次填入子代个体的相应位置，直到所有基因都被使用。

5.1.4 变异方法

- **单点变异:** 随机选取染色体中某一位或几位基因进行变异操作，操作简单，但变异概率较低，可能需要较长时间才能得到更优解。
- **均匀变异:** 用某一范围内均匀分布的随机数，以较小概率替换染色体中各个基因的值。能对整个染色体进行随机变异，引入较大的随机性，适用于算法初期运行阶段，有助于增加群体的多样性。
- **非均匀变异:** 将染色体分为多段，采用不同的变异方式。随着进化的进行，变异方式会发生变化。

- 高斯变异：即将均匀变异的随机数改为以原值为中心的高斯分布。
- 边缘变异：即将均匀变异的替换值改选为范围的边界值。

5.2 参数设置

主要有以下可变参数：

- 群体规模：追求种群多样性和计算量的平衡。
- 染色体长度(基因数量)：根据编码方式改动。
- 基因取值：根据编码方式改动。
- 交配概率：取值一般为 0.4 至 0.99。
- 变异概率：取值一般为 0.001 至 0.1。
- 适应度函数：与问题的目标函数有关，值越大表示个体越优。
- 终止条件：迭代一定次数或收敛。

5.3 其他

- 混合遗传算法：与局部搜索算法（或贪婪算法、爬山法）相结合，获得全局搜索能力和局部搜索能力，可以显著提高算法的求解效率和质量。
- 并行遗传算法：利用多核处理器或分布式计算资源，同时运行多个遗传算法实例，以加快算法的收敛速度和提高求解效率。可以将种群划分为若干小生境，每个小生境内的个体进行独立进化，以增加种群的多样性。再在不同的小生境之间交换个体，以引入新的遗传信息。
- 自适应遗传算法：随着搜索进程推进，算法的交叉概率、变异概率等参数可

以动态调整。当陷入局部最优时，增大变异概率以跳出局部最优；当多样性较高时，减小变异概率以加快收敛速度。

六、遗传算法的进一步发展与应用

6.1 现代发展趋势

以下列举了遗传算法的一些发展趋势：

- **自适应机制：**针对遗传算法中交叉和变异概率的自适应问题，M. Srinivas 提出了相应的机制^[2]。此后，为了提高遗传算法的效率，研究者们不断探索自适应机制，以适应不同问题的需求。如黄铭等人提出了一种基于自适应边界约束(ABC)的高效遗传算法^[3]，其依据前一个系数值，自适应缩放后一个系数的边界，可在任意多的父代重组情形下快速生成系数向量。
- **多目标优化：**NSGA-II 算法作为一个快速精准的多目标优化遗传算法被不断研究^[4]。现在，发展到 NSGA-III 算法，其应用范围广泛，并不断得到改进^[5]。
- **混合遗传算法(Hybrid Genetic Algorithm, HGA)：**研究者们不断将遗传算法与其他算法相结合，形成混合算法，并以此为基础进行改进。
 - **粒子群算法(Particle Swarm Optimization, PSO)：**H Duan 等人将遗传算法与混合粒子群优化算法相结合，形成了用于解决多无人机编队问题的方法^[6]。
 - **模拟退火算法(Simulated Annealing, SA)：**汪开普等人就工作中的人机协作问题展开了研究，基于任务分配复杂、装配模式多、机器人资源受限等特征，提出了一种混合遗传模拟退火算法^[7]。

- 神经网络(Neural Networks, NNs): 卢琪等人就简单链型悬挂接触网结构优化问题, 提出了基于 BP 神经网络和遗传算法的结合方案^[8]。
- 局部搜索算法(Local Search Algorithms, LSA): 在 2009 年, 针对旅行商问题, Y Kim 等人提出了整合了局部搜索方法的遗传算法^[9]。而针对遗传算法求解度约束最小生成树存在的求解质量不稳定、局部搜索不完全的问题, 鞠成安等人提出了一种融合了局部搜索策略的改进稳态遗传算法^[10]。
- 与深度学习、强化学习结合: 人工智能方兴未艾的当下, 遗传算法与深度学习、强化学习等人工智能技术的结合成为了研究热点, 其可以用于解决复杂的优化问题。如 Such, Felipe Petroski 等人设计了一种由简单的、无梯度的、基于群体的遗传算法来评估深度人工神经网络(DNN)权重的方法^[11]。
- 多群体: 并行的遗传算法有多种群体模型, 如岛屿模型、踏脚石模型、邻居模型。针对多种群遗传算法, He W 等人^[12]和 Li Q J 等人^[13]做出了改进和应用。

6.2 实际应用

遗传算法在不断改进的过程中, 为适应不同的题设环境产生了大量有针对性的变种, 而各学科的研究人员也乐于采用遗传算法来解决其在自身领域中所遇到的问题, 这使得遗传算法得到了大量的应用。以下拣选几个重要领域进行说明。

- 工业与制造业: 遗传算法在工业生产优化、智能制造、供应链管理等方面得到了广泛应用。在车间调度^[14]、物资调配^[15]等方面, 都可以看见遗传算法的影子。

- 航天领域：在航天领域，遗传算法可以应用于航天器设计优化、轨道计算等方面。如马林等人便应用改进遗传算法优化了低轨星座部署策略^[16]。
- 地理学：在地理行业，遗传算法可以用于定位等问题^[17]。
- 经济学领域：在经济行业，遗传算法可以有很多作为，比如 Narh A T 等人便用其预测企业的破产趋势^[18]，而 Prasad K K 等人利用遗传算法，对缴费型养老金的合理安排进行了计算^[19]。
- 生物信息学与医学：遗传算法在基因序列分析、药物研发、疾病诊断^[20]、医学图像处理与分析^[21]等方面发挥了重要作用。

参考文献

以上未标注内容主要参考了相应的百度百科词条、第 1 讲 遗传算法 ppt 和 csdn 相关内容。

^[1] 席裕庚,柴天佑,恽为民.遗传算法综述[J].控制理论与应用,1996,(06):697-708.

^[2] Srinivas M ,Patnaik M L .Adaptive probabilities of crossover and mutation in genetic algorithms.[J].IEEE Trans. Systems, Man, and Cybernetics,1994,24(4):656-667.

^[3] 黄铭,王龙波,肖明虹,等.一种基于自适应边界约束的高效遗传算法[J].北京大学学报(自然科学版),2024,60(04):665-672.DOI:10.13209/j.0479-8023.2024.049.

^[4] Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182-197.

- [5] 毕晓君,王朝.一种基于参考点约束支配的 NSGA-III算法[J].控制与决策,2019,34(02):369-376.DOI:10.13195/j.kzyjc.2017.1067.
- [6] Duan H , Luo Q , Shi Y ,et al.?Hybrid Particle Swarm Optimization and Genetic Algorithm for Multi-UAV Formation Reconfiguration[J].IEEE Computational Intelligence Magazine, 2013, 8(3):16-27.DOI:10.1109/MCI.2013.2264577.
- [7] 汪开普,章卫,李新宇.基于遗传模拟退火算法的资源受限人机协作装配线平衡研究[J/OL].计算机集成制造系统,1-24[2024-09-02].<https://doi.org/10.13196/j.cims.2024.0192>.
- [8] 卢琪,苏凯新,张继旺,等.基于 BP 神经网络和遗传算法的简单链型悬挂接触网结构优化[J].机械工程学报,2024,60(12):313-320.
- [9] Kim Y , Cho S B .A Hybrid Cultural Algorithm with Local Search for Traveling Salesman Problem[C]//2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA).0[2024-09-02].DOI:10.1109/CIRA.2009.5423211.
- [10] 鞠成安,王妮娅,HANZALA,等.融合局部搜索策略求解 DCMST 的改进稳态遗传算法[J].重庆邮电大学学报(自然科学版),2023,35(05):935-942.
- [11] Such F P , Madhavan V , Conti E ,et al.Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning[J]. 2017.DOI:10.48550/arXiv.1712.06567.
- [12] He W ,Tang X ,Ji W , et al.An improved multi-island genetic algorithm and its utilization in the optimal design of a micropositioning stage[J].Expert Systems With Applications,2024,257125029-125029.

- [13] Li Q J ,Xia L X ,Sun C , et al.Estimation of time-dependent laser heat flux distribution based on BPNN improved by multiple population genetic algorithm[J].International Journal of Heat and Mass Transfer,2024,233125997-125997.
- [14] 游勇,李明伟,许佩,等.遗传算法在卷包车间提效分析的应用研究[J/OL].包装工程,1-10[2024-09-02].<http://kns.cnki.net/kcms/detail/50.1094.TB.20240612.1540.004.html>.
- [15] 刘明亮.优化遗传算法在天然气管网流量分配过程模拟计算中的应用[J].计量学报,2024,45(06):848-856.
- [16] 马林,刘玥,侯祥震,等.应用改进遗传算法的低轨星座部署策略优化方法[J].航天器工程,2024,33(02):9-15.
- [17] 肖晓春,丁振,丁鑫,等.基于 P 波到时拾取的分析法与遗传算法联合定位方法与应用[J].岩土力学,2024,45(07):2195-2207.DOI:10.16285/j.rsm.2023.1240.
- [18] Narh A T ,Nortey E ,Adzri P E , et al.Enhancing corporate bankruptcy prediction via a hybrid genetic algorithm and domain adaptation learning architecture[J].Expert Systems With Applications,2024,258125133-125133.
- [19] Prasad K K ,Ghosh A .A Genetic Algorithm Approach to Optimal Asset Allocation of Defined Contribution Pension Funds: Evidence From India’ s National Pension System[J].Compensation & Benefits Review,2024,56(4):195-210.
- [20] Kothala P L ,Guntur R S .TL-LFF Net: transfer learning based lighter, faster, and frozen network for the detection of multi-scale mixed intracranial hemorrhages through genetic optimization algorithm[J].International Journal of Machine Learning

and Cybernetics,2024,(prepublish):1-27.

^[21] 许良凤,林辉,罗珣,等.基于并行遗传算法的 Otsu 双阈值医学图像分割[J].工程图学学报,2011,32(02):88-92.