

目录	1
----	---

目录

第一部分 课程论文题目	1
-------------	---

第二部分 课程论文内容	1
-------------	---

1 知识表示与推理	1
1.1 知识与知识表示	1
1.2 确定性推理	2
1.3 不确定性推理	5
2 搜索与问题求解	9
2.1 搜索与问题求解	9
2.2 遗传算法及其改进方法在八皇后问题中的对比分析	9
2.2.1 遗传算法简介	9
2.2.2 用遗传算法解决八皇后问题	10
2.2.3 普通遗传算法解决八皇后问题	13

第一部分 课程论文题目

人工智能基本理论及应用研究综述

第二部分 课程论文内容

1 知识表示与推理

1.1 知识与知识表示

知识可以理解为事实和与事实相关的规则的集合。与之类似的是离散数学中的命题逻辑的概念，众所周知命题便是各种事实以及事实之间的推

断。和命题逻辑一眼样，人工智能领域中的知识也是为了反应事物或者说是事实之间的联系与规则。相较与命题逻辑，此处的知识要求更高一些，我们希望知识至少是相对正确的（相对正确性），希望知识可以用符号等显式的表示出来以便能更好的利用这些知识（可表示、可利用性），当然知识也有很多不确定性。在不同的先决条件或者不同的规则下，所得到的知识也大相径庭。例如生物领域中“橘生淮南则为橘生于淮北则为枳”，计算机科学中二进制 10 与十进制的 10 并不相同。

知识表示不仅在人工智能领域中出现，其还是认知科学领域的重要内容，在 AI 领域，我们通常希望将知识表示训练的像人一样智慧。

1. AI 知识表示：

表示方法 我们如何表示知识？

表示范围 某一表示方法的使用范围？

表示效果 某种表示方案的效果如何？

本身性质 知识表示本身的性质问题？

知识表示分为两个步骤，将我们生活中的信息流转化为程序可以识别的信息流，然后模仿人的感知、认知、推理等思维来解决一些人类遇到的费时费力的艰巨任务。

对于 AI 领域的知识表示，通常有以下方法，见表 1。

对于 AI 中各种知识表示方法的优劣，见表 2。

而另一块是关于确定性推理的概念，即依据我们现有的初始信息或依据，按某种确定的推理策略不断运用库中已有知识来逐步得到确定的结论的推理过程。而在 AI 领域，我们的推理是借助程序也叫推理机实现的。

确定性推理的特点见表 3:

1.2 确定性推理

常见的确定性推理方法有图搜索策略，盲目搜索和启发式搜索等，还有最新的像消解原理、规则演绎系统以及产生式系统。此处我选择启发式搜索 (Heuristic) 来介绍他的原理和应用。

表 1: 知识表示方法及应用.

知识表示方法	应用
一阶谓词逻辑表示法	知识表示与推理
产生式表示法	基于遗传算法的问题求解系统 图搜索求解模型
框架表示法	复杂知识的框架网络
语义网络表示法	机器翻译、问答系统、自然语言理解

启发式搜索方法是一种帮你不断试探出答案的方法，但它给出的答案是具有偶然性的，也就是 subject to chance。如在一个状态空间中，对一点的可能所在的每一个位置进行评估，先得到一个最可能的位置。再从这个位置出发持续进行下一轮的位置评估，这样的循环往复的搜索过程就被称为启发式搜索。而评估过程所用到的估价函数是启发式搜索核心，可以根据状态场景来选取合适的估价函数，常见的如式 1:

$$f(n) = g(n) + h(n) \tag{1}$$

天气被认为是影响航班路径规划的最主要因素，有 70% 的航班延误是由于危险天气^[1]。Li He 和 Anfei Zhao^[2]使用启发式算法将飞机在危险天气下，不同飞行海拔高度对飞机飞行转态的影响离散化进入一个多边网格模型，结果发现，借助该网格所获得的最小代价的飞行路径比传统的以一个恒定海拔高度飞行的飞行路径要更短。本章认为，当飞行过程中海拔发生变化，则规划的飞行路径也应适当的变化，详情见图 1。

这篇文章讨论，如果在决策点改变飞行高度，则从决策点到达下一个路径点的成本不是水平面上的欧几里德距离。因为高度变化被视为在一定爬升角度下的直线飞行，计算到达所选高度后的下一个路径点的代价公式为式 2

$$C_{\text{height}} = \sqrt{d_{\text{height}}^2 + h^2} \tag{2}$$

表 2: 各种知识表示方法的优劣.

知识表示方法	优势	劣势
一阶谓词逻辑表示法	自然精确严密易实现	不能表示不确定的知识 耦合度高, 效率低
产生式表示法	自然、模块性 有效性、清晰性	不能表达具有结构性的知识 效率不高
框架表示法	结构性、继承性、自然性	缺乏形式理论 适应能力不强
语义网络表示法	强调联系, 符合人类思维 描述明确简洁直观 结构化显性描述语义关系	不能保证推论的严格有效 不能处理结点太多的推理 不便表达判断性深层知识

表 3: 确定性推理特点.

确定的推理策略、确定的结论
事实（条件）和知识是构成推理的两个基本要素
以数理逻辑的有关理论、方法和技术为理论基础
机械化的、可在计算机上加以实现

d_{height} 是当前决策点与选定高度中的下一个路径点在水平面上的欧几里得距离, h 是海拔的变化。同时, 在网格系统中, 使用曼哈顿距离作为代价估计的启发式函数具有更好的计算速度和搜索效率^[3]。这样做可以确保路径搜索方向始终靠近目标, 简化搜索空间, 排除远离目标的点。得到的启发式搜索函数中 $h(n)$ 为式 3。

$$h(n) = |x_g - x_n| + |y_g - y_n|$$

(3)

n 表示当前网格点, g 表示目标点, x_g, x_n 等表示相应结点坐标。

则应用次启发式搜索的规划飞行路径的代价函数就为式 4, C_{height} 表示

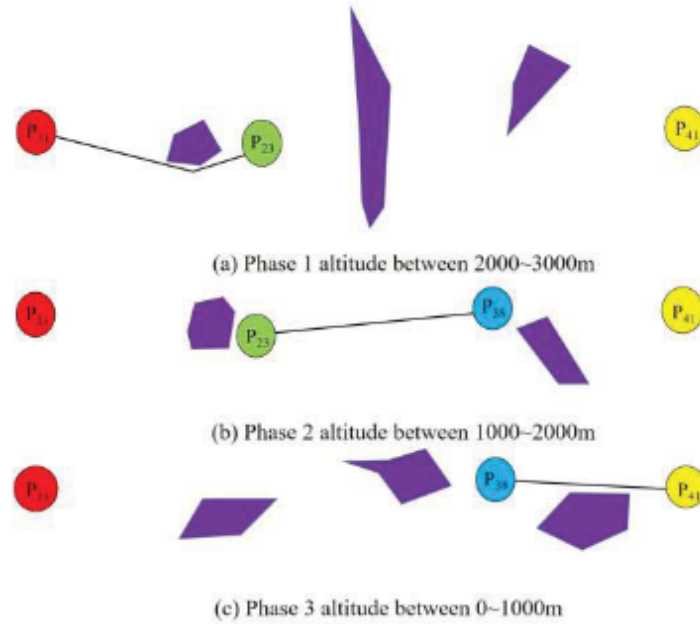


Fig. 3. Planned path after changing altitude

图 1: 海拔变化时路径节点变化图.

到达当前结点 n 之前的代价。

$$f(n) = C_{\text{height}} + |x_g - x_n| + |y_g - y_n| \quad (4)$$

最终的实验结果表明, 根据危险天气和海拔的不同, 基于启发式搜索的路径决策算法所得到的航班里程要小于恒定海拔下飞行的航班里程。通过海拔的不断调整, 可以将飞机更好的锁定在危险天气影响较小的飞行区域, 进而减少平均航行里程^[2]。

1.3 不确定性推理

对比于上文中所讲的确定性推理, 不确定性推理便好理解了。不确定性推理的初始的条件 (论据) 不确定, 推理方法 (或策略) 也不确定, 知识库也不确定, 最后的推理结果不确定但合理。常见的知识不完备、不精确以及模糊知识等的推理也属于不确定推理的范畴。不确定性推理的特点见表 4

常见的不确定性推理方法有模糊推理、可信度、证据理论、主观贝叶斯方法等等。本文主要选取主观贝叶斯方法来着重介绍一下。主观贝叶斯方

表 4: 不确定性推理特点.

证据的不确定性表示
不确定性的匹配
组合证据不确定性的计算
不确定性的更新
不确定性结论的合成

法隶属于规则推理方法，其以贝叶斯理论为依据，克服了实际问题中很难获取到先验概率的问题。^{[4] [5]} 我们都知道贝叶斯公式为式 5，而将全概率公式代入后可得贝叶斯公式的另一种形式，即式 6。

$$P(A_i | B) = \frac{P(A_i) \times P(B | A_i)}{P(B)}, i = 1, 2, \dots, n \quad (5)$$

$$P(A_i | B) = \frac{P(A_i) \times P(B | A_i)}{\sum_{j=1}^n P(A_j) \times P(B | A_j)} \quad (6)$$

引入产生式规则 IF E THEN H_i ，用该规则的前提条件 E 代替 Bayes 公式中的 B，而 A_i 则由 H_i 来代替可得到公式 7。

$$P(H_i | E) = \frac{P(E | H_i) P(H_i)}{\sum_{j=1}^n P(E | H_j) P(H_j)}, i = 1, 2, \dots, n \quad (7)$$

再引入一组不确定性数值对 (LS, LN) 来表示知识的强度，分别称为充分性度量和必要性度量，即式 8 和式 9。

$$LS = \frac{P(E | H)}{P(E | \neg H)} \quad (8)$$

$$LN = \frac{P(\neg E | H)}{P(\neg E | \neg H)} = \frac{1 - P(E | H)}{1 - P(E | \neg H)} \quad (9)$$

则此时得到主观贝叶斯公式为式 10。

$$\begin{aligned} P(H | E) &= \frac{P(H) \times P(E | H)}{P(E)} \\ P(\neg H | E) &= \frac{P(\neg H) \times P(E | \neg H)}{P(E)} \end{aligned} \quad (10)$$

借助几率函数即式 11 最终可得到分段线插值表示的后验概率为式 12。

$$\begin{aligned} O(x) &= \frac{P(x)}{1 - P(x)} = \frac{P(x)}{P(-x)} \\ P(x) &= \frac{O(x)}{1 + O(x)} \end{aligned} \quad (11)$$

$$P(x) = 0 \text{ 时, } O(x) = 0$$

$$P(x) = 1 \text{ 时, } O(x) = +\infty$$

$$P(H | S) = \begin{cases} P(H | \neg E) + \frac{P(H) - P(H|\neg E)}{P(E)} \times P(E | S), & 0 \leq P(E | S) < P(E) \\ P(H) + \frac{P(H|E) - P(H)}{1 - P(E)} \times [P(E | S) - P(E)], & P(E) \leq P(E | S) \leq 1 \end{cases} \quad (12)$$

其中 S 是对 E 的观察，而 $P(E|S)$ 表示在观察 S 下，E 发生的概率。

袁杰等^[6]借助改进后的主观贝叶斯方法来更好的进行电熔镁炉熔炼过程异常工况识别，同时利用模糊隶属度函数对个观测状态和证据进行匹配，对金属熔炼过程中的不确定性问题即异常工况获取到更好的识别效果。

这篇文章考虑到主观贝叶斯公式里 L_S 和 L_N 的取值较大，而选择将其值限制在 $[0, 1]$ 中，并设置 P_x 为证据 A 发生时对结论 B 的支持度。并选择合适的映射函数 f ，即 e 指数函数式 13。

$$1 - k_1 e^{-k_2 L_S} = p_x \quad (13)$$

从而得到新的 L_S 和 L_N 的概率公式 14 和 15。

$$L_S = \frac{\ln(1 - p_x)}{\ln(1 - P(B))} \quad (14)$$

$$L_N = \frac{\ln(1 - p_y)}{\ln(1 - P(B))} \quad (15)$$

并借助命题事件 $x \geq x_a$ 模糊隶属度函数图 2 来获得某证据 A 在观察 S 情况下匹配程度 $P(A | S)$ ，即特征变量观测值 S 特征变量限定范围 A 的匹配关系。改进了传统的主观贝叶斯方法中证据与结论的匹配非 0 即 1（即只有匹配或不匹配的结果）。也提高了异常工况识别中的识别准确率。

从现场收集并选取了 200 个异常工况样本，来制图显示电熔镁炉异常运行数据图，见图 3。可以看到，使用改进的主观贝叶斯方法对电熔镁炉

$$P(x \geq x_a | x) = \begin{cases} 1 - \exp(k \cdot (x_0 - x)), & x > x_0; \\ P(x \geq x_a), & x = x_0; \\ \exp(k \cdot (x - x_0)) - 1, & x < x_0. \end{cases}$$

图 2: 命题模糊隶属度函数.

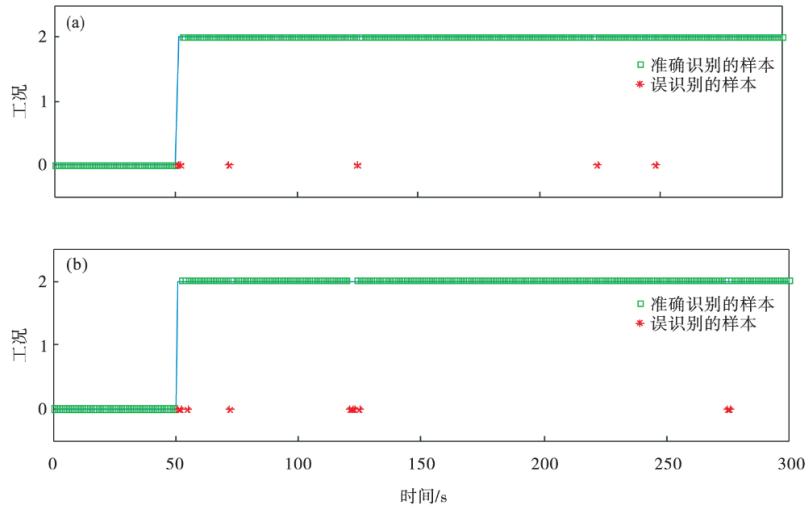


图 3: 两种贝叶斯方法异常工况识别结果.

(a) 一改进主观贝叶斯方法; (b) 一传统主观贝叶斯方法

熔炼工况进行识别后，相较于传统的主观贝叶斯方法，误识别样本减少了40%，说个改进的主观贝叶斯方法能对异常工况识别更快且提高诊断准确率^[6]。

可以发现主观贝叶斯方法的优点是理论基础比较坚实（计算公式多是由概率论的基础所推导出来），且 L_S 和 L_N 由领域内的专家给出，避免了我们自己的数据统计工作。再加上很多时候 LS 和 LN 也反映了证据与结论间的因果关系，也使得结论更为可靠。缺点是应用时，相关领域专家需要同时给出先验概率 $P(H)$ ，又由于 Bayes 的独立性假设等等都很难满足，使得该方法使用受限。这也间接看出该方法主要应用在领域专家能给出 L_S 和 L_N 以及事件满足独立性假设的情况，常见的如气象预测、人口预测等。

2 搜索与问题求解

2.1 搜索与问题求解

无信息搜索指的是已知条件很少，只有自己定义的信息可用，所以也称盲目式搜索。而有信息搜索就是找到一种策略来判断某一种搜索到的状态比另一种更好，进而能朝向更优的方向不断搜索。而超越经典搜索（局部搜索）则不太关心搜搜过程中的路径和代价，只关注解的状态。与之前的无信息搜索是在可观察、确定性以及已知的环境之下搜索不同，局部搜索跳出这些约束，不关心路径，从单个结点出发，通常只移动到该结点的邻近状态，如果存在一个最优解，最优局部搜索状态通常能找到最优解对应的最大或者最小值。

2.2 遗传算法及其改进方法在八皇后问题中的对比分析

2.2.1 遗传算法简介

遗传算法是借鉴了达尔文生物进化学说中种群遗传、基因突变、自然选择和杂交等理论的一种进化搜索算法。

1. 遗传算法的具体实现步骤：

生成随机候选群、初始化群基因个数。

设计适应度函数计算 `fitness_value`。

定义突变函数和交叉函数。

定义进化策略，包括丢弃和增补函数。

不断生成新种群，知道达到适应度要求的值。

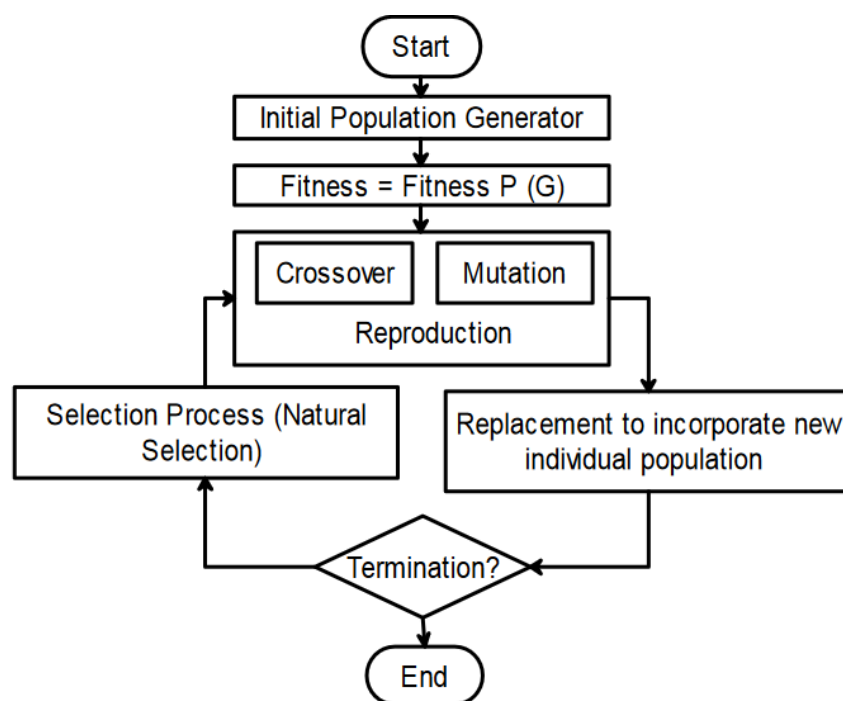


图 4: 简单遗传算法的步骤^[7].

2.2.2 用遗传算法解决八皇后问题

我们这次要解决的问题是八皇后问题，用八位数字来代表每个皇后，八个皇后组成遗传种群，并设置遗传种群的大小为 100。用不相互攻击的皇后对的数目来代表 `fitness_value`（即适应度值）。而如果检查次数（检查当前皇后位置是否与其他所有行列的皇后冲突）达到 28，则代表已找到八皇后的解。而是否选择该种群的某些个体进行下一轮的遗传则取决于个体的适应度值，适应度值越大，被选中的概率越高。简单遗传算法的 Python 实现见列表1，语言环境 Python 3.8.13 64-bit。

Listing 1: 简单遗传算法的 Python 实现^[8].

```
1 import numpy as np
2 import copy
3
4
5 # 普通选择算子遗传算法：适应度函数
6 def fitness_function(individual):
7     value = 0
8     for i in range(7):
9         for j in range(i+1, 8, 1):
10             if individual[i] != individual[j]:
11                 x_distance = np.abs(individual[j] - individual[i])
12                 y_distance = j - i
13                 if x_distance != y_distance:
14                     value += 1
15     return value
16
17
18 # 普通选择算子遗传算法：把适应度函数转化为概率分布
19 def softmax(input):
20     input = np.array(input, dtype=np.float)
21     input = np.exp(input)
22     output = input / input.sum()
23     return output
24
25
26 # 随机变异
27 def mutation(individual, prob=0.1):
28     p = np.random.rand(8)
29     individual[p > prob] = np.random.choice(range(8), 8)[p > prob]
30
31     return individual
32
33
34 # Genetic Algorithm
35 def GA(size=4):
36     # 默认种群大小为4，可以多试几个，效果不一样
37     size = size
38     num_generation = 0
39     population = []
40     Generation_Fitness = {}
41     for i in range(size):
```

```
42     population.append(np.random.choice( range(8), 8))
43 while (True):
44     fitness_list = []
45     selection = []
46     print("Generation : ", num_generation)
47
48     for individual in population:
49         fitness_value = fitness_function(individual)
50         if fitness_value == 28:
51             print("Find Target!")
52             print(individual)
53             Generation_Fitness[num_generation] = 28
54             return Generation_Fitness, individual
55         fitness_list.append(fitness_value)
56     Generation_Fitness[num_generation] = max(fitness_list)
57     print("current max fitnessvalue:" + str( max(fitness_list)))
58
59     # Selection is Here自然选择在这里
60     prob = softmax(fitness_list)
61     select_id = np.random.choice( range(size), size, replace=True, p=prob)
62     for idx in select_id:
63         selection.append(population[idx])
64     num_pair = int(size/2)
65     position = np.random.choice( range(1, 7, 1), num_pair, replace=True)
66
67     # Crossover is Here基因片段的交叉互换在这里
68     for i in range(0, size, 2):
69         start = position[ int(i/2)]
70         tempa = copy.deepcopy(selection[i][start:])
71         tempb = copy.deepcopy(selection[i+1][start:])
72         selection[i][start:] = tempb
73         selection[i+1][start:] = tempa
74
75     # Mutation is Here变异在这里
76     for i in range(size):
77         selection[i] = copy.deepcopy(mutation(selection[i], prob=0.8))
78     population = selection
79     num_generation += 1
```

2.2.3 普通遗传算法解决八皇后问题

根据种群中每个个体的 fitness_value 在当前种群中总 fitness_value 中的占比（即计算每个个体适应度值的概率分布）来选择一批用于繁衍下一代种群的新个体。实验软件为 Visual Studio Code. Version 1.56.2 测试平台：jupyter notebook，语言环境 Python 3.8.13 64-bit。采用这种策略的实验过程和结果见列表2，图 5和图 6。

Listing 2: 简单遗传算法解决八皇后问题的测试过程和结果.

```
1  # 导入测试所用到的库
2  import numpy as np
3  import copy
4  import matplotlib.pyplot as plt
5  import seaborn as sns
6
7  # 导入我们需要用到的遗传相关函数
8  from Genetic_Algorithm_N_Queen import fitness_function
9  from Genetic_Algorithm_N_Queen import softmax, mutation
10 from Genetic_Algorithm_N_Queen import GA
11
12 # 调用种群数为100的遗传函数GA
13 GeneticDict, Queen = GA(size = 100)
14
15 # 测试结果
16 -----
17 Generation : 0
18 current max fitnessvalue: 24
19 Generation : 1
20 current max fitnessvalue: 25
21 Generation : 2
22 current max fitnessvalue: 26
23 Generation : 3
24 current max fitnessvalue: 25
25 Generation : 4
26 current max fitnessvalue: 24
27 Generation : 5
28 current max fitnessvalue: 26
29
30 --中间遗传过程省略--
31
32 Generation : 1769
```

```
33 | current max fitnessvalue: 25
34 | Generation : 1770
35 | Find Target!
36 | [6 4 2 0 5 7 1 3]
37 |
38 | 总遗传代数: 1770代
39 | 计算总耗时: 15.8s
```

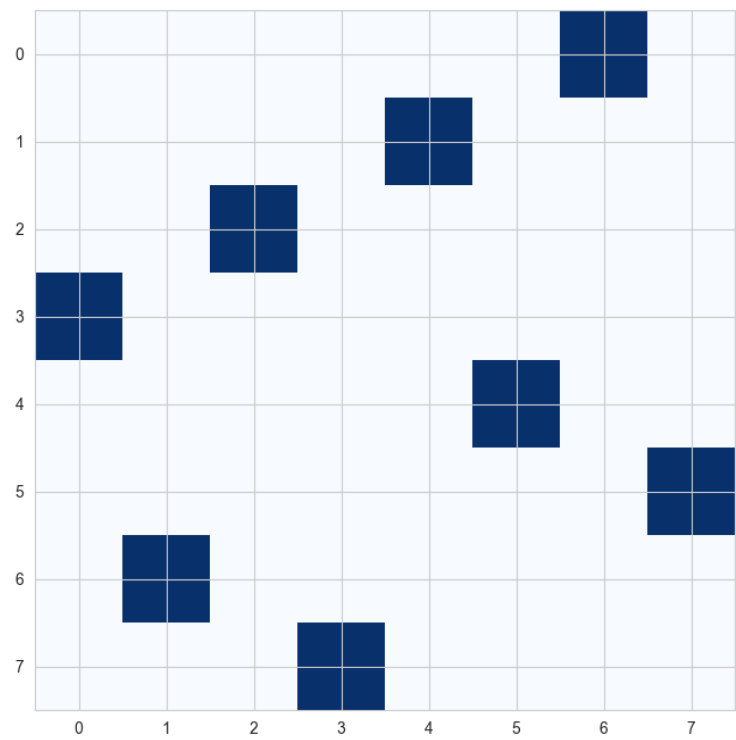


图 5: 简单遗传算法的解 8-Queen 图.

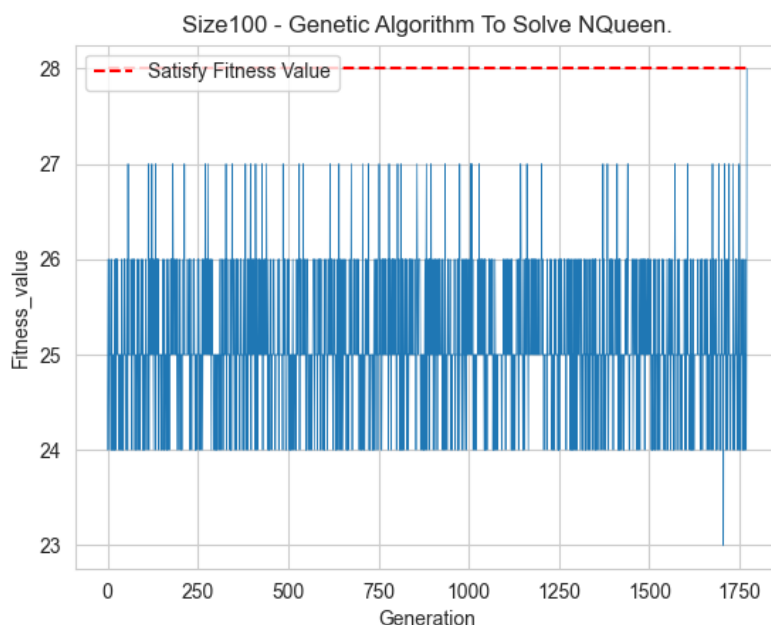


图 6: 简单遗传算法的解收敛过程.

参考文献

- [1] NARKAWICZ A, HAGEN G E. Algorithms for Collision Detection Between a Point and a Moving Polygon, with Applications to Aircraft Weather Avoidance[C/OL]//16th AIAA Aviation Technology, Integration, and Operations Conference. Washington, D.C.: American Institute of Aeronautics and Astronautics, 2016. DOI: 10.2514/6.2016-3598.
- [2] HE L, ZHAO A, WANG X, et al. Path Planning Method for General Aviation under Hazardous Weather Using Heuristic Algorithm[C/OL]//2019 5th International Conference on Transportation Information and Safety (ICTIS). 2019: 920-926. DOI: 10.1109/ICTIS.2019.8883714.
- [3] GRECHE L, JAZOULI M, Es-Sbai N, et al. Comparison between Euclidean and Manhattan distance measure for facial expressions classification[C/OL]//2017 International Conference on Wireless Technologies,

- Embedded and Intelligent Systems (WITS). Fez, Morocco: IEEE, 2017: 1-4. DOI: 10.1109/WITS.2017.7934618.
- [4] GÖNEN M, JOHNSON W O, LU Y, et al. Comparing Objective and Subjective Bayes Factors for the Two-Sample Comparison: The Classification Theorem in Action[J/OL]. *The American Statistician*, 2019, 73 (1): 22-31. DOI: 10.1080/00031305.2017.1322142.
- [5] UZUNOĞLU B. An Adaptive Bayesian Approach With Subjective Logic Reliability Networks for Preventive Maintenance[J/OL]. *IEEE Transactions on Reliability*, 2020, 69(3): 916-924. DOI: 10.1109/TR.2019.2916722.
- [6] 袁杰, 王姝, 王福利, 等. 基于改进主观贝叶斯方法识别电熔镁炉异常工况[J]. *东北大学学报 (自然科学版)*, 2021, 42(2): 153-159.
- [7] PANDEY H M, DIXIT A, MEHROTRA D. Genetic algorithms: Concepts, issues and a case study of grammar induction[C/OL]//*Proceedings of the CUBE International Information Technology Conference on - CUBE '12*. Pune, India: ACM Press, 2012: 263. DOI: 10.1145/2381716.2381766.
- [8] Eight queens puzzle | Kaggle[Z].