



廣東工業大學

《智能优化算法》

课程报告

学 院 _____ 计算机学院 _____

专 业 _____ 软件工程 _____

班 级 _____ 一班 _____

学 号 _____ 3119005028 _____

学生姓名 _____ 魏耀辉 _____

授课教师 _____ 胡晓敏 _____

2022 年 06 月

目录

1 绪论	1
1.1 智能优化算法的发展综述	1
1.2 主要内容及结构	2
2 智能优化算法求解连续函数优化问题	3
2.1 问题定义与优化目标	3
2.2 遗传算法的基本流程	3
2.2.1 遗传算法的运算流程	3
2.2.2 遗传算法操作	5
2.2.3 遗传算法实验参数	5
2.3 差分进化算法基本流程	6
2.3.1 差分进化算法的运算流程	6
2.3.2 差分进化算法操作	7
2.3.3 差分进化算法实验参数	8
2.4 免疫算法基本流程	8
2.4.1 免疫算法的运算流程	9
2.4.2 免疫算法操作	10
2.4.3 免疫算法实验参数	10
2.5 粒子群算法基本流程	11
2.5.1 粒子群算法的运算流程	12
2.5.2 粒子群算法操作	13
2.5.3 粒子群算法实验参数	13
2.6 模拟退火算法基本流程	13
2.6.1 模拟退火算法的运算流程	14
2.6.2 模拟退火算法操作	16
2.6.3 模拟退火算法实验参数	16
2.7 测试结果与分析	16
2.7.1 遗传算法测试版本说明	16

2.7.1 差分进化算法测试版本说明	17
2.7.3 各算法测试结果	17
2.7.4 各算法求解连续函数优化问题分析	19
3 智能优化算法求解旅行商问题	20
3.1 问题定义与优化目标	20
3.2 遗传算法的基本流程	20
3.2.1 遗传算法的运算流程	21
3.2.2 遗传算法操作	22
3.2.3 遗传算法实验参数	22
3.3 免疫算法基本流程	22
3.3.1 免疫算法的运算流程	23
3.3.2 免疫算法操作	24
3.3.3 免疫算法实验参数	24
3.4 蚁群算法的基本流程	25
3.4.1 蚁群算法的运算流程	25
3.4.2 蚁群算法操作	26
3.4.3 蚁群算法实验参数	27
3.5 模拟退火算法基本流程	27
3.5.1 模拟退火算法的运算流程	28
3.5.2 模拟退火算法操作	29
3.5.3 模拟退火算法操作	30
3.6 禁忌搜索算法基本流程	30
3.6.1 禁忌搜索算法的运算流程	31
3.6.2 禁忌搜索算法操作	32
3.6.3 禁忌搜索算法实验参数	33
3.5 测试结果与分析	34
4 总结与心得体会	40
4.1 总结	40
4.2 心得体会	40

参考文献	41
------------	----

1 绪论

1.1 智能优化算法的发展综述

近年来,随着计算机技术的快速发展,为了在一定程度上解决大空间、非线性、全局寻优、组合优化等复杂问题,智能优化算法不断涌现,如:进化类算法、群智能算法、模拟退火算法、禁忌搜索算法和神经网络算法等。因其独特的优点和机制,这些算法得到了国内外学者的广泛关注,掀起了研究热潮,在信号处理、图像处理、生产调度、任务分配模式识别、自动控制和机械设计等众多领域得到了成功应用。

智能优化算法可应用于电子、通信、计算机、自动化、机器人、经济学和管理学等众多学科。在这些众多学科中,不断地出现了许多复杂的组合优化问题。面对这些大型的优化问题,传统的优化方法(如牛顿法、单纯形法等)需要遍历整个搜索空间,无法在短时间内完成搜索,且容易产生搜索的“组合爆炸”。例如,许多工程优化问题,往往需要在复杂而庞大的搜索空间中寻找最优解或者准最优解。鉴于实际工程问题的复杂性、非线性、约束性以及建模困难等诸多特点,寻求高效的优化算法已成为相关学科的主要研究内容之一。

受到人类智能、生物群体社会性或自然现象规律的启发,人们发明了很多智能优化算法来解决上述复杂优化问题,主要包括:模仿自然界生物进化机制的遗传算法;通过群体内个体间的合作与竞争来优化搜索的差分进化算法;模拟生物免疫系统学习和认知功能的免疫算法;模拟蚂蚁集体寻径行为的蚁群算法;模拟鸟群和鱼群群体行为的粒子群算法;源于固体物质退火过程的模拟退火算法;模拟人类智力记忆过程的禁忌搜索算法;模拟动物神经网络行为特征的神经网络算法;等等。这些算法有个共同点,即都是通过模拟或揭示某些自然界的现象和过程或生物群体的智能行为而得到发展;在优化领域称它们为智能优化算法,它们具有简单、通用、便于并行处理等特点。

1.2 主要内容及结构

本报告主要研究了智能优化算法中遗传算法（Genetic Algorithm , GA）、差分进化算法（Differential Evolution, DE）、免疫算法（Immune Algorithm, IA）、粒子群算法（Particle Swarm Optimization, PSO）、模拟退火算法（Simulated Annealing, SA）求解连续函数优化问题的对比分析，以及遗传算法、免疫算法、蚁群算法（Ant Colony Optimization, ACO）、禁忌搜索算法（Tabu Search or Taboo Search, TS）求解旅行商问题。本报告使用 MATLAB 作为研究算法的仿真实现工具，通过对比各种算法的迭代次数和迭代最优值，单次测试和多次测试，比较最优解、最差解、平均解、标准差等来比较各种算法的收敛结果。最后测试各种算法在一定阈值内的找到满足精度的函数解的成功次数和成功率。

本报告将主要分为 4 章。

第一章是绪论，简要介绍智能优化算法的发展以及主要内容及结构。

第二章是智能优化算法求解连续函数优化问题。介绍了通过 MATLAB 进行仿真测试及比较遗传算法、差分进化算法、免疫算法、粒子群算法、模拟退火算法的参数及优化值。

第三章是智能优化算法求解旅行商问题。介绍了通过 MATLAB 进行仿真测试及比较遗传算法、免疫算法、蚁群算法、禁忌搜索算法求解旅行商问题的最优值。

第四章是总结与展望。

2 智能优化算法求解连续函数优化问题

本章对比分析遗传算法、差分进化算法、免疫算法、粒子群算法、模拟退火算法各自求解连续函数优化问题的能力，并对算法的特点进行讨论。

2.1 问题定义与优化目标

问题定义：计算下列连续函数（1）的最小值

$$f(x) = \sum_{i=1}^n x_i^2 \quad (-20 \leq x \leq 20) \quad (1)$$

优化目标：个体 x 的维数 $n=10$ 。这是一个简单的平方和函数，只有一个极小点 $x = (0, 0, \dots, 0)$ ，理论最小值 $f(0, 0, \dots, 0) = 0$ 。优化目标即是使函数的仿真值尽可能收敛为 0。

2.2 遗传算法的基本流程

遗传算法（Genetic Algorithm, GA）是模拟生物在自然环境中的遗传和进化过程，从而形成的自适应全局优化搜索算法。它起源于 20 世纪 60 年代人们对自然和人工自适应系统的研究，最早由美国 J.H. Holland 教授提出^[1]的，并于 80 年代 D.J. Goldberg 在一系列研究工作的基础上归纳总结而成。

遗传算法是通过模仿自然界生物进化机制而发展起来的随机全局搜索和优化方法。它借鉴了达尔文的进化论和孟德尔的遗传学说，使用“适者生存”的原则，本质上是一种并行、高效、全局搜索的方法；它能在搜索过程中自动获取和积累有关搜索空间的知识，并自适应地控制搜索过程以求得最优解^[2]。

2.2.1 遗传算法的运算流程

遗传算法的运算流程如图 2.1 所示。具体步骤如下：

（1）初始化。设置进化代数计数器 $g=0$ ，设置最大进化代数 G ，随机生成 N_p 个个体作为初始群体 $P(0)$ 。

- (2) 个体评价。计算群体 $P(t)$ 中各个个体的适应度。
- (3) 选择运算。将选择算子作用于群体，根据个体的适应度，按照一定的规则或方法，选择一些优良个体遗传到下一代群体。
- (4) 交叉运算。将交叉算子作用于群体，对选中的成对个体，以某一概率交换它们之间的部分染色体，产生新的个体。
- (5) 变异运算。将变异算子作用于群体，对选中的个体，以某一概率改变某一个或某一些基因值为其他的等位基因。
- (6) 循环操作。群体 $P(t)$ 经过选择、交叉和变异运算之后得到下一代群体 $P(t+1)$ 。计算其适应度值，并根据适应度值进行排序，准备进行下一次遗传操作
- (7) 终止条件判断：若： $g \leq G$ ，则 $g=g+1$ ，转到步骤（2）；若 $g > G$ ，则此进化过程中所得到的具有最大适应度的个体作为最优解输出，终止计算。

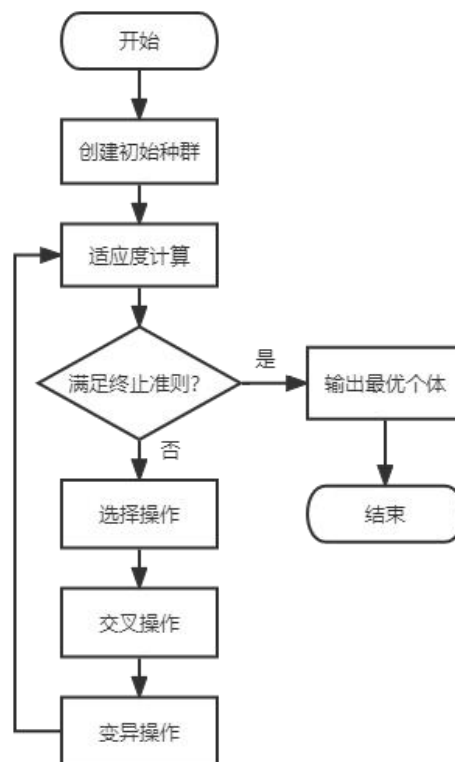


图 2.1 遗传算法的运算流程

2.2.2 遗传算法操作

遗传算法操作如下：

- (1) 创建初始种群
- (2) 适应度计算
- (3) 若满足终止准则，输出当前最优个体，结束操作，否则继续操作
- (4) 选择操作
- (5) 交叉操作
- (6) 变异操作，转到步骤 (3)

2.2.3 遗传算法实验参数

遗传算法实验参数如下：

- (1) 基于轮盘赌的遗传算法实验参数如表 2.1 所示：

表 2.1 基于轮盘赌的遗传算法实验参数

二进制串长 L	种群数量 N_p	上限 X_s	下限 X_x
20	50	20	-20
最大遗传代数 G	交叉概率 P_c	变异概率 P_m	
100	0.8	0.1	

- (2) 采用君主方案的遗传算法实验参数如表 2.2 所示：

表 2.2 基于轮盘赌的遗传算法实验参数

基因数目 D	染色体数目 N_p	上限 X_s	下限 X_x
10	100	20	-20
最大遗传代数 G	交叉概率 P_c	变异概率 P_m	
1000	0.8	0.1	

2.3 差分进化算法基本流程

差分进化 (Differential Evolution, DE) 算法最初用于解决切比雪夫多项式问题^[3]，后来发现该算法也是解决复杂优化问题的有效技术。

差分进化算法是一种新兴的进化计算技术，它基于群体智能理论，是通过群体内个体间的合作与竞争产生的智能优化搜索算法。但相比于进化计算，差分进化算法保留了基于种群的全局搜索策略，采用实数编码、基于差分的简单变异操作和“一对一”的竞争生存策略，降低了进化计算的复杂性。同时，差分进化算法具有较强的全局收敛能力和鲁棒性（又称稳健性），且不需要借助问题的特征信息，适用于求解一些利用常规的数学规划方法很难求解甚至无法求解的复杂优化问题^[4]。

2.3.1 差分进化算法的运算流程

差分进化算法的运算流程如图 2.1 所示。具体步骤如下：

- （1）确定差分进化算法的控制参数和所要采用的具体策略。差分进化算法的控制参数包括：种群数量、变异算子、交叉算子、最大进化代数、终止条件等。
- （2）随机产生初始种群，进化代数 $k=1$ 。
- （3）对初始种群进行评价，即计算初始种群中每个个体的目标函数值。
- （4）判断是否达到终止条件或达到最大进化代数：若是，则进化终止，将此时的最佳个体作为解输出；否则，继续下一步操作。
- （5）进行变异操作和交叉操作，对边界条件进行处理，得到临时种群。
- （6）对临时种群进行评价，计算临时种群中每个个体的目标函数值。
- （7）对临时种群中的个体和原种群中对应的个体，进行“一对一”的选择操作，得到新种群。
- （8）进化代数 $k=k+1$ ，转步骤（4）。

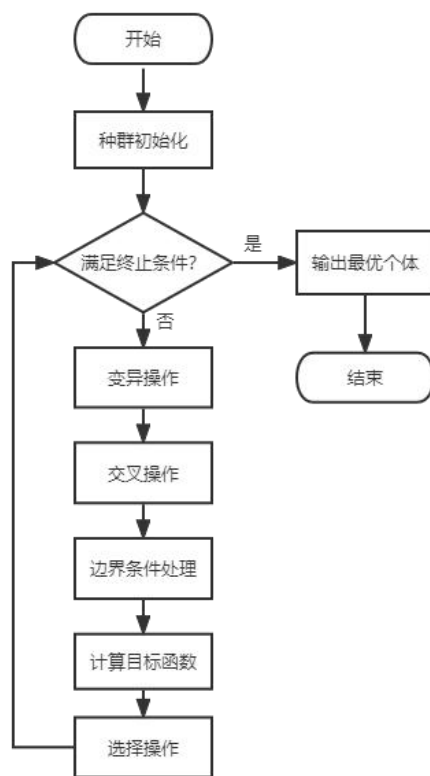


图 2.2 差分进化算法的运算流程

2.3.2 差分进化算法操作

差分进化算法操作如下：

- (1) 种群初始化
- (2) 若满足终止条件，输出当前最优个体，结束操作，否则继续操作
- (3) 变异操作
- (4) 交叉操作
- (5) 边界条件处理
- (6) 计算目标函数
- (7) 选择操作，转到步骤（2）

2.3.3 差分进化算法实验参数

(1) 无自适应算子的差分进化算法实验参数如表 2.3 所示:

表 2.3 无适应算子差分进化算法实验参数

变量维数 D	种群数量 N_p	上限 X_s	下限 X_x
10	100	20	-20
最大进化代数 G	初始变异算子 $F0$	交叉算子 CR	
200	0.8	0.1	

(2) 含自适应算子的差分进化算法实验参数如表 2.4 所示:

表 2.4 含自适应算子差分进化算法实验参数

变量维数 D	种群数量 N_p	上限 X_s	下限 X_x
10	100	20	-20
最大进化代数 G	初始变异算子 $F0$	交叉算子 CR	
200	0.8	0.1	

2.4 免疫算法基本流程

最早的免疫系统模型由 Jerne 于 1973 年提出, 他基于 Burnet 的克隆选择学说, 开创了独特型网络理论, 给出了免疫系统的数学框架, 并采用微分方程建模来仿真淋巴细胞的动态变化。Farmal 等人于 1986 年基于免疫网络学说理论构造出免疫系统的动态模型, 展示了免疫系统与其他人工智能方法相结合的可能性, 开创了免疫系统研究的先河^[5]。

免疫算法(Immune Algorithm, IA), 是模仿生物免疫机制, 结合基因的进化机理, 人工构造出的一种新型智能搜索算法。免疫算法具有一般免疫系统的特征, 它采用群体搜索策略, 通过迭代计算, 最终以较大的概率得到问题的最优解。相比于其他算法, 免疫算法克服了一般寻优过程中(特别是多峰值的寻优过程中)不可避免的“早熟”问题, 可求得全局最优解, 具有自适应性、随机性、并行性全局收敛性、种群多样性等优点^[6]。

2.4.1 免疫算法的运算流程

免疫算法的运算流程如图 2.3 所示。具体步骤如下：

- (1) 初始化免疫个体 $D=10$, 免疫种群个体数为 $NP=100$, 最大免疫代数为 $G=500$, 变异概率为 $Pm=0.7$, 激励度系数为 $\alpha=1$, $\beta=1$, 相似度阈值为 $\delta=0.2$, 克隆个数为 $N_{cl}=10$ 。
- (2) 随机产生初始种群, 计算个体亲和度、抗体浓度和激励度, 并按激励度排序。
- (3) 取激励度前 $NP/2$ 个个体进行克隆、变异、克隆抑制的免疫操作, 免疫后的种群进行激励度计算。
- (4) 随机生成 $NP/2$ 个个体的新种群, 并计算个体亲和度、抗体浓度和激励度; 免疫种群和随机种群合并, 按激励度排序, 进行免疫迭代。
- (5) 判断是否满足终止条件: 若满足, 则结束搜索过程, 输出优化值; 若不满足; 若不满足, 则继续进行迭代优化。

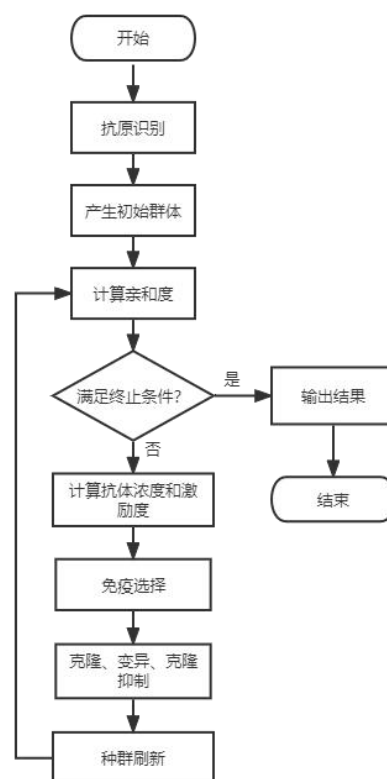


图 2.3 免疫算法的运算流程

2.4.2 免疫算法操作

免疫算法操作如下：

- (1) 首先进行抗原识别，即理解待优化的问题，对问题进行可行性分析，提取先验知识，构造出合适的亲和度函数，并制定各种约束条件。
- (2) 后产生初始抗体群，通过编码把问题的可行解表示成解空间中的抗体，在解的空间内随机产生一个初始种群。
- (3) 对种群中的每一个可行解进行亲和度评价。
- (4) 判断是否满足算法终止条件：如果满足条件，则终止算法寻优过程，输出计算结果；否则，继续寻优运算。
- (5) 计算抗体浓度和激励度。
- (6) 进行免疫操作，包括免疫选择、克隆、变异和克隆抑制。
 - 1) 免疫选择：根据种群中抗体的亲和度和浓度计算结果选择优质抗体，使其活化；
 - 2) 克隆：对活化的抗体进行克隆复制，得到若干副本；
 - 3) 变异：对克隆得到的副本进行变异操作，使其发生亲和度突变；
 - 4) 克隆抑制：对变异结果进行再选择，抑制亲和度低的抗体，保留亲和度高的变异结果。
- (7) 种群刷新，以随机生成的新抗体替代种群中激励度较低的抗体，形成新一代抗体，转步骤(3)

2.4.3 免疫算法实验参数

- (1) 免疫算法实验参数如图 2.5 所示：

表 2.5 免疫算法实验参数

免疫个体维数 D	免疫个体 N_p	上限 X_s
10	100	20
激励系数 $beta$	免疫代数 gen	克隆个数 N_{cl}
1	0	10

下限 X_r	最大免疫代数 G	变异概率 P_m
-20	200	0.8
邻域初值 $detal0$	相似度阈值 $detas$	激励系 $alfa$
$1 * X_s$	0.2	1

2.5 粒子群算法基本流程

1995 年，美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 共同提出了粒子群算法 (Particle Swarm Optimization, PSO)，该算法的提出是受对鸟类群体行为进行建模与仿真的研究结果的启发。他们的模型和仿真算法主要对 Erank Heppner 的模型进行了修正，以使粒子飞向解空间并在最优解处降落。粒子群算法一经提出，由于其算法简单，容易实现，立刻引起了进化计算领域学者们的广泛关注，形成一个研究热点。2001 年出版的 J.Kennedy 与 R.Eberhart 合著的《群体智能》将群体智能的影响进一步扩大”，随后关于粒子群算法的研究报告和研究成果大量涌现，继而掀起了国内外研究热潮。

粒子群算法来源于对鸟类群体活动规律性的研究，进而利用群体智能建立一个简化的模型。它模拟鸟类的觅食行为，将求解问题的搜索空间比作鸟类的飞行空间，将每只鸟抽象成一个没有质量和体积的粒子，用它来表征问题的一个可行解，将寻找问题最优解的过程看成鸟类寻找食物的过程，进而求解复杂的优化问题。粒子群算法与其他进化算法一样，也是基于“种群”和“进化”的概念，通过个体间的协作与竞争，实现对复杂空间最优解的搜索。同时，它又不像其他进化算法那样对个体进行交叉、变异、选择等进化算子操作，而将群体中的个体看作在 D 维搜索空间中具有质量和体积的粒子，每个粒子以一定的速度在解空间运动，并向自身历史最佳位置和群体历史最佳位置聚集，实现对候选解的进化。粒子群算法具有很好的生物社会背景而易于理解，由于参数少而容易实现，对非线性、多峰问题均具有较强的全局搜索能力，在科学研究与工程实践中得到了广泛关注。目前，该算法已广泛应用于函数优化、神经网络训练、模式分类、模糊控制等领域。

2.5.1 粒子群算法的运算流程

粒子群算法的运算流程如图 2.5 所示。具体步骤如下：

- (1) 初始化粒子群，包括群体规模 N ，每个粒子的位置 X_i 和速度 v_i
- (2) 计算每个粒子的适应度值 $fit[i]$
- (3) 对每个粒子，用它的适应度值 $fit[i]$ 和个体极值比较。如果 $fit[i] < p_{best}(i)$ ，则用 $fit[i]$ 替换掉。
- (4) 对每个粒子，用它的适应度值 $fit[i]$ 和全局极值比较。如果 $fit[i] < g_{best}(i)$ ，则用 $fit[i]$ 替换 $g_{best}(i)$
- (5) 迭代更新粒子的速度 v_i 和位置 X_i
- (6) 进行边界条件处理。
- (7) 判断算法终止条件是否满足：若是，则结束算法并输出优化结果；否则返回步骤 (2)

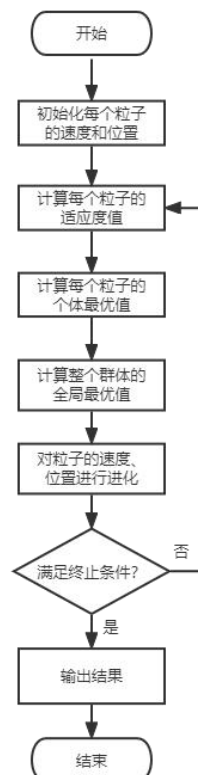


图 2.5 粒子群算法的运算流程

2.5.2 粒子群算法操作

粒子群算法操作如下：

- (1) 初始化群体粒子个数为 $N=100$ ，粒子维数为 $D=10$ ，最大迭代次数为 $T=200$ ，学习因子 $C1=C2=1.5$ ，惯性权重为 $w=0.8$ ，位置最大值为 $x_{max}=20$ ，位置最小值为 $x_{min}=-20$ ，速度最大值为 $v_{max}=10$ ，速度最小值为 $v_{min}=-10$ 。
- (2) 初始化种群粒子位置 x 和速度值 v ，粒子个体最优位置 p 和最优值 p_{best} ，以及粒子群全局最优位置 g 和最优值 g_{best} 。
- (3) 更新位置 x 和速度值 v ，并进行边界条件处理，判断是否替换粒子个体最优位置 p 和最优值 p_{best} 、粒子群全局最优位置 g 和最优值 g_{best} 。
- (4) 判断是否满足终止条件：若满足，则结束搜索过程，输出优化值；若不满足，则继续进行迭代优化。

2.5.3 粒子群算法实验参数

- (1) 粒子群算法实验参数如表 2.6 所示：

表 2.6 粒子群算法实验参数

群体粒子个数 N	粒子维数 D	最大迭代次数 T	惯性权重 w
100	10	200	0.8
学习因子 $c1$	学习因子 $c2$	位置最大值 x_{max}	
1.5	1.5	20	
位置最小值 x_{min}	速度最大值 v_{max}	速度最小值 v_{min}	
-20	10	-10	

2.6 模拟退火算法基本流程

模拟退火（Simulated Annealing, SA）算法的思想最早由 Metropolis 等人于 1953 提出；Kirkpatrick 等人于 1983 年第一次使用模拟退火算法求解组合最优化问题。模拟退火算法是一种基于 Monte Carlo 迭代求解策略的随机寻优算法，其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的

相似性。其目的在于，为具有 NP (Non-deterministic Polynomial)复杂性的问题提供有效的近似求解算法，它克服了其他优化过程容易陷入局部极小的缺陷和对初值的依赖性。模拟退火算法是一种通用的优化算法，是局部搜索算法的扩展。它不同于局部搜索算法之处是以一定的概率选择邻域中目标值大的劣质解。从理论上说，它是一种全局最优算法。模拟退火算法以优化问题的求解与物理系统退火过程的相似性为基础，它利用 Metropolis 算法并适当地控制温度的下降过程来实现模拟退火，从而达到求解全局优化问题的目的^[9]。

模拟退火算法是一种能应用到求最小值问题的优化过程。在此过程中，每生更新过程的长度都与相应的参数成正比，这些参数扮演着温度的角色。与金属退火原理相类似，在开始阶段为了更快地最小化，温度被升得很高，然后才慢慢降温以求稳定^[10]。

2.6.1 模拟退火算法的运算流程

模拟退火算法的运算流程如图 2.6 所示。具体步骤如下：

模拟退火算法新解的产生和接受可分为如下三个步骤：

(1) 由一个产生函数从当前解产生一个位于解空间的新解；为便于后续的计算和接受，减少算法耗时，通常选择由当前解经过简单交换即可产生新解的方法。注意，产生新解的变换方法决定了当前新解的邻域结构，因而对冷却进度表的选取有一定的影响。

判断新解是否被接受，判断的依据是一个接受准则，最常用的接受准则是 Metropolis 准则：若 $\Delta E < 0$ ，则接受 X' 作为新的当前解 X ；否则，以概率 $\exp(-\Delta E/T)$ 接受 X' 作为新的当前解 X 。

(2) 当新解被确定接受时，用新解代替当前解，这只需将当前解中对应于产生新解时的变换部分予以实现，同时修正目标函数值即可。此时，当前解实现了一次迭代，可在此基础上开始下一轮试验。若当新解被判定为舍弃，则在原当前解的基础上继续下一轮试验。

模拟退火算法求得的解与初始解状态（算法迭代的起点）无关，具有渐近收敛性，已在理论上被证明是一种以概率 1 收敛于全局最优解的优化算法。模

拟退火算法可以分解为解空间、目标函数和初始解三部分。该算法具体流程如下：

- (1) 初始化：设置初始温度 T_0 （充分大）、初始解状态 X_0 （是算法迭代的起点）、每个 T 值的迭代次数 L ；
- (2) 对 $k=1, \dots, L$ 进行第（3）至第（6）步操作，
- (3) 产生新解 X' ；
- (4) 计算增量 $\Delta E = E(X') - E(X)$ ，其中 $E(X)$ 为评价函数；
- (5) 若 $\Delta E < 0$ ，则接受 X' 作为新的当前解，否则以概率 $\exp(-\Delta E/T)$ 接受 X' 作为新的当前解 X ；
- (6) 如果满足终止条件，则输出当前解作为最优解，结束程序；
 T 逐渐减小，且 $T \rightarrow 0$ ，然后转第（2）步。

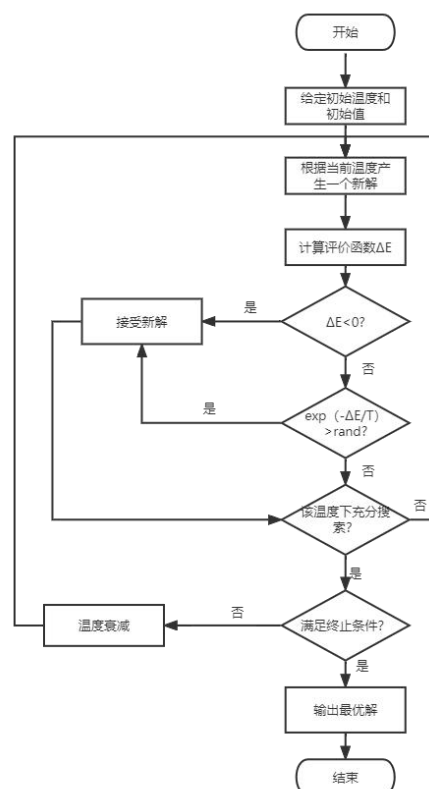


图 2.6 模拟退火算法的运算流程

2.6.2 模拟退火算法操作

模拟退火算法操作如下：

(1) 初始化 Markov 链长度为 $L=200$ ，衰减参数为 $K=0.998$ ，步长因子为 $S=0.01$ ，初始温度为 $T=100$ ，容差为 $YZ=1 \times 10^{-8}$ ，随机产生初始解，并计算其目标函数值。

(2) 在变量的取值范围内，按步长因子随机产生新解，并计算新目标两数值：以 Metropolis 算法确定是否替代旧解，在一种温度下，迭代 L 次。

(3) 判断是否满足终止条件：若满足，则结束搜索过程，输出优化值：若不足，则减小温度，进行迭代优化。

2.6.3 模拟退火算法实验参数

(1) 模拟退火算法实验参数如表 2.7 所示：

表 2.7 模拟退火算法实验参数

变量维数 D	上限 X_s	下限 X_x
10	20	-20
马尔科夫链长度 L	衰减参数 K	步长因子 S
200	0.998	0.01
初始温度 T	容差 YZ	总接受点 P
100	$1e-8$	0

2.7 测试结果与分析

2.7.1 遗传算法测试版本说明

GA1 为基于轮盘赌的遗传算法，GA2 为采用君主方案的遗传方案。遗传算法测试版本如表 2.8 所示：

表 2.8 遗传算法测试版本

策略	GA1	GA2
交叉策略	单点交叉	君主交叉
变异策略	域内随机值	域内随机值
选择策略	轮盘赌	合并新旧种群取最优

2.7.1 差分进化算法测试版本说明

DE1 为含有自适应因子的差分进化算法，DE2 为不含有自适应因子的差分进化算法。差分进化算法测试版本如表 2.9 所示：

表 2.9 差分进化算法测试版本

策略	GA1	GA2
交叉策略	DE 交叉	DE 交叉
变异策略	DE/rand/1/bin+自适应变异	DE/rand/1/bin
选择策略	新旧个体比较	新旧个体比较

2.7.3 各算法测试结果

各算法测试结果如表 2.10 所示：

表 2.10 各算法测试结果

算法	最优函数值	最差函数值	平均值	标准差	算法时间/s
GA1	3.63799e-10	3.63799e-10	3.63799e-10	0	0.34945
GA2	0.000121683	0.0020268	0.000561772	0.000442439	0.56895
DE1	3.63799e-10	3.63799e-10	3.63799e-10	0	0.24975
DE2	3.63799e-10	3.63799e-10	3.63799e-10	0	0.24525
IA	0.000335965	0.00086047	0.000595988	0.000163954	1.5502
PSO	1.12749e-09	1.73201e-07	3.95962e-08	4.07503e-08	0.0653
SA	0.0067402	0.0133729	0.0103217	0.00194717	4.4273

各算法测试成功率如表 2.11 所示：

表 2.11 各算法测试成功率

算法	算法成功率
GA1	100%
GA2	90%
DE1	100%
DE2	100%
IA	0%
PSO	100%
SA	0%

各算法平均收敛曲线图如图 2.7 所示：

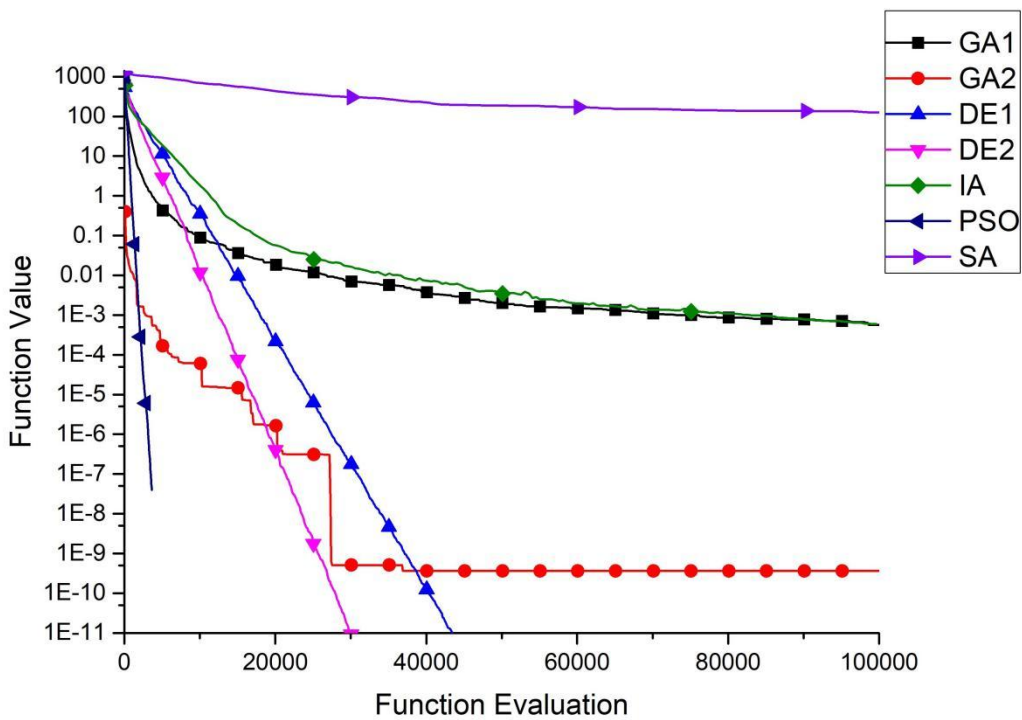


图 2.7 各算法平均收敛曲线

2.7.4 各算法求解连续函数优化问题分析

通过大量的仿真实验以及文件记录和测试，可以得出以下结论分析：

GA1 算法：基于轮盘赌的遗传算法，在经过较多次迭代后，能够得到十分不错的适应度值，且耗时在众多算法中排名中等，在解决连续函数优化问题上 GA1 的性能中等，但也能得到相对其他算法不错的函数优化值。

GA2 算法：基于君主交叉的遗传算法，在相同的函数评价次数的情况下，GA2 稍逊色于 GA1，函数值优化一般，但在解决连续函数优化问题的相同情况下优于免疫算法和模拟退火算法。

DE1 算法：能对变异算子进行自适应更新，用时比 DE2 稍长，DE1 的函数优化值多次实验后优于遗传算法，且耗时比遗传算法少。DE1 明显在解决连续函数优化问题要优于遗传算法。

DE2 算法：固定变异算子，变异效率提高，耗时比 DE1 稍好，交叉概率和变异概率的选择较为重要，会直接影响每次遗传的结果，从而对结果有较大的影响。

IA 算法：免疫算法，免疫算法在相同的函数评价次数时，函数优化值不理想，排名倒数第二，且耗时较长，可见，免疫算法能够解决函数连续优化问题，但在本次实验的测试函数中不如遗传算法和差分进化算法。

PSO 算法：函数值在较少的函数评价次数能有最好的函数优化值，粒子群算法的成功率达到了 100%且耗时最短，所以 PSO 算法在解决连续函数优化问题时，是本次实验中表现最好的算法。

SA 算法：模拟退火算法在本次众多算法中，优化值最差，耗时最长，因此在相同条件下，模拟退火算法的求解能力最差，故不推荐使用。

总结：粒子群算法在这几种算法中有着非常好的表现，其次为差分进化算法和遗传算法，而免疫算法和模拟退火算法在该问题上表现一般且耗时长，不推荐。粒子群算法的耗时最短，成功率最高，由此得出，粒子群算法在对于求解连续函数优化问题上无论是效率还是成功率都有着非常大的优势。

3 智能优化算法求解旅行商问题

本章对比分析遗传算法、免疫算法、蚁群算法、模拟退火算法、禁忌搜索算法各自求解旅行商问题的能力，并对算法的特点进行讨论。

3.1 问题定义与优化目标

问题定义：旅行商问题（TSP）。假设有一个旅行商人要拜访全国 31 个省会城市，他需要算则所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。对路径选择的要求是：所选路径的路程为所有路径之中的最小值。

全国 31 个省会城市的坐标为[1304 2312;3639 1315;4177 2244;3712 1399;3488 1535;3326 1556;3238 1229;4196 1044;4312 790;4386 570;3007 1970;2562 1756;2788 1491;2381 1676;1332 695;3715 1678;3918 2179;4061 2370;3780 2212;3676 2578;4029 2838;4263 2931;3429 1908;3507 2376;3394 2643;3439 3201;2935 3240;3140 3550;2545 2357;2778 2826;2370 2975]

优化目标：通过算法生成最优路径

3.2 遗传算法的基本流程

遗传算法（Genetic Algorithm,GA）是模拟生物在自然环境中的遗传和进化过程，从而形成的自适应全局优化搜索算法。它起源于 20 世纪 60 年代人们对自然和人工自适应系统的研究，最早由美国 JH.Holland 教授提出的，并于 80 年代 D.J.Goldberg 在一系列研究工作的基础上归纳总结而成。

遗传算法是通过模仿自然界生物进化机制而发展起来的随机全局搜索和优化方法。它借监了达尔文的进化论和孟德尔的遗传学说，使用“适者生存”的原则，本质上是一种并行、高效、全局搜索的方法；它能在搜索过程中自动获取和积累有关搜索空间的知识，并自适应地控制搜索过程以求得最优解。

3.2.1 遗传算法的运算流程

遗传算法的运算流程如图 3.1 所示。具体步骤如下：

- (1) 初始化。设置进化代数计数器 $g=0$ ，设置最大进化代数 G ，随机生成 N_p 个个体作为初始群体 $P(0)$ 。
- (2) 个体评价。计算群体 $P(t)$ 中各个个体的适应度。
- (3) 选择运算。将选择算子作用于群体，根据个体的适应度，按照一定的规则或方法，选择一些优良个体遗传到下一代群体。
- (4) 交叉运算。将交叉算子作用于群体，对选中的成对个体，以某一概率交换它们之间的部分染色体，产生新的个体。
- (5) 变异运算。将变异算子作用于群体，对选中的个体，以某一概率改变某一个或某一些基因值为其他的等位基因。
- (6) 循环操作。群体 $P(t)$ 经过选择、交叉和变异运算之后得到下一代群体 $P(t+1)$ 。计算其适应度值，并根据适应度值进行排序，准备进行下一次遗传操作
- (7) 终止条件判断：若： $g \leq G$ ，则 $g=g+1$ ，转到步骤（2）；若 $g > G$ ，则此进化过程中所得到的具有最大适应度的个体作为最优解输出，终止计算。

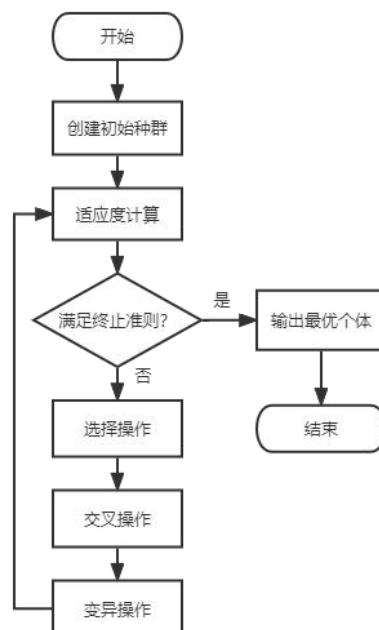


图 3.1 遗传算法的运算流程

3.2.2 遗传算法操作

遗传算法操作如下：

- (1) 创建初始种群
- (2) 适应度计算
- (3) 若满足终止准则，输出当前最优个体，结束操作，否则继续操作
- (4) 选择操作
- (5) 交叉操作
- (6) 变异操作，转到步骤 (3)

3.2.3 遗传算法实验参数

- (1) 遗传算法实验参数如表 3.1 所示：

表 3.1 遗传算法实验参数

种群数目 N_p	基因维数 N	最大进化代数 G
200	31	1000

3.3 免疫算法基本流程

最早的免疫系统模型由 Jernl 于 1973 年提出，他基于 Burnet 的克隆选择学说，开创了独特型网络理论，给出了免疫系统的数学框架，并采用微分方程建模来仿真淋巴细胞的动态变化。Farmal 等人于 1986 年基于免疫网络学说理论构造出免疫系统的动态模型，展示了免疫系统与其他人工智能方法相结合的可能性，开创了免疫系统研究的先河。

免疫算法(Immune Algorithm, IA)，是模仿生物免疫机制，结合基因的进化机理，人工构造出的一种新型智能搜索算法。免疫算法具有一般免疫系统的特征，它采用群体搜索策略，通过迭代计算，最终以较大的概率得到问题的最优解。相比于其他算法，免疫算法克服了一般寻优过程中（特别是多峰值的寻优过程中）不可避免的“早熟”问题，可求得全局最优解，具有自适应性、随机性、并行性全局收敛性、种群多样性等优点。

3.3.1 免疫算法的运算流程

免疫算法的运算流程如图 3.2 所示。具体步骤如下：

- (1) 初始化免疫个体 $D=10$, 免疫种群个体数为 $NP=100$, 最大免疫代数为 $G=500$, 变异概率为 $Pm=0.7$, 激励度系数为 $\alpha=1$, $\beta=1$, 相似度阈值为 $\delta=0.2$, 克隆个数为 $N_{cl}=10$ 。
- (2) 随机产生初始种群, 计算个体亲和度、抗体浓度和激励度, 并按激励度排序。
- (3) 取激励度前 $NP/2$ 个个体进行克隆、变异、克隆抑制的免疫操作, 免疫后的种群进行激励度计算。
- (4) 随机生成 $NP/2$ 个个体的新种群, 并计算个体亲和度、抗体浓度和激励度; 免疫种群和随机种群合并, 按激励度排序, 进行免疫迭代。
- (5) 判断是否满足终止条件: 若满足, 则结束搜索过程, 输出优化值; 若不满足; 若不满足, 则继续进行迭代优化。

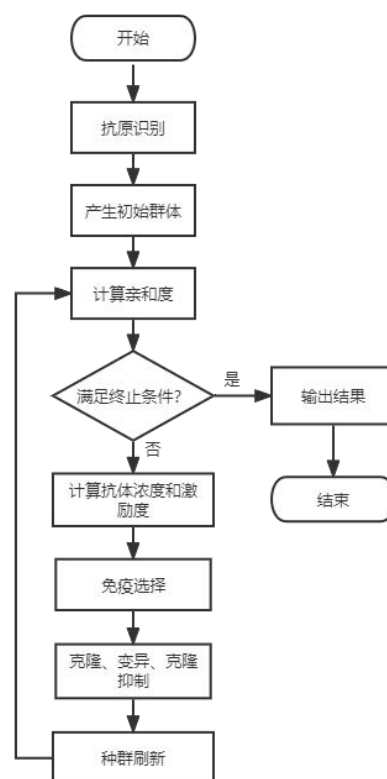


图 3.2 免疫算法的运算流程

3.3.2 免疫算法操作

免疫算法操作如下：

- (1) 首先进行抗原识别，即理解待优化的问题，对问题进行可行性分析，提取先验知识，构造出合适的亲和度函数，并制定各种约束条件。
- (2) 后产生初始抗体群，通过编码把问题的可行解表示成解空间中的抗体，在解的空间内随机产生一个初始种群。
- (3) 对种群中的每一个可行解进行亲和度评价。
- (4) 判断是否满足算法终止条件：如果满足条件，则终止算法寻优过程，输出计算结果；否则，继续寻优运算。
- (5) 计算抗体浓度和激励度。
- (6) 进行免疫操作，包括免疫选择、克隆、变异和克隆抑制。
 - 1) 免疫选择：根据种群中抗体的亲和度和浓度计算结果选择优质抗体，使其活化；
 - 2) 克隆：对活化的抗体进行克隆复制，得到若干副本；
 - 3) 变异：对克隆得到的副本进行变异操作，使其发生亲和度突变；
 - 4) 克隆抑制：对变异结果进行再选择，抑制亲和度低的抗体，保留亲和度高的变异结果。
- (7) 种群刷新，以随机生成的新抗体替代种群中激励度较低的抗体，形成新一代抗体，转步骤(3)

3.3.3 免疫算法实验参数

- (2) 免疫算法实验参数如表 3.2 所示：

表 3.2 免疫算法实验参数

城市个数 N	种群个体数 NP	最大免疫代数 G
10	100	20
克隆个数 N_{cl}	城市距离矩阵 D	
1	D	

3.4 蚁群算法的基本流程

蚁群算法是一种源于大自然生物世界的新的仿生进化算法，是由意大利学者 M. Dorigo, V. Maniezzo 和 A. Colomi 等人于 20 世纪 90 年代初期通过模拟自然界中蚂蚁集体寻径行为而提出的一种基于种群的启发式随机搜索算法^[7]。蚂蚁有能力在没有任何提示的情形下找到从巢穴到食物源的最短路径，并且能随环境的变化，适应性地搜索新的路径，产生新的选择。其根本原因是蚂蚁在寻找食物时，能在其走过的路径上释放一种特殊的分泌物信息素（也称外激素），随着时间的推移该物质会逐渐挥发，后来的蚂蚁选择该路径的概率与当时这条路径上的信息素强度成正比。当一条路径上通过的蚂蚁越来越多时，其留下的信息素也越来越多，后来蚂蚁选择该路径的概率也就越高，从而更增加了该路径上的信息素强度。而强度大的信息素会吸引更多的蚂蚁，从而形成一种正反馈机制。通过这种正反馈机制，蚂蚁最终可以发现最短路径^[8]。

3.4.1 蚁群算法的运算流程

蚁群算法的运算流程如图 3.3 所示。具体步骤如下：

- （1）参数初始化。令时间 $t=0$ 和循环次数 $N_c=0$ ，设置最大循环次数 G ，将 m 个蚂蚁置于 n 个元素（城市）上，令有向图上每条边 (i,j) 的初始化信息量 $\tau_{ij}(t)=c$ ，其中 c 表示常数，且初始时刻 $\Delta\tau_{ij}(0)=0$ 。
- （2）循环次数 $N_c=N_c+1$ 。
- （3）蚂蚁的禁忌表索引号 $k=1$ 。
- （4）蚂蚁数目 $k=k+1$ 。
- （5）蚂蚁个体根据状态转移概率公式计算的概率选择元素 j 并前进，
- （6）修改禁忌表指针，即选择好之后将蚂蚁移动到新的元素，并把该元素移动到该蚂蚁个体的禁忌表中。
- （7）若集合 C 中元素未遍历完，即 $k < m$ ，则跳转到第（4）步；否则，执行第（8）步。
- （8）记录本次最佳路线。

(9) 更新每条路径上的信息量。

(10) 若满足结束条件，即如果循环次数 $N_c \geq G$ ，则循环结束并输出程序优化结果；否则清空禁忌表并跳转到第（2）步。

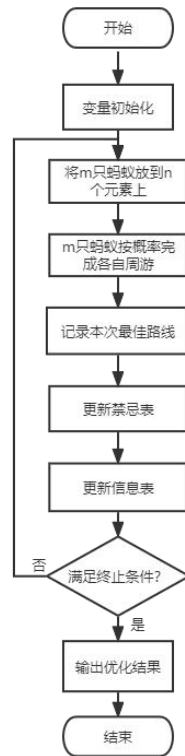


图 3.3 蚁群算法的运算流程

3.4.2 蚁群算法操作

蚁群算法操作如下：

(1) 初始化蚂蚁个数 $m=50$ ，信息素重要程度参数 $Alpha=1$ ，启发式因子重要程度参数 $Beta=5$ ，信息素蒸发系数 $Rho=0.1$ ，最大迭代次数 $G=200$ ，信息素增加强度系数 $Q=100$ 。

(2) 将 m 个蚂蚁置于 n 个城市上，计算待选城市的概率分布， m 只蚂蚁按概率函数选择下一座城市，完成各自的周游。

(3) 记录本次迭代最佳路线，更新信息素，禁忌表清零。

(4) 判断是否满足终止条件：若满足，则结束搜索过程，输出优化值；若不满足，则继续进行迭代优化。

3.4.3 蚁群算法实验参数

(1) 蚁群算法实验参数如表 3.3 所示:

表 3.3 蚁群算法实验参数

蚂蚁个数 m	信息素重要程度 $Alpha$	启发式因子重要程度 $Beta$
50	1	5
信息素蒸发系数 Rho	最大迭代次数 G	信息素增加强度 Q
0.1	200	100

3.5 模拟退火算法基本流程

模拟退火 (Simulated Annealing, SA) 算法的思想最早由 Metropolis 等人于 1953 提出; Kirkpatrick 等人于 1983 年第一次使用模拟退火算法求解组合最优化问题。模拟退火算法是一种基于 Monte Carlo 迭代求解策略的随机寻优算法, 其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。其目的在于, 为具有 NP (Non-deterministic Polynomial) 复杂性的问题提供有效的近似求解算法, 它克服了其他优化过程容易陷入局部极小的缺陷和对初值的依赖性。模拟退火算法是一种通用的优化算法, 是局部搜索算法的扩展。它不同于局部搜索算法之处是以一定的概率选择邻域中目标值大的劣质解。从理论上说, 它是一种全局最优算法。模拟退火算法以优化问题的求解与物理系统退火过程的相似性为基础, 它利用 Metropolis 算法并适当地控制温度的下降过程来实现模拟退火, 从而达到求解全局优化问题的目的。

模拟退火算法是一种能应用到求最小值问题的优化过程。在此过程中, 每生更新过程的长度都与相应的参数成正比, 这些参数扮演着温度的角色。与金属退火原理相类似, 在开始阶段为了更快地最小化, 温度被升得很高, 然后才慢慢降温以求稳定。

3.5.1 模拟退火算法的运算流程

模拟退火算法的运算流程如图 3.4 所示。具体步骤如下：

模拟退火算法新解的产生和接受可分为如下三个步骤：

(1) 由一个产生函数从当前解产生一个位于解空间的新解；为便于后续的计算和接受，减少算法耗时，通常选择由当前解经过简单交换即可产生新解的方法。注意，产生新解的变换方法决定了当前新解的邻域结构，因而对冷却进度表的选取有一定的影响。

判断新解是否被接受，判断的依据是一个接受准则，最常用的接受准则是 Metropolis 准则：若 $\Delta E < 0$ ，则接受 X' 作为新的当前解 X ；否则，以概率 $\exp(-\Delta E/T)$ 接受 X' 作为新的当前解 X 。

(2) 当新解被确定接受时，用新解代替当前解，这只需将当前解中对应于产生新解时的变换部分予以实现，同时修正目标函数值即可。此时，当前解实现了一次迭代，可在此基础上开始下一轮试验。若当新解被判定为舍弃，则在原当前解的基础上继续下一轮试验。

模拟退火算法求得的解与初始解状态（算法迭代的起点）无关，具有渐近收敛性，已在理论上被证明是一种以概率 1 收敛于全局最优解的优化算法。模拟退火算法可以分解为解空间、目标函数和初始解三部分。该算法具体流程如下：

(1) 初始化：设置初始温度 T_0 （充分大）、初始解状态 X_0 （是算法迭代的起点）、每个 T 值的迭代次数 L ；

(2) 对 $k=1, \dots, L$ 进行第 (3) 至第 (6) 步操作，

(3) 产生新解 X' ；

(4) 计算增量 $\Delta E = E(X') - E(X)$ ，其中 $E(X)$ 为评价函数；

(5) 若 $\Delta E < 0$ ，则接受 X' 作为新的当前解，否则以概率 $\exp(-\Delta E/T)$ 接受 X' 作为新的当前解 X ；

(6) 如果满足终止条件，则输出当前解作为最优解，结束程序；

T 逐渐减小，且 $T \rightarrow 0$ ，然后转第 (2) 步。

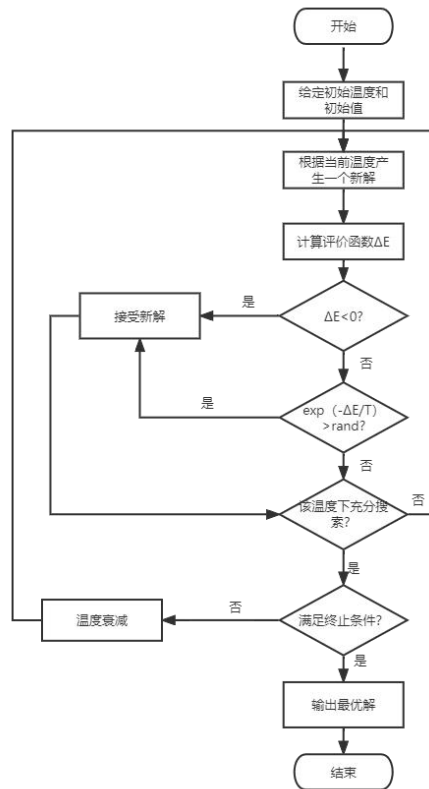


图 3.4 模拟退火算法的运算流程

3.5.2 模拟退火算法操作

模拟退火算法操作如下：

- (1) 初始化 Markov 链长度为 $L=200$, 衰减参数为 $K=0.998$, 步长因子为 $S=0.01$, 初始温度为 $T=100$, 容差为 $YZ=1 \times 10^{-8}$, 随机产生初始解, 并计算其目标函数值。
- (2) 在变量的取值范围内, 按步长因子随机产生新解, 并计算新目标两数值: 以 Metropolis 算法确定是否替代旧解, 在一种温度下, 迭代 L 次。
- (3) 判断是否满足终止条件: 若满足, 则结束搜索过程, 输出优化值: 若不满足, 则减小温度, 进行迭代优化。

3.5.3 模拟退火算法操作

(1) 模拟退火算法实验参数如表 3.4 所示：

表 3.4 模拟退火算法实验参数

变量维数 D	上限 X_s	下限 X_x
10	20	-20
马尔科夫链长度 L	衰减参数 K	步长因子 S
200	0.998	0.01
初始温度 T	容差 YZ	总接受点 P
100	1e-8	0

3.6 禁忌搜索算法基本流程

一个问题的求解过程就是搜索，它是人工智能的一个基本问题，目前人工智能在各应用领域中被广泛地使用。搜索技术渗透在各种人工智能系统中，可以说没有哪一种人工智能的应用不用搜索方法。禁忌搜索 (Tabu Search or Taboo Search,TS)算法的思想最早由美国工程院院上 Glover 教授于 1986 年提出，并在 1981 年 1990 年对该方法进行了进一回步的定义和改进。在自然计算的研究领域中，禁忌搜索算法以其灵活的存储结构和相应的禁忌准则来避免迂回搜索，在智能算法中独树一帜，成为一个研究热点，受到国内外学者的广泛关注。迄今为止，禁忌搜索算法在组合优化、生产调度、机器学习、电路设计和神经网络等领域取得了很大的成功,近年来又在函数全局优化方面得到较多的研究，并有迅速发展的趋势。

所谓禁忌，就是禁止重复前面的操作。为了改进局部邻域搜索容易陷入局部最优点的不足，禁忌搜索算法引入一个禁忌表，记录下已经搜索过的局部最优点，在下一次搜索中，对禁忌表中的信息不再搜索或有选择地搜索，以此来跳出局部最优点，从而最终实现全局优化。禁忌搜索算法是对局部邻域搜索的一种扩展，是一种全局邻域搜索、逐步寻优的算法^[11]。

禁忌搜索算法是一种迭代搜索算法，它区别于其他现代启发式算法的显著特点，是利用记忆来引导算法的搜索过程；它是对人类智力过程的一种模拟，是人工智能的一种体现。禁忌搜索算法涉及邻域、禁忌表、禁忌长度、候选解、藐视准则等概念，在邻域搜索的基础上，通过禁忌准则来避免重复搜索，并通过藐视准则来赦免一些被禁忌的优良状态，进而保证多多样化的有效搜索来最终实现全局。

3.6.1 禁忌搜索算法的运算流程

禁忌搜索算法的运算流程如图 3.5 所示。具体步骤如下：

禁忌搜索算法的基本思想是：给定一个当前解（初始解）和一种邻域，然后在当前解的邻域中确定若干候选解；若最佳候选解对应的目标值优于“*best so far*”状态，则忽视其禁忌特性，用它替代当前解和“*best so far*”状态，并将相应的对象加入禁忌表，同时修改禁忌表中各对象的任期；若不存在上述候选解，则在候选解中选择非禁忌的最佳状态为新的当前解，而无视它与当前解的优劣，同时将相应的对象加入禁忌表，并修改禁忌表中各对象的任期。如此重复上述迭代搜索过程，直至满足停止准则。其算法步骤可描述如下：

- （1）给定禁忌搜索算法参数，随机产生初始解 x ，置禁忌表为空。
- （2）判断算法终止条件是否满足：若满足，则结束算法并输出优化结果；否则，继续以下步骤。
- （3）利用当前解的邻域函数产生其所有（或若干）邻域解，并从中确定若干候选解。
- （4）对候选解判断藐视准则是否满足：若满足，则用满足藐视准则的最佳状态 y 替代 x 成为新的当前解，即 $x=y$ 了，并用与 y 对应的禁忌对象替换最早进入禁忌表的禁忌对象，同时用 y 替换“*best so far*”状态，然后转步骤（6）：否则，继续以下步骤。
- （5）判断候选解对应的各对象的禁忌属性，选择候选解集中非禁忌对象对应的最佳状态为新的当前解，同时用与之对应的禁忌对象替换最早进入禁忌表的禁忌对象。

判断算法终止条件是否满足：若满足，则结束算法并输出优化结果；否则，转步骤（3）。

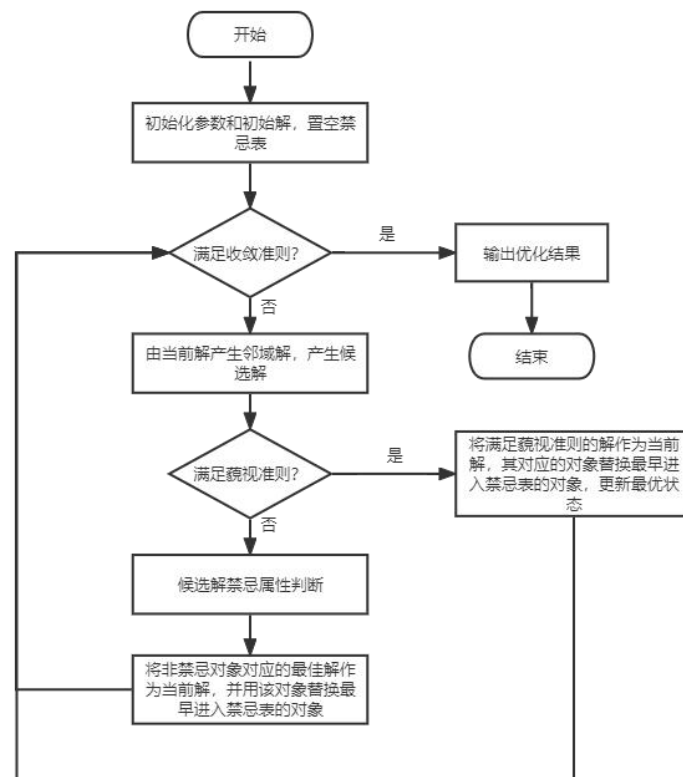


图 3.6 禁忌搜索算法的运算流程

3.6.2 禁忌搜索算法操作

禁忌搜索算法操作如下：

- （1）初始化优化城市规模 $N=31$ ，禁忌长度 $TabuL=22$ ，候选集的个数 $Ca=200$ ，最大迭代次数 $G=1000$ 。
- （2）计算任意两个城市的距离间隔矩阵 D ；随机产生一组路径为初始解 S_0 ，计算其适配值，并将其赋给当前最优解 $bestsofar$ 。
- （3）定义初始解的邻域映射为 $2-opt$ 形式，即初始解路径中的两个城市坐标进行对换。产生 Ca 个候选解，计算候选解的适配值，并保留前 $Ca/2$ 个最好候选解。
- （4）对候选解判断是否满足藐视准则：若满足，则用满足藐视准则的解替代初

始解成为新的当前最优解，并更新禁忌表 $Tabu$ 和禁忌长度 $TabuL$ ，然后转步骤（6）；否则，继续以下步骤。

（5）判断候选解对应的各对象的禁忌属性，选择候选解集中非禁忌对象所对应的最佳状态为新的当前解，同时更新禁忌表 $Tabu$ 和禁忌长度 $TabuL$ 。

（6）判断是否满足终止条件：若满足，则结束搜索过程，输出优化值；若不满足，则继续进行迭代优化。

3.6.3 禁忌搜索算法实验参数

（1）禁忌搜索算法实验参数如表 3.5 所示：

表 3.5 禁忌搜索算法实验参数

城市规模 N	禁忌长度 $TabuL$	候选集个数 Ca	最大迭代次数 G
31	22	200	1000

3.5 测试结果与分析

遗传算法适应度曲线如图 3.6 所示：

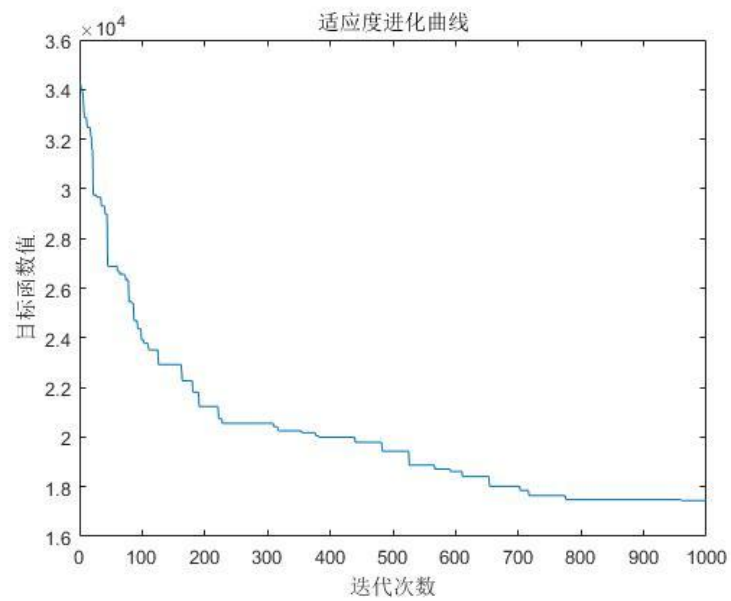


图 3.6 遗传算法旅行商问题适应度进化曲线

遗传算法旅行商问题路径图如图 3.7 所示：

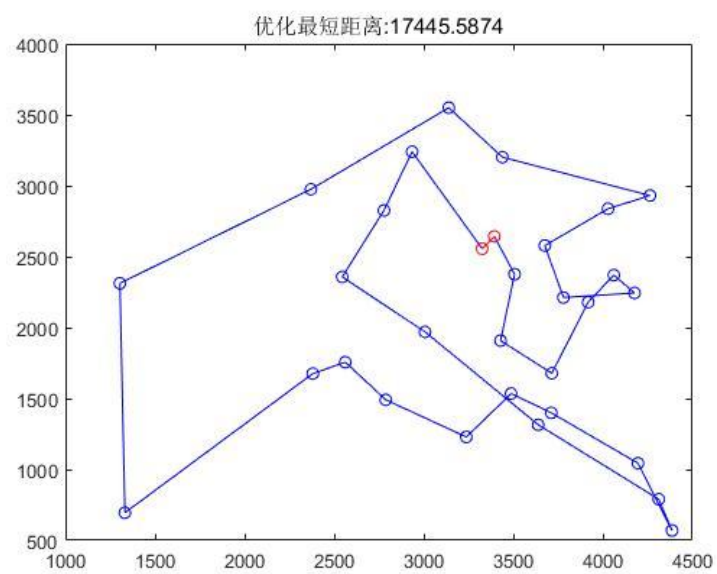


图 3.7 遗传算法旅行商问题路径图

免疫算法适应度曲线如图 3.8 所示：

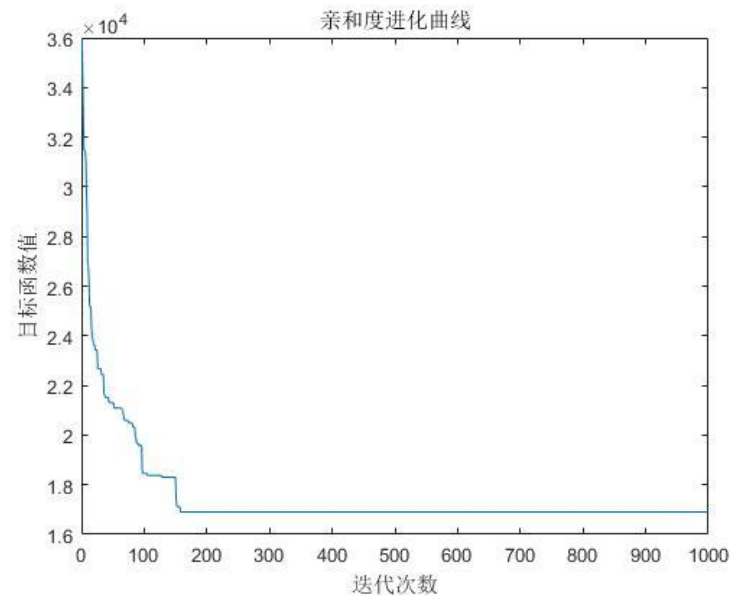


图 3.8 免疫算法旅行商问题亲和度进化曲线

免疫算法旅行商问题路径图如图 3.9 所示：

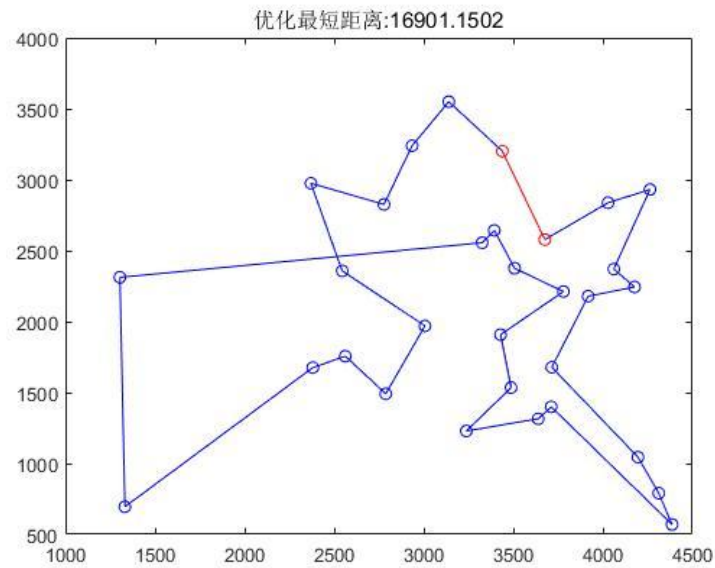


图 3.9 免疫算法旅行商问题路径图

蚁群算法适应度曲线如图 3.10 所示：

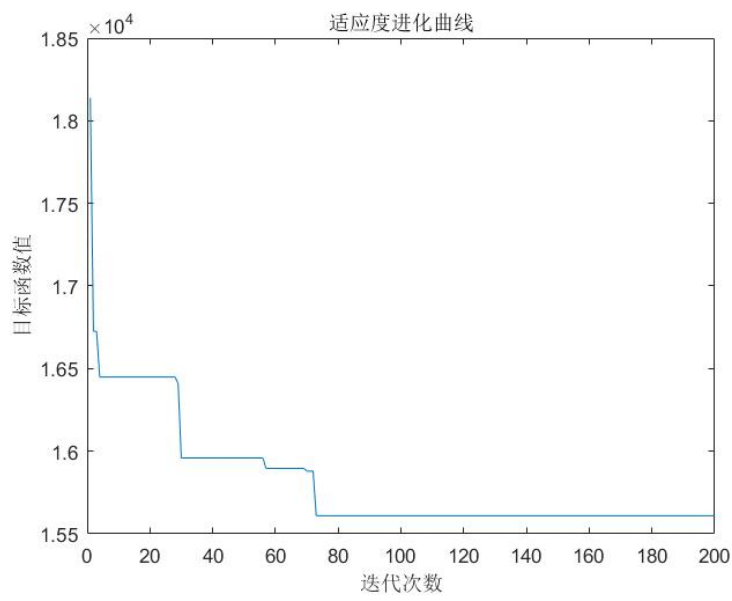


图 3.10 蚁群算法旅行商问题适应度进化曲线

蚁群算法旅行商问题路径图如图 3.11 所示：

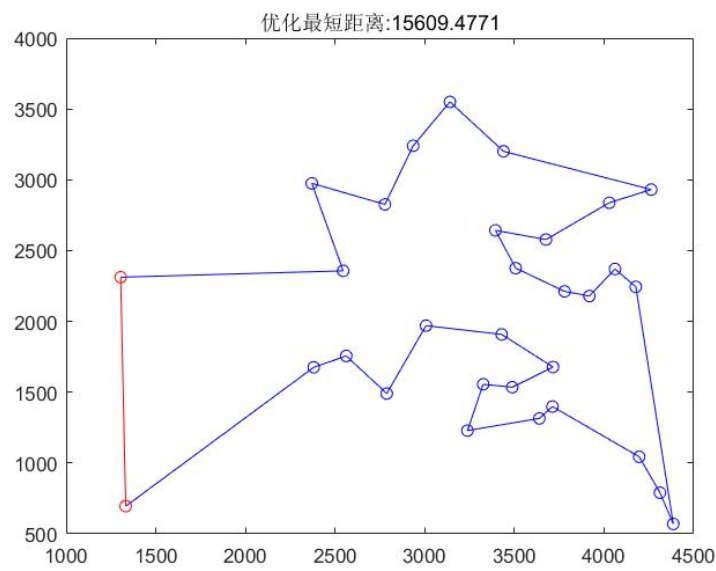


图 3.11 蚁群算法旅行商问题路径图

模拟退火算法适应度曲线如图 3.12 所示：

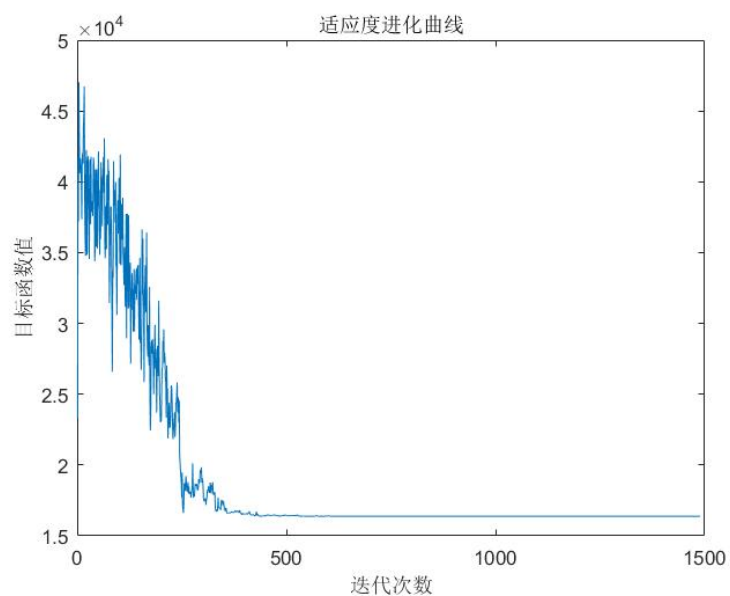


图 3.12 模拟退火算法旅行商问题适应度进化曲线

模拟退火算法旅行商问题路径图如图 3.13 所示：

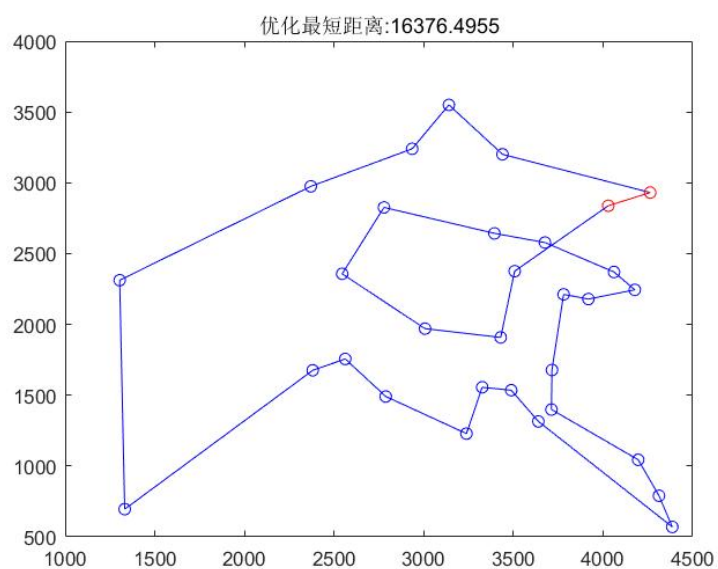


图 3.13 模拟退火算法旅行商问题路径图

禁忌搜索算法适应度曲线如图 3.14 所示：

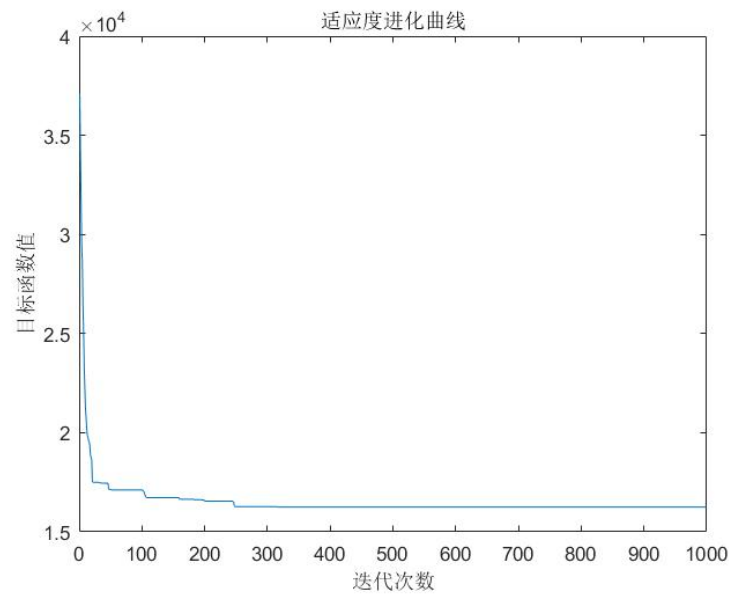


图 3.14 禁忌搜索算法旅行商问题适应度进化曲线

禁忌搜索算法旅行商问题路径图如图 3.15 所示：

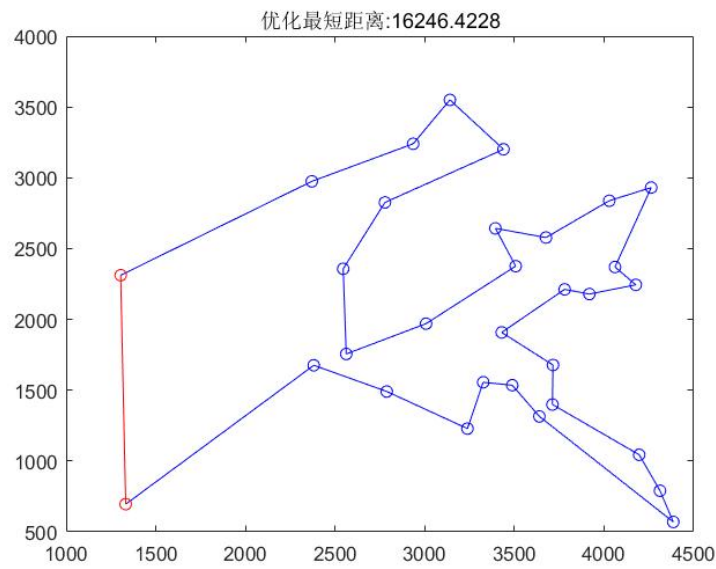


图 3.15 禁忌搜索算法旅行商问题路径图

通过多次独立实验，根据图示记录实验数据并进行分析，遗传算法、免疫算法、蚁群算法、模拟退火算法。禁忌搜索算法在求解旅行商问题时有以下的特点：

遗传算法：本次实验遗传算法的优化最短路径为 17445.5874，耗时 2S 左右，算法较快达到最优值且没有继续进行优化。查阅得知，遗传算法对路径的优化不稳定。在大量的独立实验中，多次出现了路径优化的不稳定现象，因此遗传算法求解旅行商问题等离散优化问题并非最优解。

免疫算法：本次实验免疫算法的优化最短路径为 16901.1052，耗时 5S，免疫算法较遗传算法求解时间略长，但最优值要优于遗传算法，通过适应度曲线可以看出，免疫算法的路径优化稳定，起伏较小，相较于遗传算法，免疫算法求解离散优化问题要更加适合。

蚁群算法：本次实验蚁群算法的优化最短路径为 15609.4771，耗时 16S，蚁群算法在三个算法中的优化值最好，但耗时最长。适应度进化曲线呈现阶梯状。如果需要选择最好优化值，应该选择蚁群算法。

模拟退火算法：本次实验模拟退火算法的优化最短路径为 16376.4966，耗时 63S，算法的迭代速度较慢，耗时较长。由适应度进化曲线可以看出，模拟退火算法在求解旅行商问题时的波动较大，上下起伏大。所以，模拟退火算法求解该问题并不是较优的方案。

禁忌搜索算法：本次实验禁忌搜索算法的优化最短路径为 16246.4228，耗时 38S，算法的耗时较长。由适应度进化曲线可以得出，算法较为稳定，但优化值不及蚁群算法，且耗时较长于蚁群算法。

综上所述，蚁群算法求解旅行商问题为上述算法中的较优方案。

4 总结与心得体会

4.1 总结

本次实验主要通过仿真测试求解连续函数最优值问题和旅行商问题来分别验证不同算法在连续函数和离散问题的求解能力。通过对算法的最优化函数值和优化最短路径以及成功率耗时等，来体现算法的求解能力。经过多次的独立实验，得出结果如下：

求解连续函数优化问题：粒子群算法为较优方案，粒子群中的粒子通过合作与竞争参与了问题的搜索优化，能够更加高效的进行函数收敛，所以其在相同函数评价次数的情况下能以较好的优化值和较少的耗时领先于其他算法。

旅行商问题：蚁群算法为较优方案，蚁群算法基于启发式的概率搜索，可以使得搜索不容易陷入局部最优，因此易于寻找最优解。蚁群算法是一种基于种群的进化算法，具有本质并行性，易于并行实现。因此，蚁群算法的路径优化值较好，耗时较短。

4.2 心得体会

通过本次的实验设计及撰写报告，我了解到了智能优化算法中各算法间的优劣特点，以及学会在不同的优化问题中使用不同的算法来进行优化，甚至可以将多种算法进行嵌套使用求解。在获取算法求解能力时，进行的测试应为多次实验，单次实验具有偶然性，而算法的初始化具有偶然性，所以必须进行多次独立实验来保证结果的正确。同时，需要将实验数据进行画表处理，使数据可视化。方便进行探索与了解。

本次实验书中只提供了基础的源代码，为进一步获得数据和更改实验参数来进一步实验，必须仔细阅读书籍和资料中的源代码，理解算法的思想与执行过程。本次实验，获益良多，除了了解到了智能优化算法中的诸如遗传算法、差分进化算法等知识，更提升了报告的撰写能力，使用 Origin 软件进行画图的相关操作。

参考文献

- [1] Holland John H.. Adaptation in Natural and Artificial Systems:An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence[M].The MIT Press:1992-04-29.
- [2] 梁旭, 黄明, 宁涛, 等. 现代智能优化混合算法及其应用 (第二版) [M]. 北京:电子工业出版社, 2014:373-391
- [3] Rainer Storn,Kenneth Price. Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces.[J]. Journal of Global Optimization,1997,11(4).
- [4] Dhanesh G. Kurup,Mohamed Himdi,Anders Rydberg. Synthesis of Uniform Amplitude Unequally Spaced Antenna Arrays Using the Differential Evolution Algorithm[J]. IEEE Transactions on Antennas and Propagation,2003,51(9).
- [5] Jerne N K. Towards a network theory of the immune system.[J]. Annales d"immunologie,1974,125C(1-2).
- [6] 孙宁. 人工免疫优化算法及其应用研究[D].哈尔滨工业大学,2006.
- [7] Dorigo M.,Maniezzo V.,Colorni A.. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics),1996,26(1).
- [8] 段海滨,马冠军,王道波,于秀芬.一种求解连续空间优化问题的改进蚁群算法[J].系统仿真学报,2007(05):974-977.
- [9] S. Kirkpatrick,C. D. Gelatt,M. P. Vecchi. Optimization by Simulated Annealing[J]. Science,1983,220(4598).
- [10] Xin Yao,Guojie Li. General simulated annealing[J]. Journal of Computer Science and Technology,1991,6(4).
- [11] 贺一. 禁忌搜索及其并行化研究[D].西南大学,2006.