日期： 2023 年 7 月 8 日

中山大學 SUN YAT-SEN UNIVERSITY

成绩： _____

学院：智能工程学院　　课程：最优化理论与方法　　周次：第 19 周

专业：智能科学与技术　　姓名：方桂安　　学号：20354027

1 题一

1.1 题目

编写 MATLAB 程序实现如下二次规划问题, 并写出 Lagrangian 函数和分析过程。

$$\begin{cases} \min & x_1^2 + x_2^2 \\ \text{s.t.} & x_1 + x_2 = 1 \\ & x_2 \leq \frac{1}{4} \end{cases} .$$

1.2 解答

算法分析:

这是一个二次规划问题, 我们可以通过拉格朗日乘数法来求解。在这个问题中, 目标函数是 $x_1^2+x_2^2$, 并且有两个约束条件：$x_1 + x_2 = 1$ 和 $x_2 \leq \frac{1}{4}$。我们需要先构建拉格朗日函数，然后求解相应的拉格朗日乘数。

拉格朗日函数可以写成如下形式：

$$L(x_1, x_2, \lambda, \mu) = x_1^2 + x_2^2 + \lambda(1 - x_1 - x_2) + \mu(\frac{1}{4} - x_2)$$

其中，λ 和 μ 是拉格朗日乘数，对应于两个约束条件。我们需要找到 L 的极小值点，这可以通过求解如下一阶条件得到：

$$\frac{\partial L}{\partial x_1} = 2x_1 - \lambda = 0$$
$$\frac{\partial L}{\partial x_2} = 2x_2 - \lambda - \mu = 0$$

$$\frac{\partial L}{\partial \lambda} = 1 - x_1 - x_2 = 0$$
$$\frac{\partial L}{\partial \mu} = \frac{1}{4} - x_2 = 0$$

这四个方程可以通过 MATLAB 的‘fmincon’函数求解。‘fmincon’是 MATLAB 中用于求解非线性约束优化问题的函数。

编程解答：

```
% 定义目标函数
fun = @(x) x(1)^2 + x(2)^2;

% 定义约束条件
A = []; b = []; % 无线性不等式约束
Aeq = [1 1]; beq = 1; % 线性等式约束 x1 + x2 = 1
lb = [-Inf, -Inf]; ub = [Inf, 1/4]; % x2 <= 1/4

% 定义初始值
x0 = [0, 0];

% 调用 fmincon 函数求解
options = optimoptions('fmincon','Display','iter','Algorithm','interior-point');
[x,fval] = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, [], options);

% 输出结果
disp('The solution is:'), disp(x)
disp('The minimum value of the objective function is:'), disp(fval)
```

最终结果为：**最小值点：(0.7500, 0.2500)；目标函数的值：0.6250**。

分析验证：

引入松弛变量 ε, 记 $x_2 + \varepsilon^2 = \frac{1}{4}$, 则问题改写为

$$\begin{cases} \min f(x_1, x_2) = x_1^2 + x_2^2 \\ \text{s.t. } g_1(x_1, x_2) = x_1 + x_2 - 1 = 0 \\ g_2(x_1, x_2, \varepsilon) = x_2 + \varepsilon^2 - \frac{1}{4} = 0 \end{cases}$$

Lagrangian 函数:

$$\mathcal{L}(x_1, x_2; \lambda_1, \lambda_2, \varepsilon) = f + \lambda_1 g_1 + \lambda_2 g_2$$

此时要求 $\lambda_2 \geq 0$

$$\frac{\partial \mathcal{L}}{\partial x_1} = 2x_1 + \lambda_1 = 0 \ldots (1)$$
$$\frac{\partial \mathcal{L}}{\partial x_2} = 2x_2 + \lambda_1 + \lambda_2 = 0 \ldots (2)$$
$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = x_1 + x_2 - 1 = 0 \ldots (3)$$
$$\frac{\partial \mathcal{L}}{\partial \lambda_2} = x_2 + \varepsilon^2 - \frac{1}{4} = 0 \ldots (4)$$
$$\frac{\partial \mathcal{L}}{\partial \varepsilon} = 2\varepsilon\lambda_2 = 0 \ldots (5)$$

对 (5) 式进行分类讨论:

$$\varepsilon = 0 :$$

由 (4) 得 $x_2 = \frac{1}{4}$, 则可以进一步解得 $x_1 = \frac{3}{4}, \lambda_2 = 1$ 满足 $\lambda_2 \geq 0$ 此时 $f(x_1, x_2) = \frac{5}{8}$

$$\lambda_2 = 0 :$$

对比 (1), (2) 可得 $x_2 = x_1$ 代入 (3), $x_{1,2} = \frac{1}{2}$, 此时 $f(x_1, x_2) = \frac{1}{2}$, 则 $\min f = \frac{1}{2}$。此时不满足约束条件：$x_2 \leq \frac{1}{4}$。

故 MATLAB 结果正确。

2 题二

2.1 题目

编写 MATLAB 程序, 利用外点罚函数法求解如下问题:

$$\begin{cases} \min \quad f(x) = x_1^2 + x_2^2 + x_3^2 \\ \text{s.t.} \quad x_1 + 2x_2 - x_3 = 4 \\ \qquad x_1 - x_2 + x_3 = -2 \end{cases} .$$

2.2 解答

算法分析:

外点罚函数法是一种处理约束优化问题的方法，它通过在目标函数中添加一个罚项，使得违反约束的解的函数值变大，因此优化算法会被引导向满足约束条件的解。

对于等式约束，我们的罚函数可以选择为约束违反程度的绝对值。所以我们可以定义一个罚函数如下:

$$P(x, r) = f(x) + r \cdot g(x)$$

其中，$f(x) = x_1^2 + x_2^2 + x_3^2$ 是目标函数，$g(x) = |x_1 + 2x_2 - x_3 - 4| + |x_1 - x_2 + x_3 + 2|$ 是约束条件，r 是一个大于 0 的常数，表示罚项的权重。

然后，我们可以使用 MATLAB 的优化函数，如‘fminunc’，来求解这个无约束优化问题。

编程解答：

```
% 定义目标函数和罚函数
fun = @(x) x(1)^2 + x(2)^2 + x(3)^2;
g = @(x) abs(x(1) + 2*x(2) - x(3) - 4) + abs(x(1) - x(2) + x(3) + 2);
P = @(x, r) fun(x) + r * g(x);

% 定义初始值和罚项权重
x0 = [0, 0, 0];
r = 1;

% 使用 fminunc 函数求解
options = optimoptions('fminunc','Display','iter','Algorithm','quasi-newton');
[x,fval] = fminunc(@(x) P(x, r), x0, options);

% 输出结果
disp('The solution is:'), disp(x)
disp('The minimum value of the objective function is:'), disp(fval)
```

最终结果为：**最小值点：(0, 1.2000, -0.8000)；目标函数的值：2.8800**。

分析验证：

1. 要最小化函数 $f(x) = x_1^2+x_2^2+x_3^2$，并且有两个约束条件：$x_1+2x_2-x_3 = 4$ 和 $x_1-x_2+x_3 = -2$。

2. 我们首先解这两个约束条件，得到 $x_2 = 2 - 2x_1$ 和 $x_3 = -3x_1$。这是一个参数解，其中 x_1 是任意实数。

3. 然后我们将 x_2 和 x_3 的解代入原函数 $f(x)$，得到一个只包含 x_1 的函数：$f(x) = x_1^2 + (2 - 2x_1)^2 + (-3x_1)^2$。

4. 我们找到这个函数的最小值，得到 $f(x)$ 的最小值为 20/7，此时 $x_1 = 2/7$。

5. 最后，我们将 $x_1 = 2/7$ 代入 $x_2 = 2 - 2x_1$ 和 $x_3 = -3x_1$，得到 $x_2 = 10/7$ 和 $x_3 = -6/7$。

所以，这个问题的解是 $x_1 = 2/7, x_2 = 10/7, x_3 = -6/7$，并且在这个解下，函数 $f(x) = x_1^2+x_2^2+x_3^2$ 的最小值为 20/7。

由于代数计算结果与 MATLAB 不一致，我决定使用几何分析求解：直线方程：$\frac{x_1-1}{1} = \frac{x_2}{-2} = \frac{x_3+3}{-3}$，过原点垂面 $x_1 - 2x_2 - 3x_3 = 0$，交点为 $\left(\frac{2}{7}, \frac{10}{7}, -\frac{6}{7}\right)$。显然这才是解析解，可以看出使用外点罚函数求解时为了获得解析解得定义 $r = inf$，但是答案会随 r 增大而收敛。

3 题三

3.1 题目

编写 MATLAB 程序, 采用遗传算法求如下函数在指定的约束集上的最大值:

$$f(x) = x + 5\sin(5x) + 10\cos(4x)$$

约束集为 $\Omega = \{x \in R : x \in [0, 10]\}$ 。

3.2 解答

算法分析:

遗传算法 (GA) 是一种基于模仿生物进化的自然选择过程求解无约束和有约束非线性优化问题。该算法反复修改由个体解构成的群体。在每个步骤，遗传算法从当前的群体随机选择个体，并将它们用作父级来生成下一代子级。经过一代又一代后，该群体“演化”为最优解。

遗传算法包括以下步骤：

1. 初始化种群：创建一组随机解作为初始种群。
2. 适应度评估：计算种群中每个解的适应度。
3. 选择：基于适应度选择解进行繁殖。
4. 交叉：随机选择种群中的解，并交换它们的部分以创建新的解。
5. 变异：以一定的概率随机改变解的部分。
6. 新的种群：新产生的解构成新的种群，返回步骤 2，直到达到预设的停止条件。

MATLAB 提供了一个名为‘ga’的函数，可以用于求解使用遗传算法的优化问题。在这个问题中，我们需要找到函数 $f(x) = x + 5\sin(5x) + 10\cos(4x)$ 在区间 [0, 10] 上的最大值。

注意，MATLAB 的‘ga’函数默认是求最小值的，所以我们需要将目标函数取反，使得求最大值问题转化为求最小值问题。

编程解答:

```
% 定义目标函数
fun = @(x) -(x + 5*sin(5*x) + 10*cos(4*x)); % 注意我们取负值，因为 MATLAB 的遗传算法默认是求最小值

% 定义遗传算法的参数
numberOfVariables = 1;

% 定义约束上下界
```

```
lb = 0;
ub = 10;

% 使用 ga 函数求解
[x,fval] = ga(fun,numberOfVariables,[],[],[],[],lb,ub);

% 输出结果
disp('The solution is:'), disp(x)
disp('The maximum value of the objective function is:'), disp(-fval)
```

最终结果为：**最大值点：(7.8575)；目标函数的值：22.8557**。

分析验证：

首先我进行化简求导：

$$
\begin{aligned}
\sin 5x &= \sin^5 x - 10\sin^3 x\cos^2 x + 5\sin x\cos^4 x \\
&= \sin^5 x + 10\sin^5 x - 10\sin^3 x + 5\sin^5 x - 10\sin^3 x + 5\sin x \\
&= 16\sin^5 x - 20\sin^3 x + 5\sin x \\
\cos 4x &= \cos^4 x - 6\cos^2 x\sin^2 x + \sin^4 x \\
&= \sin^4 x - 2\sin^2 x + 1 + 6\sin^4 x - 6\sin^2 x + \sin^4 x \\
&= 8\sin^4 x - 8\sin^2 x + 1 \\
f(x) &= x + 80\sin^5 x + 80\sin^4 x - 100\sin^3 x - 80\sin^2 x + 25\sin x + 10 \\
\text{令} t &= \sin x, x = \arcsin t, \text{ 考虑到周期性, 取} x = [2\pi, 10], \\
f(t) &= 2\pi + \arcsin t + 80t^5 + 80t^4 - 100t^3 - 80t^2 + 25t + 10 \\
f'(t) &= \frac{1}{\sqrt{1-t^2}} + 400t^4 + 320t^3 - 300t^2 - 160t + 25
\end{aligned}
$$

求导的结果较为复杂，我选择使用可视化的方法来判断结果正确与否，如图1所示。

从可视化的曲线中可以直观地看出 MATLAB 的结果正确。

4 题四

4.1 题目

编写 MATLAB 程序, 利用内点罚函数法求解问题:

$$
\begin{cases}
\min & f(x) = x_1^2 + x_2^2 \\
\text{s.t.} & x_1 - 2 \geq 0
\end{cases}.
$$

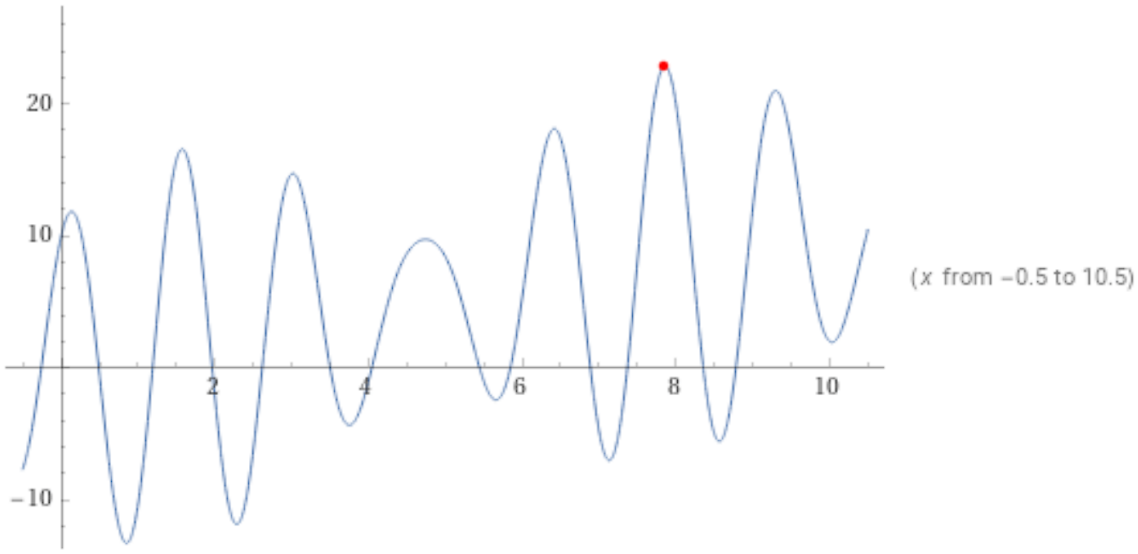

图 1: $f(x) = x + 5\sin(5x) + 10\cos(4x)$

4.2 解答

算法分析:

内点罚函数法是一种处理约束优化问题的方法，它通过在目标函数中添加一个罚项，使得违反约束的解的函数值变大，因此优化算法会被引导向满足约束条件的解。

对于不等式约束，我们的罚函数可以选择为约束违反程度的对数。所以我们可以定义一个罚函数如下:

$$P(x, r) = f(x) - r \cdot \ln(g(x))$$

其中，$f(x) = x_1^2 + x_2^2$ 是目标函数，$g(x) = x_1 - 2$ 是约束条件，r 是一个大于 0 的常数，表示罚项的权重。

然后，我们可以使用 MATLAB 的优化函数，如‘fminunc’，来求解这个无约束优化问题。

编程解答:

```
% 定义目标函数和罚函数
fun = @(x) x(1)^2 + x(2)^2;
g = @(x) x(1) - 2;
P = @(x, r) fun(x) - r * log(g(x));

% 定义初始值和罚项权重
x0 = [3, 0]; % 初始值需要满足约束条件
r = 0.0001;

```

```
% 使用 fminunc 函数求解
options = optimoptions('fminunc','Display','iter','Algorithm','quasi-newton');
[x,fval] = fminunc(@(x) P(x, r), x0, options);

% 输出结果
disp('The solution is:'), disp(x)
disp('The minimum value of the objective function is:'), disp(fval)
```

最终结果为：**最小值点：(2.000, 0)；目标函数的值：4.0012**。

分析验证：

1. 目标函数和约束条件：我们的目标函数是 $f(x) = x_1^2 + x_2^2$，这是一个二次函数，它在整个实数域内都是定义的。我们的约束条件是 $x_1 - 2 \geq 0$，这意味着 x_1 的取值范围是 $[2, +\infty)$。

2. 分析目标函数：函数 $f(x) = x_1^2 + x_2^2$ 是一个开口向上的抛物面，它在 $x_1 = x_2 = 0$ 处取得最小值。然而，由于我们的约束条件，x_1 的取值不能小于 2。因此，我们需要在 $x_1 \geq 2$ 的区域内找到函数的最小值。

3. 求解：在 $x_1 \geq 2$ 的区域内，函数 $f(x) = x_1^2 + x_2^2$ 的最小值出现在 $x_1 = 2, x_2 = 0$ 处。这是因为当 $x_1 = 2$ 时，我们可以将 x_2 设置为 0 来最小化 x_2^2。此时，函数的值为 $f(2, 0) = 2^2 + 0^2 = 4$。

所以，这个问题的解是 $x_1 = 2$，$x_2 = 0$，并且在这个解下，函数 $f(x) = x_1^2 + x_2^2$ 的最小值为 4。故 MATLAB 的结果在误差允许的范围内是正确的。

5 题五

5.1 题目

编写 MATLAB 程序, 实现模拟退火法。$X^{(k)} \in \Omega$ 的邻域定义为

$$N\left(X^{(k)}\right) = \left\{X : x_i^{(k)} - \alpha \leq x_i \leq x_i^{(k)} + \alpha\right\} \subset \Omega$$

其中 Ω 为约束集; $\alpha > 0$ 可以自行指定; 新点 X' 按照均匀分布原则在领域 $N\left(X^{(k)}\right)$ 中随机抽取。利用如下函数对算法进行测试, 并总结 α 变化时所产生的影响。

$$f(x) = \sum_{i=1}^{10} x_i^2$$

求解函数在约束集上 $\Omega = \{x \in R^{10} : x_i \in [-15, 15]\}$ 的极小点。

5.2 解答

算法分析：

模拟退火算法是一种启发式的全局优化方法，它模拟了固体在冷却过程中能量的最小化。它以一定的概率接受比当前解差的解，以跳出局部最优，从而在全局范围内寻找最优解。

模拟退火算法包括以下步骤：

1. 初始化：选择一个初始解和初始温度。
2. 重复以下步骤直到满足终止条件：
   (a) 在当前解的邻域中随机选择一个解。
   (b) 如果新解的目标函数值更小，接受这个新解。否则，以一定的概率接受新解，这个概率随着目标函数值的差距和温度的变化而变化。
   (c) 降低温度。

编程解答：

```
% 定义目标函数
fun = @(x) sum(x.^2);

% 定义参数
alphas = [0.001,0.1,0.5, 1, 2, 5]; % 邻域大小
T = 100; % 初始温度
T_min = 1e-3; % 最小温度
cooling_rate = 0.95; % 冷却率
max_iter = 100; % 每个温度下的最大迭代次数

% 定义约束条件
lb = -15 * ones(1, 10); % 下界
ub = 15 * ones(1, 10); % 上界

results = zeros(length(alphas), 2);

for a = 1:length(alphas)
    % 初始化解
    x = lb + (ub - lb) .* rand(1, 10);
    alpha = alphas(a);

    T_current = T;
    while T_current > T_min
        for i = 1:max_iter
            % 在邻域中随机生成新解
            x_new = x + alpha * (2*rand(1, 10) - 1);
            x_new = max(min(x_new, ub), lb); % 确保新解满足约束条件

            % 计算目标函数的改变量
            Δ_f = fun(x_new) - fun(x);

            % 如果新解更好，或者满足 Metropolis 准则，则接受新解
```

```
            if Δ_f < 0 || rand() < exp(-Δ_f / T_current)
                x = x_new;
            end
        end

        % 降低温度
        T_current = cooling_rate * T_current;
    end

    % 记录结果
    results(a, :) = [alpha, fun(x)];

    % 输出结果
    fprintf('For alpha = %f:\n', alpha);
    disp('The solution is:'), disp(x)
    disp('The minimum value of the objective function is:'), disp(fun(x))
end

% 可视化结果
figure;
semilogy(results(:, 1), results(:, 2), '-o', 'Color', [0.2 0.4 0.6], 'LineWidth', 2, 'MarkerSize', 8);
xlabel('Alpha');
ylabel('Minimum Value of Objective Function');
title('Impact of Alpha on the Solution');
```

最终结果为：**最小值点：(0.0300, 0.0261, 0.0287, 0.0065, 0.0200, 0.0380, -0.0379, 0.0063, -0.0018, -0.0489)；目标函数的值：0.0082**。此时选取的 α 为 0.1。

分析验证：

如图2所示，我通过多次实验来探究关于 α 的影响。由于第一个值远大于其他值，所以这里对 y 轴进行对数变换，方便可视化。可以看出 α 控制了搜索邻域的大小。较大的 α 可以使算法在每一步探索更大的区域，可能有助于跳出局部最优解，但也可能导致算法在全局最优解附近震荡而无法收敛，如图中最后一点的最小值为 14.3685 显然是错误的。较小的 α 可以使算法更精确地搜索，但可能会陷入局部最优解，如图中第一个点的最小值为 914.5227 显然也是错误的。因此，选择合适的 α 是很重要的。在实际使用时，可能需要多次试验来找到最佳的 α。

本题是一个多元函数的优化问题，我们需要找到函数 $f(x) = \sum_{i=1}^{10} x_i^2$ 在约束集 $\Omega = \{x \in R^{10} : x_i \in [-15, 15]\}$ 上的极小点。

这个函数是一个二次函数，它在整个实数域上都是凸的，所以它的全局最小值就是它的极小值。对于这个函数，我们可以通过求导并令导数等于零来找到极小值点。

对于函数 $f(x) = \sum_{i=1}^{10} x_i^2$，我们有：

$$\frac{\partial f}{\partial x_i} = 2x_i$$

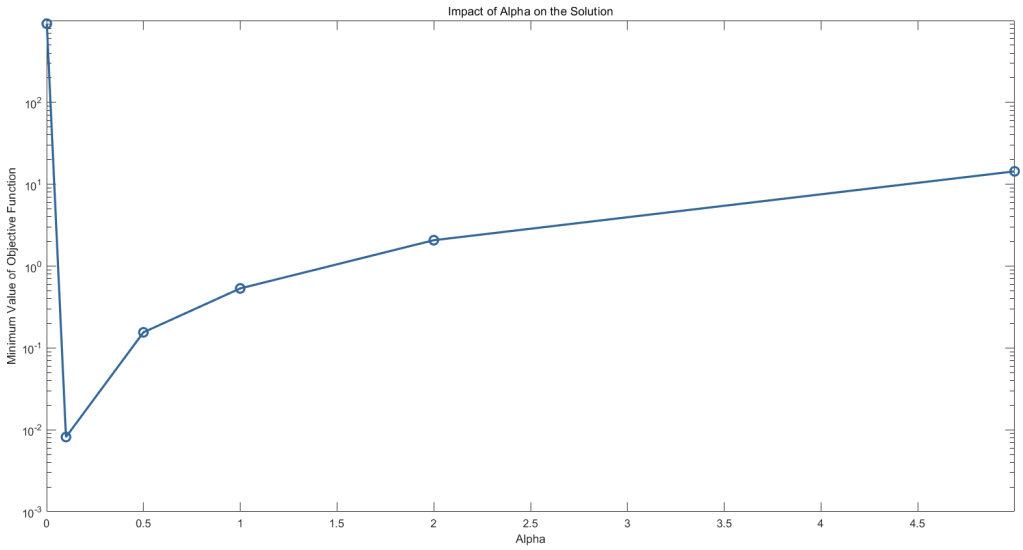

图 2: α 对函数最小值求解的影响

令 $\frac{\partial f}{\partial x_i} = 0$，我们得到 $x_i = 0$，对于所有 i。

所以，函数 $f(x) = \sum_{i=1}^{10} x_i^2$ 的极小值点是 $x = (0, 0, ..., 0)$。这个点也在约束集 $\Omega = \{x \in R^{10} : x_i \in [-15, 15]\}$ 中，所以这就是我们的解。故 MATLAB 的结果在误差允许的范围内是正确的。

6 题六

6.1 题目

后半学期课程总结，包括：

1. 知识点总结与分析；
2. 个人角度的课程难点；

6.2 知识点总结与分析

首先对于知识点总结与分析，我采取思维导图的方式展示，将我平常在课上所做的 markdown 笔记利用 markmap 转换成 pdf，展示如图3。当然由于篇幅限制，这里的观看效果较差，老师可以通过我的博客来查阅。

在后半学期里，我们学习了坐标轮换法、约束优化最优性条件、二次规划、罚函数与方法、现代优化算法。下面是对这些知识点的总结与分析：

坐标轮换法：这种方法是最优化算法的一种，它通过在每一步中固定除一个坐标外的所有其他坐标，然后针对这个坐标进行优化，以求解复杂的优化问题。这种方法的主要优点是其实现简单，并且对于某些问题，可以保证收敛到全局最优。

约束优化最优性条件：在约束优化问题中，最优性条件是用来判断一个解是否可能是最优解的重要工具。在这里，我们主要学习了 KKT 条件，包括可行性条件、对偶可行性条件和互补松弛条件。我们也学习了如何使用拉格朗日乘数来将约束条件融合到目标函数中。

二次规划：二次规划是优化理论中的重要分支，它涉及的问题是在一些约束条件下，最小化一个二次目标函数。这种问题在许多应用中都很重要，包括投资组合优化、支持向量机等。

罚函数与方法：罚函数方法是一种用于解决约束优化问题的方法，它通过在目标函数中添加一个与约束违反程度相关的罚项，将约束优化问题转化为无约束优化问题。我们学习了两种主要的罚函数：外点罚函数和内点罚函数。

现代优化算法：我们还学习了两种重要的现代优化算法：模拟退火算法和遗传算法。模拟退火算法是一种概率搜索算法，它通过模拟固体冷却过程，逐渐接近全局最优解。而遗传算法则是通过模拟自然选择的过程，通过种群选择、交叉和变异操作，来搜索最优解。

通过学习这些主题，我们对最优化理论与方法有了更深入的理解。在实际问题中，我们可以根据问题的特点和需求，选择合适的优化方法来求解。

6.3 个人角度的课程难点

在学习最优化理论与方法的过程中，我逐渐理解了它不仅仅是一个纯理论的数学工具，而是一个应用广泛的解决实际问题的有效手段，特别是在我参加数学建模竞赛中起到了很大的帮助。以下是我在学习过程中的一些深入理解和尝试的应用：

1. 最优化理论对于建模的重要性：在实际问题中，我们往往需要将问题抽象为一个最优化模型，通过对模型的求解，找到问题的最佳答案。在这一过程中，选择合适的目标函数和约束条件是非常重要的。例如，在一次物流优化问题中，我选择了运输时间作为目标函数，通过最小化总的运输时间，成功找到了最佳的物流方案。

2. 对优化算法的深入理解：我尝试通过深入理解各种优化算法的原理和特点，提高选择和应用算法的能力。例如，我了解到模拟退火算法在解决带有多个局部最优解的问题时具有较好的性能，因为其能够通过一定概率接受次优解，以避免陷入局部最优。

3. 在实际问题中尝试应用优化算法：我尝试将所学的优化算法应用到实际问题中。例如，在一次设备调度问题中，我尝试使用遗传算法进行求解，通过设置适当的适应度函数和遗传操作，成功找到了较优的调度方案。虽然在调参过程中遇到了一些困难，但通过反复试验和学习，我逐渐理解了如何设置合适的参数。

4. 深入理解罚函数方法：在处理约束优化问题时，罚函数方法是一个重要的工具。我通过深入学习和实践，理解了如何通过设置罚项，将约束优化问题转化为无约束优化问题，以便使用更多的优化算法进行求解。在解决一次有约束的资源分配问题时，我成功应用了罚函数方法，得到了满足所有约束的优化方案。

总的来说，最优化理论与方法课程的学习让我深刻理解了最优化问题的复杂性以及解决这类问题的方法。虽然我在学习过程中遇到了一些难点，但通过努力学习和实践，我已经取得了一些成果，并对未来的学习和工作充满了信心。

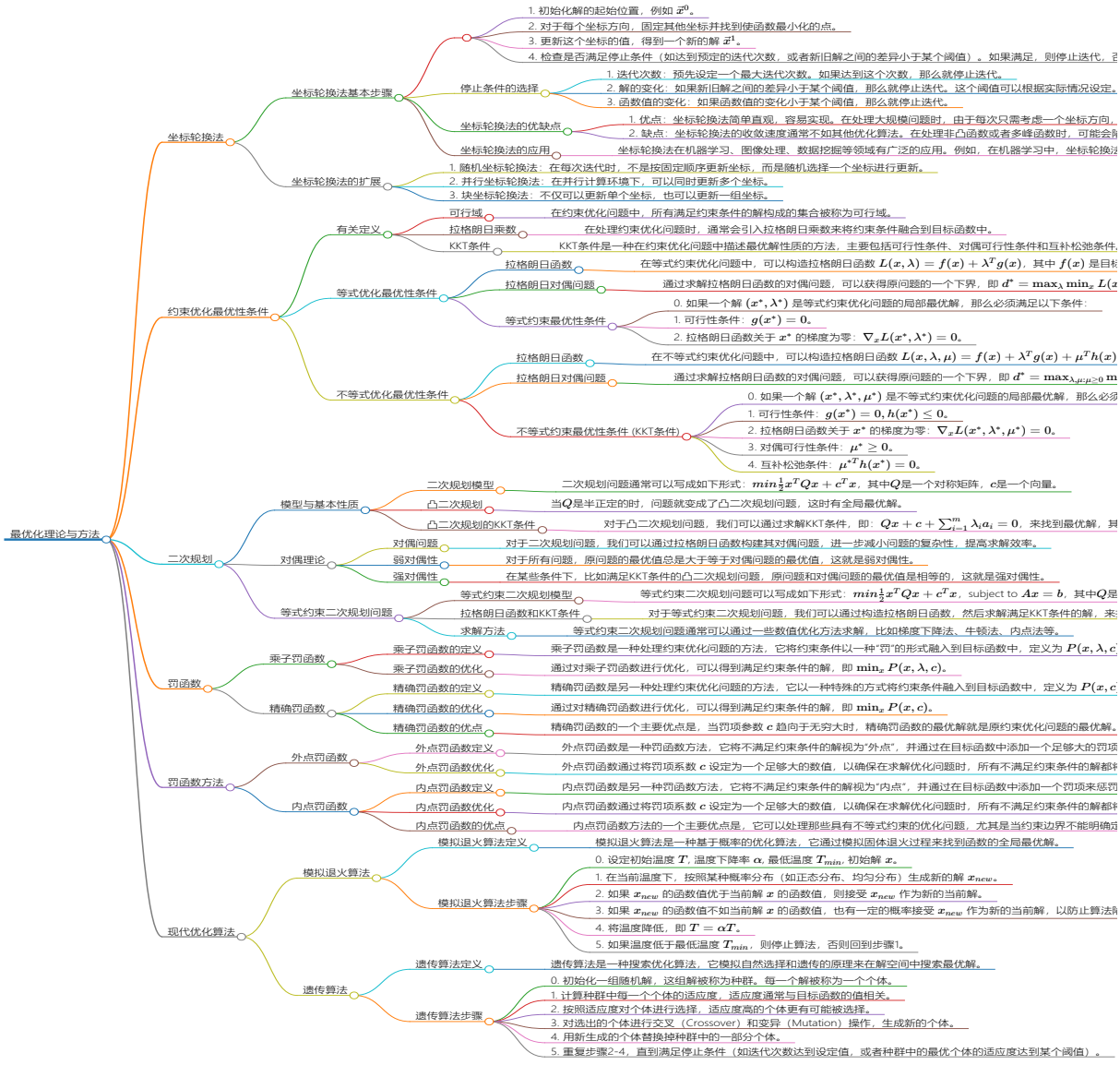

图 3: 最优化期末知识思维导图