

烟幕干扰弹的投放策略解题报告

小组成员：周陈序 杨仕言 陈金铭 邓伊涵

指导老师：吴瑞佳

一、问题背景

本问题研究如何利用具有长续航能力的无人机，通过投放烟幕干扰弹，设计最佳投放策略，以最大化对固定保护目标（圆柱体）的有效遮蔽时间，从而干扰来袭的空地导弹。

关键设定与参数：

1. 目标与坐标系：

- 假目标（原点）： $O = (0, 0, 0)$ 。
- 真目标（圆柱体）：底面中心 $T_{base} = (0, 200, 0)$ ，半径 $R_T = 7\text{ m}$ ，高 $H_T = 10\text{ m}$ 。

2. 来袭导弹 (M)：

- 速度 $V_M = 300\text{ m/s}$ ，沿直线匀速飞向假目标 O 。
- 初始位置： $M_1(20000, 0, 2000)$ ， $M_2(19000, 600, 2100)$ ， $M_3(18000, -600, 1900)$ 。

3. 无人机 (FY)：

- 受领任务后可瞬时调整方向，以 $V_{UAV} \in [70, 140]\text{ m/s}$ 等高匀速直线飞行。
- 初始位置：5架无人机 FY1-FY5 已知。
- 投放约束：同一架无人机投放两枚烟幕弹至少间隔 1 s 。

4. 烟幕云团 (S)：

- 脱离无人机后仅受重力（抛体运动）。
- 起爆后瞬时形成球状云团，有效半径 $R_{eff} = 10\text{ m}$ 。
- 云团中心以 $V_{sink} = 3\text{ m/s}$ 匀速下沉。
- 有效持续时间 $T_{eff} = 20\text{ s}$ 。

5. 遮蔽判定：

- 有效遮蔽定义：导弹视线与真目标之间的连线被烟幕云团完全遮蔽。在实际模型中，通常转化为判定导弹到目标圆柱体上所有视线是否被烟幕球体阻断（或达到某一高比例阈值）。

6. 优化目标：

- 设计无人机航向、速度、投放点、起爆点等参数，使多枚烟幕弹对真目标的有效遮蔽总时长（时间并集测度）尽可能长。

二、物理建模

2.1 运动学模型

1. 导弹运动 $P_M(t)$ ：

导弹从初始位置 P_{M0} 以恒定速度 V_M 飞向假目标 $P_O = (0, 0, 0)$ 。

$$\text{单位方向向量 } \mathbf{D}_M = \frac{P_O - P_{M0}}{\|P_O - P_{M0}\|}$$

导弹位置：

$$P_M(t) = P_{M0} + V_M \cdot t \cdot \mathbf{D}_M$$

对应函数： `get_MissilePos.m`

2. 无人机运动 $P_{UAV}(t)$ ：

无人机以恒定速度 V_{UAV} 和航向角 θ 等高飞行。

$$\text{水平速度向量 } \mathbf{V}_{UAV} = V_{UAV} \cdot (\cos \theta, \sin \theta, 0)$$

无人机位置：

$$P_{UAV}(t) = P_{UAV0} + \mathbf{V}_{UAV} \cdot t$$

3. 烟幕弹抛体运动 P_{Det} (投放到起爆)：

烟幕弹在 t_{drop} 投放，在 $t_{det} = t_{drop} + \tau$ 时刻起爆。

起爆点位置 P_{Det} :

$$P_{Det} = P_{Drop} + \mathbf{V}_{UAV} \cdot \tau + \frac{1}{2} \mathbf{g} \cdot \tau^2$$

其中 $\mathbf{g} = (0, 0, -9.8) \text{ m/s}^2$ 。

对应函数: `get_SmokeDetPos.m`

4. 烟幕云团中心动态 $C(t)$:

云团在 t_{det} 形成, 随后以 $\mathbf{V}_{sink} = (0, 0, -3) \text{ m/s}$ 匀速下沉。

云团中心位置 (当 $t \geq t_{det}$) :

$$C(t) = P_{Det} + (t - t_{det}) \cdot \mathbf{V}_{sink}$$

对应函数: `get_Smokecenter.m`

2.2 几何遮蔽模型

简化判定:

将真目标简化为一个锚点 $T_{base} = (0, 200, 0)$, 判定线段 $\overline{P_M T_{base}}$ 是否与烟幕球体 $S(C, R_{eff})$ 相交。遮蔽条件为球心 C 到线段 $\overline{P_M T_{base}}$ 的最近距离 $d_{min} \leq R_{eff}$ 。

精确判定 (最终采用的策略):

为了处理圆柱形目标, 模型采用目标采样点和多烟幕并集判定:

1. 目标采样: 在圆柱体表面和轴线生成一组离散采样点 \mathcal{P}_T (如 `is_FullyShielded.m` 中所示)。
2. 多烟幕并集判定: 检查导弹到每个采样点 $P_T \in \mathcal{P}_T$ 的射线是否与任一当前有效的烟幕球体 S_j 相交。
3. 有效遮蔽: 若被遮蔽的采样点比例达到 100% (或高阈值, 如 95%), 则判定为有效遮蔽。

对应函数: `is_FullyShielded.m`

三、 问题解答

3.1 问题一： 单机单弹， 固定参数计算时长

优化问题建模

本问题为确定性计算问题，目标是计算给定参数下的遮蔽持续时间 L 。

$$L = \int_{t_{eff}^{start}}^{t_{eff}^{end}} \mathbb{I}(d_{min}(t) \leq R_{eff}) dt$$

求解方案与代码实现

求解方案：

采用时间步扫描与几何判定法进行数值积分。由于遮蔽判定函数非线性，难以求解析解，故在烟幕有效时间窗 $[5.1 \text{ s}, 25.1 \text{ s}]$ 内，以固定时间步长 dt 对时间轴进行扫描。在每个时间步 t ，计算导弹位置 $P_M(t)$ 和云团中心 $C(t)$ ，并调用精确的几何判定函数 (`is_FullyShielded`) 判定遮蔽是否成立，最终累加有效时间步长。

代码实现描述 (`Problem1.m`):

- 时间步长： 设置扫描步长 $dt = 0.1 \text{ s}$ 。
- 动态位置计算： 调用 `get_SmokeDetPos` 计算起爆点；在循环中调用 `get_MissilePos` 和 `get_Smokecenter` 获取动态位置。
- 遮蔽判定： 调用 `is_FullyShielded` (精确判定版本)，该函数通过目标圆柱体采样或优化方法，判定导弹视线是否被云团阻断。
- 结果累加： 累加有效遮蔽时间 $Total_Shielded_Time$ 。

结果： 1.41s

3.2 问题二： 单机单弹， 参数优化最大化时长

优化问题建模

这是一个非光滑、非凸的连续优化问题，目标是选择最优的无人机飞行策略，最大化 M1 的有效遮蔽时间。

决策变量 \mathbf{x} (4维):

$$\mathbf{x} = [\theta, V_{UAV}, t_{drop}, \tau]$$

目标函数：最小化负的遮蔽时长 $F(\mathbf{x}) = -L(\mathbf{x})$ 。

$$F(\mathbf{x}) = -\max_{\mathbf{x}} \left(\int_{t_{drop}+\tau}^{t_{drop}+\tau+T_{eff}} \mathbb{I}(\text{Shielded}(t, \mathbf{x})) dt \right)$$

约束条件：

$$70 \leq V_{UAV} \leq 140 \text{ m/s}$$

$$t_{drop} \geq 0 \text{ s}, \quad \tau \geq 0 \text{ s}$$

求解方案与代码实现

求解方案：采用粒子群优化 (**PSO**) 算法进行全局搜索。由于目标函数 $F(\mathbf{x})$ 不可导，传统梯度法不适用，PSO 的全局探索能力更适合解决此类黑盒、非凸优化问题。

1. **PSO 结构**：粒子（候选解）在 4 维空间中移动，更新速度时学习个体最优经验和群体最优经验。
2. 目标函数评估：目标函数 `Calculate_Shielding_Time.m` 通过时间步扫描精确计算单弹遮蔽时长，并将其负值作为 PSO 的适应度。
3. 多轮优化：运行多轮 PSO 搜索，以避免陷入局部最优，选取最优结果。

代码实现描述 (`Problem2.m`, `run_simple_pso.m`, `Calculate_Shielding_Time.m`):

- 目标函数 (`Calculate_Shielding_Time.m`): 封装了问题一的数值计算逻辑，输出单烟幕弹对 M1 的总遮蔽时长 L ，主程序将其取负值传入 PSO。
- 优化器 (`run_simple_pso.m`): 实现了带动态惯性权重和平移速度钳制的经典 PSO 算法。
- 主程序 (`Problem2.m`): 定义变量边界和 PSO 参数，循环调用 `run_simple_pso` 进行多轮搜索，并记录全局最佳策略。

结果:4.60s

参数	值
航向角(度)	4.40
飞行速度(m/s)	90.43
投放时刻(s)	1.4438

参数	值
起爆延时(s)	0.0000

3.3 问题三：单机三弹，参数优化最大化并集时长

优化问题建模

问题三是单架无人机投放 3 枚烟幕弹的优化问题，目标是最大化遮蔽时间的并集测度。

决策变量 \mathbf{x} (8维):

$$\mathbf{x} = [\theta, V_{UAV}, t_{d,1}, t_{d,2}, t_{d,3}, \tau_1, \tau_2, \tau_3]$$

目标函数：最小化负的并集遮蔽时长 $F(\mathbf{x}) = -L_{union}(\mathbf{x})$ 。

$$L_{union}(\mathbf{x}) = \text{measure}(\{t \mid \exists j \in \{1, 2, 3\}, \text{Shielded}_j(t, \mathbf{x})\})$$

约束条件（含线性不等式约束）：

1. 速度/延时/航向约束： $\text{lb} \leq \mathbf{x} \leq \text{ub}$ (边界约束)

2. 投放间隔约束：

$$\begin{cases} t_{d,2} - t_{d,1} \geq 1 \text{ s} \\ t_{d,3} - t_{d,2} \geq 1 \text{ s} \end{cases}$$

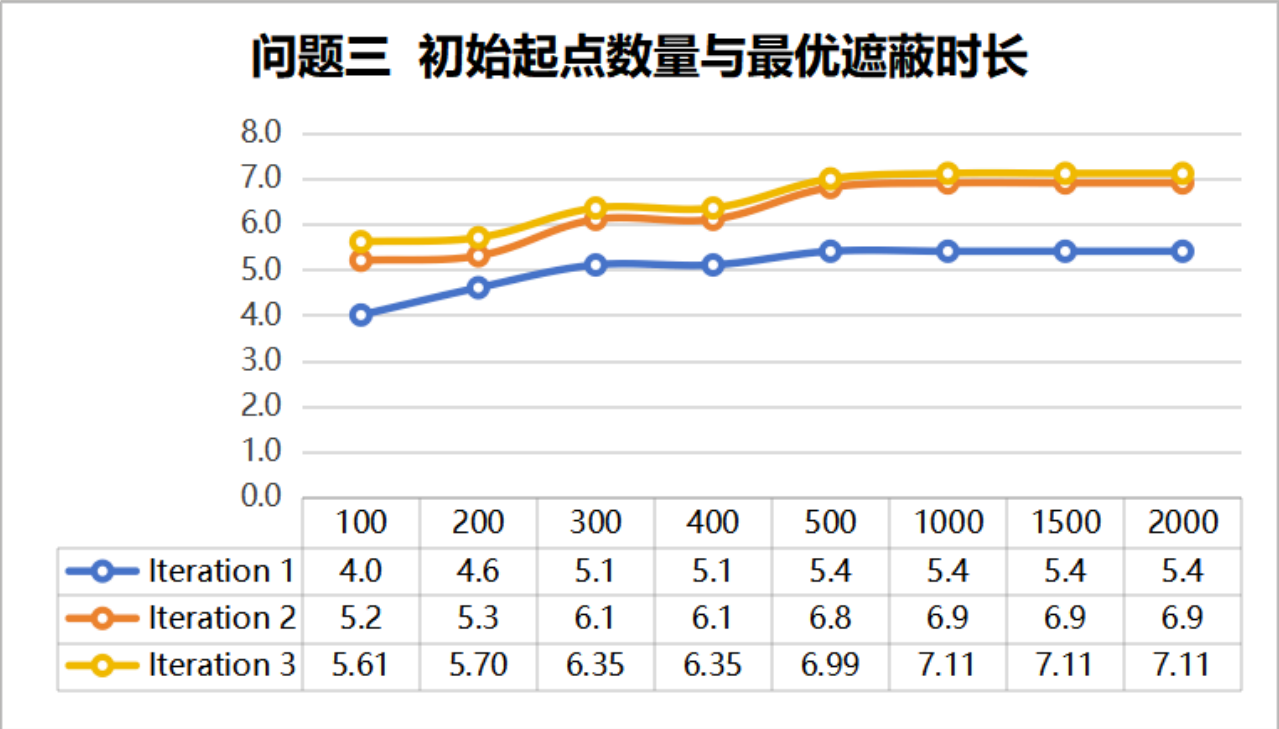
求解方案与代码实现

求解方案：采用多起点网格搜索与 **SQP** 局部精修（迭代细化搜索）的混合策略。首先通过粗网格搜索找到有潜力的解，然后利用高效的局部优化算法进行收敛。

- 多起点生成：通过多维网格划分搜索空间，生成满足线性间隔约束的初始点集。
- 局部优化 (**SQP**)：使用 MATLAB 的 **fmincon** 优化器，配置 **SQP**（序列二次规划）算法，对每个起点进行局部精修。**SQP** 能够高效处理带有约束的连续非线性优化问题。
- 并集评估：目标函数 **compute_shielding_duration_3smoke** 负责计算所有烟幕事件，并对时间轴进行扫描，在每一步判断是否有任一烟幕遮蔽导弹视线，从而稳健计算并集时长。

代码实现描述 (**Problem3.m**):

- 约束定义：使用 $Ax \leq b$ 形式定义投放间隔的线性约束。
- 迭代循环：主程序进行多轮迭代细化搜索，根据上一次的最优解缩小搜索范围。
- `fmincon` 调用：使用 'Algorithm', 'sqp' 配置，将目标函数 `objective_function_3smoke`（返回负时长）和约束 A, b, lb, ub 传入。
- 目标函数：`compute_shielding_duration_3smoke` 内部实现烟幕事件解码、运动学计算，并通过时间扫描和多烟幕并集判定逻辑计算总时长。



结果

参数/指标		值	
最优遮蔽时长(秒)		7.11	
航向角(度)		179.7	
飞行速度(m/s)		140.0	

烟幕弹 ID	投放时刻 (t_{drop})	起爆延时 (τ)	实际起爆时刻 (t_{det})
1	0.25	3.62	3.87
2	3.58	5.38	8.96
3	5.42	6.00	11.42

3.4 问题四：三机三弹，参数优化最大化并集时长

优化问题建模

问题四是 3 架无人机各投放 1 枚烟幕弹协同干扰 M1 的优化问题。

决策变量 \mathbf{x} (12维):

$$\mathbf{x} = [\theta_1, V_1, t_{d,1}, \tau_1, \theta_2, V_2, t_{d,2}, \tau_2, \theta_3, V_3, t_{d,3}, \tau_3]$$

目标函数：最小化负的并集遮蔽时长 $F(\mathbf{x}) = -L_{union}(\mathbf{x})$ 。

$$L_{union}(\mathbf{x}) = \text{measure}(\{t \mid \exists j \in \{1, 2, 3\}, \text{Shielded}_j(t, \mathbf{x})\})$$

约束条件：

$$\text{lb} \leq \mathbf{x} \leq \text{ub}$$

求解方案与代码实现

求解方案：采用遗传算法 (GA) 进行全局优化。GA 的搜索能力在高维非凸空间中优于传统的局部优化方法，适合处理这种多机协同、高维度的复杂问题。

1. GA 适应度：采用目标函数 `calculate_multiple_UAVs_shielding_time_new` (返回时长)，将负值作为 GA 的适应度函数进行最小化。
2. 精确判定：目标函数的评估器内部采用目标圆柱体的多点采样，并使用 `isObscuredByMultipleSmokes` 进行精确的多烟幕并集遮蔽判定，确保对圆柱体目标的覆盖效果。

代码实现描述 (Problem4.m):

- 主程序 (Problem4.m): 定义 12 维决策变量的边界 lb, ub。配置 GA 参数 (如 `MaxGenerations` 和 `PopulationSize`)，并调用 `ga` 函数。
- 目标函数 (`calculate_multiple_UAVs_shielding_time_new`): 解包 12 维变量，计算 3 枚烟幕弹的起爆事件。在时间扫描中，收集当前有效的烟幕中心，并调用 `isObscuredByMultipleSmokes` 进行遮蔽判定。
- 判定核心函数 (`isObscuredByMultipleSmokes`): 实现了 95% 遮蔽比例阈值的判定逻辑，依赖子函数 `generateCylinderSamplingPoints` (生成采样点) 和 `isRayBlockedBySphere` (射线-球体相交判定)。

结果: 11.29s

无人机 ID	飞行方向角 (度 / 弧度)	飞行速度 (M/S)	烟幕投放时间 (S)	起爆延时 (S)	烟幕起爆时间 (S)
FY1	5.12 / 0.0894	100.67	0.734	0.455	1.190
FY2	237.36 / 4.1427	81.95	12.303	7.533	19.836
FY3	107.64 / 1.8787	97.09	27.194	6.249	33.443

3.5 问题五：五机多弹，协同干扰三枚导弹

方案一：多起点全局优化 (Problem5.m)

优化问题建模

总共 5 架无人机，每架最多 3 弹，共 15 枚烟幕弹，干扰 3 枚导弹 (M1, M2, M3)。

决策变量 \mathbf{x} (40维): 5 架无人机 \times 8 个参数 ($\theta, V, t_{d,1}, \Delta t_{d,2}, \Delta t_{d,3}, \tau_1, \tau_2, \tau_3$)。

目标函数：最小化负的总有效遮蔽时长之和 $F(\mathbf{x}) = -L_{total}(\mathbf{x})$ 。

$$L_{total}(\mathbf{x}) = \sum_{k=1}^3 L_k(\mathbf{x})$$

其中 L_k 是 15 枚烟幕对导弹 M_k 的遮蔽时间并集的测度。

求解方案与代码实现

求解方案：针对 40 维的高维非凸空间，采用启发式混合优化策略：随机启发式初始化 \rightarrow **PSO** 全局搜索 \rightarrow **SQP** 局部精修 \rightarrow 多起点并行搜索。

1. 启发式初始化 (**Heuristic_Init_Randomized.m**): 生成部分基于几何先验（例如，让无人机随机瞄准某个导弹的预定拦截点）的初始解，以引导搜索方向。
2. 改进 **PSO** 全局搜索：使用 PSO (**run_improved_pso_silent**) 对高维空间进行初步全局探索，定位潜在最优区域。
3. **SQP** 局部精修：以 PSO 结果为初值，使用 **fmincon** 搭配 SQP 算法进行精确、快速的收敛。
4. 多起点并行搜索：重复上述步骤多次 (**Num_Restarts**)，并使用 **parfor** 进行

并行计算（见 `Problem5.m`），以提高找到全局最优解的概率。

代码实现描述 (`Problem5.m`, `Obj_Problem5_MultiTarget.m`):

- 目标函数 (`Obj_Problem5_MultiTarget.m`): 输入 40 维 \mathbf{x} ，解码 15 枚烟幕事件。对 M_1, M_2, M_3 分别进行时间扫描，计算各自的并集遮蔽时长 L_k ，最后返回 $-\sum L_k$ 。其中多烟幕对单导弹的判定逻辑依赖于 `is_FullyShielded_Fast` 或类似的并集判定。
- 主程序 (`Problem5.m`): 实现了多起点循环和结果记录逻辑，依次调用 `Heuristic_Init_Randomized`、`run_improved_pso_silent` 和 `fmincon`。

结果 (方案一): 11.2s

UAV ID	航向角 (ANGLE, °)	飞行速度 (VEL, M/S)	弹1投放时刻 ($t_{d,1}$, S)	弹2投放时刻 ($t_{d,2}$, S)	弹3投放时刻 ($t_{d,3}$, S)	弹1延时 (τ_1 , S)	弹2延时 (τ_2 , S)	弹3延时 (τ_3 , S)
FY-1	201.2	112.5	20.2	24.0	26.1	4.3	1.5	3.2
FY-2	191.7	129.7	26.6	29.7	31.5	3.8	2.0	3.4
FY-3	94.4	131.0	16.5	20.1	22.1	5.3	2.9	2.8
FY-4	350.0	104.9	23.4	25.4	28.1	3.8	3.2	4.4
FY-5	127.0	115.6	12.8	14.1	16.4	2.6	3.3	5.5

方案二：强化学习（RL）

求解方案与代码实现

求解方案：采用基于进化策略/交叉熵方法（CEM）的协同训练。将问题视为连续动作空间下的强化学习任务，通过两阶段训练，使 5 架无人机学会协同策略。

- Stage 1:** 单机预训练：对每架无人机独立运行 CEM 算法，优化其单机策略 μ_i ，使其能够对任一目标产生有效遮蔽（`env.get_individual_reward`）。
- Stage 2:** 多机协同微调：将 5 个预训练策略组合成一个 40 维联合策略 \mathbf{x}_{init} 。运行协同 CEM，通过奖励 (Reward Shaping) 来指导协同行为：

$$\text{Reward} = L_{union} - \alpha \cdot L_{overlap} - \beta \cdot N_{lazy}$$

此奖励函数最小化重叠浪费 (α)，并严惩贡献度低的无人机 (β)，从而迫使所有无人机高效地分配干扰任务。

代码实现描述 (Problem5_RL.m, Problem5_RL_Env.m):

- 环境封装 (Problem5_RL_Env.m): 定义了 40 维归一化动作空间 $\in [-1, 1]$ 。核心方法 `get_cooperative_reward` 负责计算总并集时长 L_{union} 、重叠时长 $L_{overlap}$ 和怠工数量 N_{lazy} ，并计算带惩罚的最终得分 `Score`。
- 训练程序 (Problem5_RL.m): 实现了 CEM 算法。
 - Stage 1 循环训练 5 个独立的 8 维高斯分布 μ_i 。
 - Stage 2 将 μ_i 组合为 40 维，进行协同训练，使用 `get_cooperative_reward` 作为适应度函数。

结果 (方案二): 21.2s

UAV ID	航向角 (ANGLE, °)	飞行速度 (VEL, M/S)	弹1投放时刻 ($t_{d,1}$, S)	弹2投放时刻 ($t_{d,2}$, S)	弹3投放时刻 ($t_{d,3}$, S)	弹1延时 (τ_1 , S)	弹2延时 (τ_2 , S)	弹3延时 (τ_3 , S)
FY-1	213.8	119.4	34.3	38.0	40.8	2.3	6.7	3.9
FY-2	259.9	103.9	3.6	5.4	6.5	3.9	4.2	3.5
FY-3	78.9	122.4	23.2	25.4	27.4	2.5	0.0	1.4
FY-4	180.6	127.4	16.3	21.3	27.6	4.4	2.5	1.9
FY-5	117.6	135.3	12.5	14.9	17.4	4.4	3.6	6.7

四、总结与不足

4.1 模型优势与总结

- 物理模型全面：完整建立了导弹、无人机、烟幕弹的运动模型，并考虑了烟幕云团的匀速下沉效应。
- 几何判定精确：采用了基于目标圆柱体采样点和 95% 遮蔽比例的精确判定标准，克服了将目标简化为点的不足。
- 优化策略合理：针对问题的非凸、非光滑、高维特性，采用了 **PSO**、**GA**、多起点 **SQP** 精修 等混合优化策略，并对问题五探索了协同进化算法 (**RL**)，有效处理了黑盒目标函数的全局寻优。
- 并集测度稳健：目标函数的评估器能够精确计算多枚烟幕弹在时间轴上的并集遮蔽时长，符合实际干扰效果。
- 模块化设计：代码模块化清晰，易于调试和扩展。

4.2 主要不足与改进方向

1. 计算复杂度高： 目标函数评估（计算遮蔽时长）是主要的计算瓶颈，其涉及时间轴扫描和精确的几何判定（或多点采样）。
 - 改进： 可采用更高效的零点查找（如二分法）替代暴力时间扫描，以提高计算效率。
2. 遮蔽标准仍可精化： 尽管采用了目标采样，但在数学上，100% 采样点遮蔽与完全遮蔽目标圆柱体并不完全等价。
 - 改进： 可采用更严格的几何模型，例如基于阴影投射或极小化目标上未被遮蔽点到烟幕边界的距离来建立非线性约束。
3. 多目标优化局限： 问题五的目标是最大化 $\sum L_k$ （总时长之和），这可能导致资源倾向于拦截容易遮蔽的导弹。若要求“确保所有导弹的最大遮蔽时间都不低于某个阈值”，则需采用 **Max-Min** 多目标优化策略。
4. **RL** 算法可升级： 方案二采用的 **CEM** 属于进化算法。若能构建一个平滑可微的近似目标函数（对示性函数 $\mathbb{I}(\cdot)$ 进行软化），可以尝试使用更高效的基于梯度的 **RL** 算法（如 **PPO** 或 **SAC**），以加快收敛速度。