强化学习课程习题

德梅萃 · P. 博赛卡斯（Dimitri P. Bertsekas）
李宇超（Yuchao Li）

习题 1 [Ber17, 习题 2.1] 考虑由节点（node）$1,\dots,6$ 以及连接它们的边（edge）构成的图（graph）如图 1所示。请采用动态规划算法计算节点 $1,\dots,5$ 到节点 6 的最短路径。采用编程或者手算方式均可。*提示*：在此问题中，阶段数目 N 应当设为多少？每阶段中应当包含哪些状态呢?

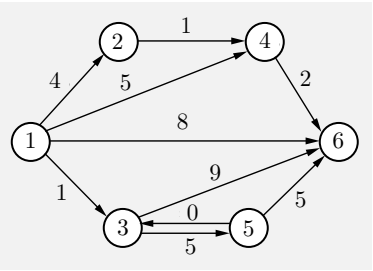

图 1: 习题 1 中涉及的图。标注于边旁的数值表示边长。

习题 2 [Ber17, 习题 1.8] 考虑计算如下矩阵序列的乘积:

$$M_1M_2\cdots M_kM_{k+1}\cdots M_N,$$

其中矩阵 M_k 的维度为 $n_k\times n_{k+1}$。在计算该乘积时，相邻矩阵乘积运算的顺序将影响计算量。例如，当计算 $M_1M_2M_3$ 时，如果 $n_1=10$，$n_2=10$，$n_3=1$ 且 $n_4=10$，那么 $\big((M_1M_2)M_3\big)$ 需要执行 20 次标量乘积运算，而 $\big(M_1(M_2M_3)\big)$ 则需要执行 200 次（$m\times n$ 和 $n\times k$ 维的两个矩阵相乘需要执行 mnk 次标量乘积运算。）请采用动态规划的方法计算出最优的乘积顺序。

习题 3 在本习题中，我们考虑动态规划算法在解决确定性的有限阶段有限状态问题时的计算复杂度。假设在此问题的阶段为 $k=0,1,\dots,N$，即共有 $N+1$ 个阶段。每阶段中状态 x_k 数量的上限为 n，即阶段 k 最多有 n 个不同的状态。类似的，每个控制约束集 $U_k(k_k)$ 中最多有 m 个元素。那么，

在执行动态规划算法时，最多需要做多少次求和计算呢？与之相比，如果采取穷举法，最多需要考虑多少种不同的控制序列 $\{u_0, u_1, \ldots, u_{N-1}\}$ 呢？

习题 4 [Ber17, 习题 1.6] 假设我们有一艘船，其最大载重量为 z，船上要装载 N 种不同数量的不同物品。设 v_i 表示第 i 种物品的价值，w_i 表示第 i 种物品的重量，x_i 表示船上装载的第 i 种物品的数量。问题是要找到最有价值的货物，即最大化

$$\sum_{i=1}^{N} x_i v_i$$

同时满足约束条件

$$\sum_{i=1}^{N} x_i w_i \leqslant z$$

且 $x_i = 1, 2, \ldots, N$. 请用动态规划来表述这个问题。

习题 5 考虑习题 1 中的最短路径问题。请采用策略前展算法（rollout）给出该问题的近似解。*提示*：可以采用贪心策略作为策略前展中的启发式方法。例如，当处于节点 3 时，可选的下一个节点包括了节点 5 和节点 6。贪心策略比较前往这两个节点的边的长度（即 5 和 9），并选择前往边长较短的后续节点（即对应于边长 5 的节点 5）。

习题 6 [Ber17, 例 3.5.1] 某智力竞赛共有 N 道题目，记作题目 $1, 2, \ldots, N$。参赛者可以自由选择其答题次序，当答对题目 i 时，参赛者可以得 R_i 的奖励，并继续回答后续问题。一旦某题目回答错误，参赛者便不可以回答后续问题。小明答对题目 i 的概率为 p_i，那么他应当如何安排答题顺序从而使他期望的收益最大化呢？请采用动态规划给出该问题的解析解。*提示*：在此问题中，什么是状态、控制和系统呢？

习题 7 [Ber76, 第一章习题 14] 某农民每年收获 x_k 单位重量的粮食。他将 $(1 - u_k)x_k$ 的粮食储藏起来，而另外的 $u_k x_k$ 粮食则用于促进增产，即 $0 \leqslant u_k \leqslant 1$ 决定了投资占收获的比例。假设下一年度粮食产量 x_{k+1} 满足

$$x_{k+1} = x_k + w_k u_k, x_k, \qquad , k = 0, 1, \ldots, N-1.$$

其中 w_k 是独立的随机变量，其概率分布不依赖于 x_k，u_k 和 k。而且，我们记 $E\{w_k\} = \bar{w}$。那么我们要求解的最优控制问题为选择投资策略从而最大化年后剩余粮食总储量的期望值，即

$$\underset{\substack{w_k \\ k=0,1,\ldots,N-1}}{E} \left\{ x_N + \sum_{k=0}^{N-1} (1 - u_k) x_k \right\}.$$

习题 8 假设某玩家参加某个探险游戏，每次可选择在规定时间内在甲、乙和丙三处寻宝。每处都有确定的概率找到宝藏或触发陷阱。假设每个玩家最多可以寻找三轮，每次只能选择一处寻宝。若触发陷阱，则本轮及后续奖励均清零；若安全（不论是否寻到宝藏），则可以继续做下一次选择。是否触发陷阱与是否发现宝藏相互独立。每轮结束后，所有地方宝藏和陷阱将会重置（如果上一轮中发现了宝藏，则会添加新的宝藏），且不同阶段可以选择同一地点进行探索，并且同一地点在不同轮次中发现宝藏和触发陷阱的概率不变。甲处发现宝藏的概率是 0.5，奖励是 20，触发陷阱的概率是 0.2；乙处找到宝藏的概率是 0.7，奖励 15，触发陷阱的概率是 0.3；丙处发现宝藏的概率是 0.3，奖励 30，触发陷阱的概率是 0.1。采用动态规划求解此问题。

习题 9 [Ber22, 习题 1.5] 本习题的目的是通过一维的线性二次型问题来体现策略迭代与牛顿法的等效性。在此问题中，系统为 $f(x,u)=x+bu$，阶段费用为 $g(x,u)=x^2+ru^2$，其中 $b\neq 0$ 且 $r>0$。

(a) 请验证贝尔曼方程

$$Kx^2=\min_{u\in\Re}\left[x^2+ru^2+K(x+bu)^2\right]$$

可以写作等效形式 $H(K)=0$，其中

$$H(K)=K-\frac{rK}{r+b^2K}-1.$$

(b) 现考虑策略迭代算法

$$K_k=\frac{1+rL_k^2}{1-(1+bL_k)^2},$$

其中

$$L_{k+1}=-\frac{bK_k}{r+b^2K_k},$$

且 $\mu(x)=L_0x$ 为起始策略。证明该算法等效于通过牛顿法

$$K_{k+1}=K_k-\left(\frac{\partial H(K_k)}{\partial K}\right)^{-1}H(K_k)$$

求解贝尔曼方程 $H(K)=0$。

(c) 请将上述结论推广到系统为 $f(x,u)=ax+bu$，阶段费用为 $g(x,u)=qx^2+ru^2$，其中 $a,b\neq 0$ 且 $q,r>0$ 的情况。

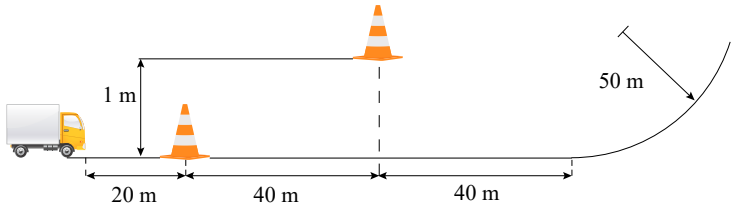

图 2: 习题 10 中讨论的自动驾驶问题示意图。

习题 10 [LCM20] 本习题的目的是练习使用状态扩充（state augmentation）将含有复杂控制约束的问题描述为标准的动态规划问题。考虑某自动驾驶卡车的转向控制问题；如图3所示。该问题共有 N 个阶段。在第 k 个阶段，状态 x_k 表示当前车辆距离规划轨迹的偏差，并且 x_k 应处于集合 X_k 中。控制 u_k 表示该阶段方向盘的转向，并且对于 $k = 1, \ldots, N-1$，该阶段的控制需满足

$$u_k \in U_k(x_k), \qquad |u_k - u_{k-1}| \leqslant \epsilon,$$

此处 u_{k-1} 表示第 $k-1$ 个阶段的控制，而 ϵ 为某个给定常数。其中约束 $|u_k - u_{k-1}| \leqslant \epsilon$ 表示相邻两个时刻方向盘转向的差值不能超过 ϵ。初始时刻的控制 u_0 需满足 $u_0 \in U_0(x_0)$。假设该问题的系统函数为 $x_{k+1} = f_k(x_k, u_k)$，阶段费用函数为 $g_k(x_k, u_k)$，且终止阶段费用为 $g_N(x_N)$。请通过状态扩充方法将该问题表述为标准的动态规划问题。

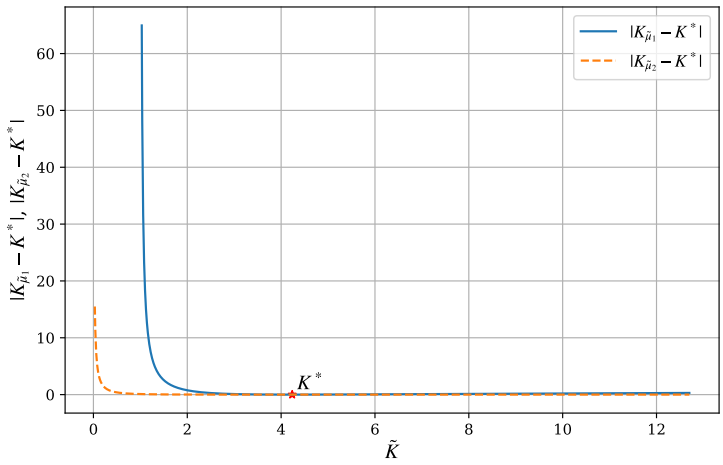

图 3: 习题 11（d）得到的图。

习题 11 本习题的目的是通过编程的方式验证牛顿法用于无穷阶段线形二次型问题时的超线性收敛。考虑一维的线形二次型问题，系统为 $f(x,u)=2x+u$，阶段费用为 $g(x,u)=x^2+u^2$。

(a) 编写值迭代算法的代码，找出最优费用函数的参数 K^*，即找出非负的参数 K^*，使其满足

$$K^*x^2=\min_{u\in\Re}[x^2+u^2+K^*(2x+u)^2].$$

(b) 当终止费用函数近似为 $\tilde{K}x^2$，通过一步前瞻最小化得到的策略 $\tilde{\mu}_1$ 满足 $\tilde{\mu}_1(x)=L_1x$，其中

$$L_1x\in\arg\min_{u\in\Re}[x^2+u^2+\tilde{K}(2x+u)^2]. \tag{1}$$

显然，不同取值的 $\tilde{K}$ 将会影响得到的一步前瞻 $\tilde{\mu}_1(x)=L_1x$。请计算 $\tilde{K}_1$，从而使得到的一步前瞻策略满足 $2+L_1=1$。

(c) 类似的，两步前瞻最小化得到的策略 $\tilde{\mu}_2$ 满足 $\tilde{\mu}_2(x)=L_2x$，其中

$$L_2x_0\in\arg\min_{u_0\in\Re}\left[x_0^2+u_0^2+\min_{x_1=2x_0+u_0,u_1\in\Re}[x_1^2+u_1^2+\tilde{K}(2x_1+u_1)^2]\right]. \tag{2}$$

请计算 $\tilde{K}_2$，从而使得到的两步前瞻策略满足 $2+L_2=1$。

(4) 给定某策略 $\mu(x)=Lx$，如果满足 $|2+L|<1$，那么该策略的费用函数为 $K_\mu x^2$，其中 K_μ 满足

$$K_\mu=\frac{1+L^2}{1-(2+L)^2}.$$

请通过编程在同一图中画出两条曲线：两图的横坐标均为 $\tilde{K}$，在第一条曲线中，纵坐标为 $K_{\tilde{\mu}_1}-K^*$，其中 $\tilde{\mu}_1$ 由式(1)给出；第二条曲线的纵坐标为 $K_{\tilde{\mu}_2}-K^*$，其中 $\tilde{\mu}_2$ 由式(2)给出。提示：在画第一条曲线时，$\tilde{K}$ 的取值范围是什么呢？类似的，画第二条曲线时呢？

习题 12 [KG88] 本习题的目的是通过编写代码实现课程中讲解的经典形式的模型预测控制。我们考虑线性系统 $f(x,u)=Ax+Bu$，其中

$$A=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix},\quad B=\begin{bmatrix}0\\ 1\end{bmatrix}.$$

阶段费用为 $g(x,u)=x'Qx+Ru^2$，其中

$$Q=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\quad R=1.$$

在此基础上，我们要求状态的每个组分的绝对值小于 5，即状态 x 处于约束集 X 中，且

$$X=\{x\,|\,x=(y,z),\,|y|\leqslant 5,\,|z|\leqslant 5\}.$$

控制约束不随状态改变，即 $U(x)=U$，且

$$U=\{u\,|\,|u|\leqslant 1\}.$$

针对上述问题，请编程实现教材中式 (1.85)-(1.88) 所表示的经典形式的模型预测控制。其中参数 ℓ 可以设为 5。通过随机生成属于约束集 X 的初始状态 x_0 测试所得模型预测控制的性能。另外，也可以尝试扩大或减小 ℓ 的取值，探索其对于所得策略性能的影响。

习题 13 考虑上一题中定义的无穷阶段最优控制问题。接下来我们尝试一种模型预测控制的变形。

(a) 让我们首先考虑一个线性二次型问题，其中的系统方程和阶段费用分别为

$$x_{k+1}=Ax_k+Bu_k,\qquad x_k'Qx_k+u_kRu_k,$$

这里的 A、B、Q 和 R 的取值见上一题。教材中给出的值迭代算法可拓展适用于该问题。具体而言，定义 K_0 为 0 矩阵，那么值迭代即生成一个矩阵序列 $\{K_i\}$，满足

$$K_{i+1}=A'K_iA-(A'K_iB)(R+B'K_iB)^{-1}(B'K_iA)+Q.$$

请编程实现该算法，并检验所得的矩阵序列 $\{K_i\}$ 是否收敛。

(b) 假设所得的矩阵序列 $\{K_i\}$ 收敛。我们将该序列的极限记为 K^*。接下来我们将修改上一题目中的模型预测控制算法。具体而言，去掉约束 $x_{k+\ell}=0$，并将优化目标修改为

$$x_{k+\ell}'K^*x_{k+\ell}+\sum_{t=k}^{k+\ell-1}g(x_t,u_t),$$

其余部分保持不变。通过随机生成属于约束集 X 的初始状态 x_0 测试所得模型预测控制的性能。另外，也可以尝试扩大或减小 ℓ 的取值，探索其对于所得策略性能的影响。

(c) 比较本题中的模型预测控制与上一题目中模型预测控制性能的差异。另外，当选取的 ℓ 较小时，是否出现了从某一初始状态 x_0 开始，模型预测控制对应的优化问题有解，但在该控制的作用下，后续某阶段 k 时，模型预测控制对应的问题没有解了呢?

习题 14 [Ber17, 习题 3.24，哈代定理] 令 $\{a_1, \ldots, a_n\}$ 和 $\{b_1, \ldots, b_n\}$ 表示实数数列。我们希望给每个 i 分配不同的 j_i，从而最大化 $\sum_{i=1}^{n} a_i b_{j_i}$。

(a) 请采用动态规划给出该问题的解析解。*提示*：在此问题中，什么是状态、控制和系统呢?

(b) 尽管该问题有解析解，但接下来我们将通过编程实现策略前展(rollout)算法，从而给出该问题的近似解，并将其与解析解给出的最优做比较，以便观察最优解、基本启发式（base heuristic）给出的解、以及前展策略（rollout policy）给出的解的关系。随机生成实数序列 $\{a_1, \ldots, a_n\}$ 和 $\{b_1, \ldots, b_n\}$，采用贪心策略作为基本启发式，实现策略前展算法。测试 $n = 10, 50, 100$ 时对应的问题，并比较最优解、基本启发式给出的解、以及前展策略给出的解的关系。

(c) 尝试采用其他的基本启发式并采用策略前展算法求解 $n = 10, 50, 100$ 时对应的问题。

习题 15 在习题 4 中，我们考虑了在船上放置物品的问题，以求最大化货船商品的价值。在本题目中，我们将编程求解其精确和近似解，并加以比较。

(a) 编写程序随机生成问题中的参数，包括商品种类数目 N、最大载重量 z、第 i 种商品的价值 v_i 和单位重量 w_i。

(b) 编写程序给出相关问题的精确解。*提示*：该问题为典型的整数规划问题，有众多的开源软件可用于求解该问题；例如`https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.milp.html`

(c) 尝试设计多个启发式方法，给出该问题的次优解。

(d) 利用上述的启发式方法，编程计算相应的策略前展给出的解，验证所得解是否优于相应的基本启发式的解，并将所得结果与最优解相比较。

(e) 将上述的基本启发式结合，从而得到一个超级启发式方法，并以此为基本启发式，计算相应的策略前展所得的解。将所得结果与所有启发式所得的解、(e) 中利用单一启发式得到的策略前展的解以及最优解进行比较。

习题 16 假设我们采用 20 步前瞻最小化（20-step lookahead minimization）求解某问题，此时的前瞻最小化就对应于在图 4所示的树状图中，找出从最上端的节点（对应于当前状态 x_0）到最下端一层的最短路径，且最下端的节点 x_{20} 还包含终止费用 $\tilde{J}(x_{20})$。除去终止费用外，其余各边的长度均为 0。图中第 20 层共有 21 个节点，其终止费用从左到右依次为 $3, 5, 4, 2, 0, -2, -5, -3, -1, 1, 0, 2, 3, 4, 5, 4, 3, 2, 1, 3, 2$。请通过编程实现增量策略前展（incremental rollout）从而给出该最短路径问题的近似解。基本策略可以是固定选取某一侧的边，也可以是交替选取一侧的边。尝试采用不同的 δ 取值并比较该算法的性能。

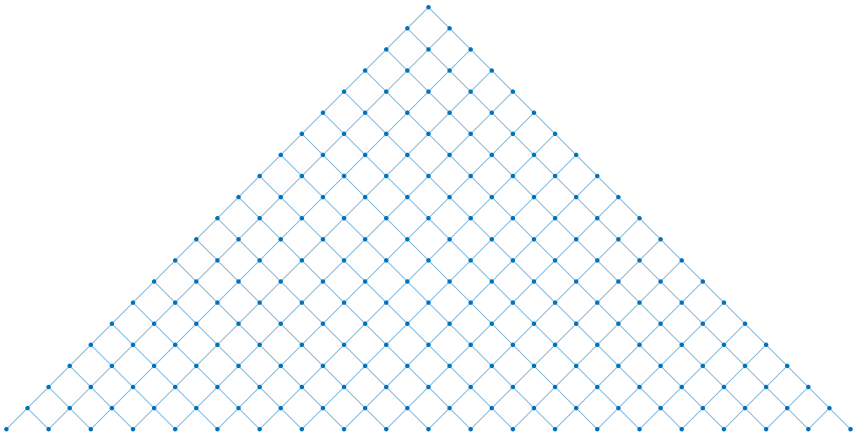

图 4: 习题 15 中涉及的图。该树状图中，$\ell = 20$。最上端的节点为当前状态 x_0，最下一层含有 21 个节点，代表了不同的 x_{20}。

习题 17 [Ber17, 第 3.4 节] 考虑涉及 $N = 5$ 阶段的资产出售问题。在该问题中，我们根据买方出价来决定是否出售资产，从而最大化换算到 $N = 5$ 这一的阶段预期收益。该问题中的初始阶段的状态空间为 $X_0 = \{0\}$，后续阶段的状态空间为 $X_k = \{8, 9, 10, T\}$，$k = 1, 2, \ldots, N$，其中 T 表示资产已经出售。对于所有阶段，我们都有两个控制选项：$u = 0$ 代表接受上一阶段

的报价出售资产，以及 $u=1$ 代表拒绝已有报价等待本阶段新的随机报价。我们将 k 阶段收到的新报价记为 $w_k \in \{8,9,10\}$。那么对于 $x_k \neq T$，系统的状态函数 f_k 为

$$x_{k+1} = \begin{cases} w_k & \text{如果 } u=1, \\ T & \text{如果 } u=0. \end{cases}$$

当处于阶段 N 时，对于 $x_N \neq T$，终止收益为 $g_N(x_N) = x_N$。对于 $k \neq N$ 以及 $x_k \neq T$，每阶段收益为

$$g_k(x_k, u_k) = \begin{cases} 0 & \text{如果 } u=1, \\ (1+r)^{N-k} x_k & \text{如果 } u=0. \end{cases}$$

其中 $r=0.1$ 表示利率。显然一旦处于 T，后续阶段也将处于 T，并且不再有收益。我们的目标是找出 $\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\}$ 从而最大化

$$\underset{\substack{w_k \\ k=0,1,\ldots,N-1}}{E} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g\big(x_k, \mu_k(x_k), w_k\big) \right\}.$$

假设在 $k=0,1,\ldots,N-2$ 阶段时，w_k 为 $8,9,10$ 的概率分别为 $0.3, 0.5, 0.2$，而 w_{N-1} 取 $8,9,10$ 的概率则为 $0.5, 0.2, 0.3$。请编程求解如下问题。

(a) 请采用动态规划给出该问题的最优解。

(b) 假定基本策略 π 为 $x_k > 8$ 即选择接受报价，请编程计算相应的一步前瞻策略前展策略 $\tilde{\pi}$。请通过解析计算以及采样平均两种方式来计算 $J_{\pi,k}$。样本数可设定为 50。比较其与最优解的关系。

(c) 基于 (b) 中的基本策略 π，请采用讲义中例 2.7.4 的方法，运用蒙特卡罗树搜索（MCTS）计算一步前瞻的策略。样本总量为 20。比较其与最优解的关系。*提示*：为了在探索项 R 中将常数 c 设为 $c=\sqrt{2}$，Q-因子的取值需要缩放到 $[0,1]$ 的范围内。

习题 18 [ERL+22] 本习题的目的是学习运用多智体策略前展算法求解多智体路径规划问题，我们将考虑图5中涉及的三种不同的情境。图中的实心圆代表机器人，而同色的空心圆则是相应的目的地。各机器人的编号为：蓝色 1 号，红色 2 号，绿色 3 号。我们的目标是为三个机器人规划路径，使其尽快抵达各自目的地，并避免碰撞。黑色实心方块为静置障碍物，此外机器人也不允许移动到黑色边框标注的围墙之外。每时刻各机器人可最多

选择‘上’、‘下’、‘左’、‘右’和‘静止’五个控制，分别对应于移动到当前网格的相邻网格或停留在原网格。每个机器人在没有到达目的地前，每时刻费用为 1，而一次碰撞的费用为 200。因此最小化各机器人的总费用就代表了我们的设计需求。请编程求解如下问题。在涉及策略前展算法的题目中，当有多个控制可以取得最小值并且基本策略给出的控制是其中之一时，我们选取基本策略给出的控制。

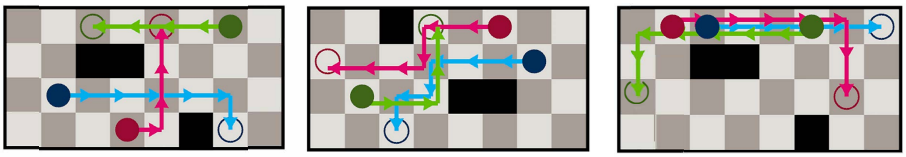

图 5: 习题 9 中涉及的图。箭头给出了假设其他机器人不存在时的最短路径。

(a) 在假设环境中不存在其他机器人的前提下，采用前向动态规划（forward dynamic programming）求解各机器人到达其目的地的最短路径。

(b) 以（a）中方法为基本策略，采用一般策略前展算法计算当前时刻各机器人的控制。*提示*：以左图为例，1 号机器人当前所有控制都可行，即有 5 个控制选择；2 号和 3 号机器人则有 4 个控制可选。因此，当前时刻需要考虑的控制个数为 $5\times4\times4$ 个，即需要执行同等数目的仿真。

(c) 以（a）中方法为基本策略，采用多智体策略前展算法计算当前时刻各机器人的控制。其中执行优化的顺序为机器人编号次序。比较其与一般策略前展算法的计算量。

(d) 针对右侧图中的问题，以（a）中方法为基本策略，采用多智体策略前展算法计算当前时刻各机器人的控制。其中执行优化的顺序为机器人编号的倒序，即先优化 3 号机器人，然后 2 号，最后 1 号。比较所得控制与（c）中的差别。

习题 19 [Ber20, 例 3.1.4] 本习题的目的是更深入地了解增量梯度法在

求解优化问题时的行为。我们考虑采用梯度法求解如下问题：[1]

$$\begin{aligned} \text{minimize} \quad & f(y) = \frac{1}{2}\sum_{i=1}^{m}(c_i y - b_i)^2 \\ \text{subject to} \quad & y \in \Re, \end{aligned}$$

其中 c_i 和 b_i 是给定标量，且对所有 i 都有 $c_i \neq 0$。对于单个组分 $f_i(y) = \frac{1}{2}(c_i y - b_i)^2$，取得最小值的点是

$$y_i^* = \frac{b_i}{c_i},$$

而二次型费用函数 f 的最小值点则是

$$y^* = \frac{\sum_{i=1}^{m} c_i b_i}{\sum_{i=1}^{m} c_i^2}.$$

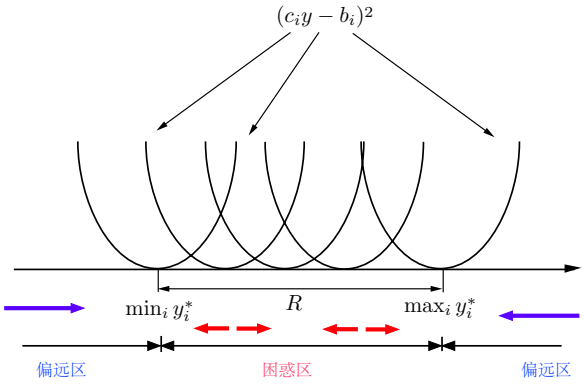

图 6: 习题 19 中优化问题的偏远区与困惑区。

(a) 证明 y^* 处于组分最小值点构成的区域

$$R = \left[\min_i y_i^*, \max_i y_i^*\right]$$

之内。

(b) 采用增量梯度法求解该问题时，计算公式为

$$y^{k+1} = y^k - \gamma^k c_{i_k}(c_{i_k} y^k - b_{i_k}).$$

[1]下列问题中，“minimize” 表示 “最小化”，而 “subject to” 则表示 “约束条件为”。

假设 $y^k \notin R$，请证明如果 $\gamma^k \leqslant \min_i \frac{1}{c_i^2}$，那么 y^{k+1} 比 y^k 更靠近困惑区 R。

(c) 采用一般梯度法求解该问题时，计算公式为

$$y^{k+1} = y^k - \gamma^k \sum_{i=1}^{m} c_i(c_i y^k - b_i).$$

请证明对于任意满足

$$0 < \gamma \leqslant \frac{1}{\sum_{i=1}^{m} c_i^2}$$

的常数步长 γ，生成的序列 $\{y^k\}$ 都会收敛到 y^*。

参考文献

[Ber76] Dimitri P. Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, 1976.

[Ber17] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 4 edition, 2017.

[Ber20] Dimitri P. Bertsekas. 强化学习与强化学习（*Reinforcement Learning and Optimal Control*）. 清华大学出版社, 2020.

[Ber22] Dimitri P. Bertsekas. 策略前展、策略迭代与分布式强化学习（*Rollout, Policy Iteration, and Distributed Reinforcement Learning*）. 清华大学出版社, 2022.

[ERL+22] William Emanuelsson, Alejandro Penacho Riveiros, Yuchao Li, Karl H Johansson, and Jonas Mårtensson. Multiagent rollout with reshuffling for warehouse robots path planning. *arXiv preprint arXiv:2211.08201*, 2022.

[KG88] S. Sathiya Keerthi and Elmer G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293, 1988.

[LCM20] Yuchao Li, Xiao Chen, and Jonas Mårtensson. Linear time-varying model predictive control for automated vehicles: feasibility and stability under emergency lane change. *IFAC-PapersOnLine*, 53(2):15719–15724, 2020.