文章编号: 1001-0920(2007)02-0155-05

SMDP 基于 Actor 网络的统一 NDP 方法

唐 昊, 陈 栋, 周 雷, 吴玉华
(合肥工业大学 计算机与信息学院, 合肥 230009)

摘 要: 研究半马尔可夫决策过程(SMDP)基于性能势学习和策略逼近的神经元动态规划(NDP)方法. 通过 SMDP 的一致马尔可夫链的单个样本轨道,给出了折扣和平均准则下统一的性能势 TD(λ)学习算法,进行逼近策略评估;利用一个神经元网络逼近结构作为行动器(Actor)表示策略,并根据性能势的学习值给出策略参数改进的两种方法. 最后通过数值例子说明了有关算法的有效性.

关键词: 半 Markov 决策过程; 性能势; TD(λ)学习; 神经元动态规划
中图分类号: TP202 **文献标识码**: A

Unified NDP method for SMDP by an actor network

TANG Hao, CHEN Dong, ZHOU Lei, WU Yu-hua
(School of Computer and Information, Hefei University of Technology, Hefei 230009, China. Correspondent: TANG Hao, E-mail: htang@hfut.edu.cn)

Abstract: A neuro-dynamic programming (NDP) method for a semi-Markov decision processes (SMDP) is studied based on the learning of performance potentials and approximating of policy. Using a single sample path of a uniformized Markov chain of the SMDP, a unified TD(λ) learning formula is presented for both discounted and average criteria as the approximate policy evaluation of the actor algorithms. Approximation architecture such as a neural network is used to represent the policy, and two methods of policy updating are proposed by improving the policy parameters based on the estimates of potentials. A numerical example shows the effectiveness of the corresponding algorithms.

Key words: Semi-Markov decision processes; Performance potentials; TD(λ) learning; Neuro-dynamic programming

1 引 言

性能势概念在马尔可夫决策过程(MDP)的灵敏度分析、数值计算理论优化和仿真优化中具有重要作用[1-4]. 半马尔可夫决策过程(SMDP)是比 MDP 更为广泛的一类随机过程,能描述广泛的一类实际系统[5,6]. 曹希仁教授建立了 SMDP 的性能势理论[7],指出平均性能势是折扣性能势在折扣因子趋向零时的特例,因而两种性能准则下基于性能势的优化可统一起来研究.

人工智能领域的强化学习(RL)是解决马尔可夫决策问题的一个重要仿真技术[8],而神经元动态规划(NDP)是建立在强化学习基础上,解决大规模离散事件动态系统(DEDS)优化的一种有效方法[9]. 性能势可通过样本轨道仿真进行估计或学习,因而能够与基于仿真思想的 RL 和 NDP 有机结合. 基于性能势的 NDP 与通常的查表表示优化方法的一个关键区别是:前者用一些逼近结构(如神经元网络或特征函数等)来表示性能势或策略,后者则需要建立状态与性能势或行动的一一对应关系,即计算机保存的是数据表格. 若逼近结构表示性能势,则称之为评判器(Critic);若表示策略,则称之为行动器(Actor);若存在两个逼近结构分别表示性能势和策略,则称为 NDP 的 Actor-Critic 模式.

文献[10]考虑了 Critic 模式下,两种性能准则 SMDP 基于蒙特卡罗(Monte-Carlo)估计和性能势参数逼近的统一优化问题;文献[11]研究了 Critic 模式下 MDP 基于参数 TD(0)学习的统一 NDP 优化方法,其结果对 SMDP 也适用. 本文根据 SMDP

收稿日期: 2005-10-23; **修回日期**: 2006-01-12.
基金项目: 国家自然科学基金项目(60404009); 安徽省自然科学基金项目(050420303); 合肥工业大学中青年科技创新群体计划项目.
作者简介: 唐昊(1972—),男,安徽庐江人,副教授,博士,从事离散事件动态系统、强化学习和神经元动态规划等研究; 陈栋(1978—),男,江苏宿迁人,硕士生,从事计算机应用技术、强化学习等研究.

的等价一致链,导出性能势的 TD(λ)学习算法,建立折扣和平均准则 SMDP 在 Actor 模式下统一的 NDP 优化方法.

2 SMDP

考虑约束在紧致行动集 D 和平稳策略集 $\Omega_s = \{v \mid v = (v(1), v(2), \cdots, v(M)), v(i) \in D\}$ 上的 SMDP $X = (X_t, \Phi, D, Q^v(t), f^v)$[7],其状态过程 $X_t (t \geq 0)$ 在有限状态空间 $\Phi = \{1, 2, \cdots, M\}$ 中取值,$Q^v(t) = [Q(i,j,v(i),t]$ 为半 Markov 核,$f^v = [f(i,j,v(i))]$ 为性能矩阵. 令 X 的平均代价性能准则为

$$\eta^v = \lim_{T\to\infty} \frac{1}{T} E\left\{\int_0^T f(X_t, Y_t, v(X_t))\,\mathrm{d}t\right\}, \quad (1)$$

这里 Y_t 表示 X_t 的下一状态. 按文献[7],折扣代价向量 η_α^v 的分量满足

$$\eta_\alpha^v(i) = \alpha \lim_{N\to\infty} E\left[\int_0^N e^{-\alpha t} f(X_t, Y_t, v(X_t))\,\mathrm{d}t \mid X_0 = i\right]. \quad (2)$$

不失一般性,可令折扣因子 α 在有界区间取值,即 $\alpha \in (0, a]$,这里 a 是一个较大的常数. 优化目标就是在策略空间上寻找一个最优策略 v^*,使所选择的平均或折扣性能准则最小.

根据文献[7,10,12],对任意 $\alpha \geq 0$,SMDP X 等价于连续时间 MDP $X^\alpha = (X_t^\alpha, \Phi, D, A_a^v, f_a^v)$. 其中:$A_a^v$ 为等价无穷小矩阵;f_a^v 为等价性能向量,其分量记为 $f_\alpha(X, v(X))$. 假设半 Markov 核函数 $Q^v(t)$ 关于策略 v 在行动集 D 上连续,则可定义一个一致化参数 μ[10]. 于是,折扣因子 $\beta_\alpha = \mu/(\mu+\alpha)$ 和转移矩阵 $P_a^v = [p_\alpha(i,j,v(i))] = A_a^v/\mu + I$,决定了 X 的一个等价 α- 一致 Markov 链

$$\overline{X} = (\overline{X_n}, \Phi, D, P_a^v, f_a^v),$$

其中 I 表示单位矩阵. 根据文献[7,12],X 的平均代价问题可看作 X,X^α 或 $\overline{X}$ 的折扣问题的极限情况,是其特例,故可统一研究.

3 Actor 模式下基于性能势学习的 NDP 优化方法

下面主要讨论 Actor 模式下的 NDP 优化,即利用神经元网络逼近表示策略. 整个策略迭代算法包括逼近策略评估和逼近策略更新. 前者主要用 TD(λ) 学习来得到近似的性能势;后者是对神经元网络的参数进行改进,得到更新的权系数,即策略参数,从而获得改进的策略.

3.1 平均和折扣性能势的统一 TD(λ) 学习

TD(λ) 是由 Singh 等将 Klopf 提出的适合迹(Eligibility trace) 记忆机制引入强化学习发展而来的[13]. 它是将 TD(0) 学习与 Monte-Carlo 方法有机结合的重要强化学习方法.

对任意 $\alpha \geq 0$,令 SMDP X 的性能势向量 g_a^v 满足 Poisson 方程 $(\alpha I - A_a^v + \mu e\pi_a^v) g_a^v = f_a^v$[7,10,12]. 这里:$\pi_a^v$ 为 X^α 或 $\overline{X}$ 的稳态分布,e 为元素皆为 1 的列向量. 一致链 $\overline{X}$ 的性能势向量 $\overline{g}_{\beta_a}^v$ 满足 $(I - \beta_a P_a^v + \beta_a e\pi_a^v)\overline{g}_{\beta_a}^v = f_a^v$. 记性能势的分量分别为 $g_a^v(i)$ 和 $\overline{g}_{\beta_a}^v(i)$. 因为 $\overline{g}_{\beta_a}^v = (\mu+\alpha) g_\alpha^v$[10],若能根据一致链 $\overline{X}$ 的一条样本轨道得到 $\overline{g}_{\beta_a}^v$ 的学习值或估计值 $\widetilde{g}_{\beta_a}^v$,则 SMDP X 的性能势 g_α^v 的近似值 $\widetilde{g}_\alpha^v$ 可按 $\widetilde{g}_\alpha^v = \widetilde{g}_{\beta_a}^v/(\mu+\alpha)$ 计算.

根据性能势的样本轨道定义[10],引入迹的多步思想,用下列 Robbins-Monro 随机逼近算法对性能势进行逼近:

$$\widetilde{g}_{\beta_a}^v(\overline{X_n}) := \widetilde{g}_{\beta_a}^v(\overline{X_n}) + \gamma \sum_{m=n}^{\infty} (\lambda\beta_a)^{m-n} d_m.$$

这里:λ 为衰减因子;γ 为步长;即时差分 d_m 为

$$d_m = f_a(\overline{X_m}, v(\overline{X_m})) - \beta_a\widetilde{\eta}^v + \beta_a\widetilde{g}_{\beta_a}^v(\overline{X_{m+1}}) - \widetilde{g}_{\beta_a}^v(\overline{X_m}), \quad (3)$$

其中 $\widetilde{\eta}^v$ 是一致链的平均代价估计. 上述两个表达式构成了性能势的 TD(λ) 学习公式. 当 $\lambda = 0$ 时,即为常见的 TD(0) 学习;$\lambda = 1$ 时,则与 Monte-Carlo 方法等价. 当随机过程从状态 $\overline{X_n}$ 跳转到 $\overline{X_{n+1}}$ 时,可对所有状态的性能势学习值进行更新,即

$$\widetilde{g}_{\beta_a}^v(i) := \widetilde{g}_{\beta_a}^v(i) + \gamma_n(i) z_n(i) d_n, \quad (4)$$

其中 $z_n(i)$ 为适合迹. $\widetilde{\eta}^v$ 按下式更新:

$$\widetilde{\eta}^v := (1-\delta_n)\widetilde{\eta}^v + \delta_n f_a(\overline{X_n}, v(\overline{X_n})), \quad (5)$$

其中 δ_n 为平均代价的学习步长,一般比 $\gamma_n(i)$ 衰减快.

适合迹分为累积迹和替换迹两种[8,9],分别对应 $z_n(i)$ 的两种计算方法,即重返累积(Every visit) 形式

$$z_n(i) = \begin{cases} \beta_a\lambda z_{n-1}(i), & \overline{X_n} \neq i; \\ \beta_a\lambda z_{n-1}(i) + 1, & \overline{X_n} = i; \end{cases} \quad (6)$$

和重返置位(Restart) 形式

$$z_n(i) = \begin{cases} \beta_a\lambda z_{n-1}(i), & \overline{X_n} \neq i; \\ 1, & \overline{X_n} = i. \end{cases} \quad (7)$$

TD(λ) 算法具体描述如下:

算法 1 性能势学习算法

Step1: 初始化 n,$\widetilde{\eta}^v$,$\widetilde{g}_{\beta_a}^v$ 以及 $z_{n-1}(i)$.

Step2: 由当前状态 $\overline{X_n}$,根据一致链的转移矩阵 P_a^v,仿真转移到下一状态 $\overline{X_{n+1}}$.

Step3: 对所有的状态 i,选择合适的步长参数 δ_n 和 $\gamma_n(i)$,按式(6) 或式(7),(5),(3) 和(4) 分别更新 $z_{n-1}(i)$,$\widetilde{\eta}$,d_n 和 $\widetilde{g}_{\beta_a}^v$.

Step4: 判断停止条件是否满足. 若满足,则跳

出;否则,令 $n:=n+1$,转 Step2.

与文献[9]类似,不难证明上述算法在无穷步时的收敛性,此处不再赘述. 注意到,文献[9]中给出的即时差分公式与式(3)不同,对 $\alpha=0$,即 $\beta_t=1$ 时的平均情况不适用. 这里,根据性能势思想构造的 TD 学习算法对平均和折扣准则都适用,因而可以构造统一的基于性能势学习的优化算法. 算法 1 的停止条件可以采用固定的学习步数.

3.2　基于 Actor 网络的 NDP 优化

在通常的优化方法中,不仅需要建立表格来保存每个状态的性能势学习值,还需保存对应每个状态的行动值. 而建立在强化学习基础上的神经元动态规划方法是利用参数数目比状态数少的逼近结构来代替这些表格,因而具有节省计算机内存的功能. 本文利用一个神经元网络作为行动器来表示策略,讨论 Actor 模式下的 NDP 算法. 其策略评估可通过上节讨论的 TD(λ) 学习加以实现,策略改进就是更新神经元网络的权系数向量 θ. 固定一个参数向量 θ,就相当于给定了一个策略,简记这样的参数策略为 $v(\theta)=(v(1,\theta),v(2,\theta),\cdots,v(M,\theta))$.

在第 k 次迭代中,逼近策略改进有两种实现方法. 一种是在 $v(\theta_k)$ 作用下,从当前状态 $\overline{X_k}$ 仿真一段样本轨道,学习得到 $\tilde{g}_\alpha^{v(\theta_k)}$,并执行更新

$$\theta_{k+1}=\arg\min_\theta\Big(f_\alpha(\overline{X_k},v(\overline{X_k},\theta))+\beta_\alpha\sum_j^M p_\alpha(\overline{X_k},j,v(\overline{X_k},\theta))\,\tilde{g}_\alpha^{v(\theta_k)}(j)\Big). \tag{8}$$

因为 $f_\alpha(\overline{X_k},v(\overline{X_k},\theta))$ 和 $p_\alpha(\overline{X_k},j,v(\overline{X_k},\theta))$ 是参数向量 θ 的复合函数,故式(8)可用下列梯度形式迭代实现:

$$\theta:=\theta-\gamma_\theta\Big(\Big(\partial\Big(f_\alpha(\overline{X_k},d)+\beta_\alpha\sum_j^M p_\alpha(\overline{X_k},j,d)\,\tilde{g}_\alpha^{v(\theta_k)}(j)\Big)\Big)/\partial d\Big)\Big|_{d=v(\overline{X_k},\theta)}\cdot\frac{\partial v(\overline{X_k},\theta)}{\partial\theta},$$

这里 γ_θ 为步长参数. 另一种方法是在 $v(\theta_k)$ 作用下,仿真一段样本轨道,学习得到 $\tilde{g}_\alpha^{v(\theta_k)}$,并记录轨道经过的状态,构成样本状态集 Φ_k;然后对任意的 $i\in\Phi_k$,求解

$$d_i=\arg\min_{d\in D}\Big(f_\alpha(i,d)+\beta_\alpha p_\alpha(i,j,d)\,\tilde{g}_\alpha^{v(\theta_k)}(j)\Big); \tag{9}$$

最后将 $\{(i,d_i)\mid i\in\Phi_k\}$ 作为样本来训练神经元网络,得到改进的 θ_{k+1}. 上述两种策略参数改进方法对应的具体算法过程如下:

算法 2　参数直接更新法

Step1: 根据实际问题规模和特性,选择一个神经元网络或其他逼近结构. 令 $k=0$,初始化网络权值 θ_k.

Step2: 利用当前参数 θ_k,调用算法 1,得到 $\tilde{g}_\alpha^{v(\theta_k)}$,并记录样本轨道的初始状态 $\overline{X_k}$ 以及轨道所经历的状态,构成样本状态集合 Φ_k. 令 $t=k$,且 $\theta_t=\theta_k$,并选择一个小的常数 ε.

Step3: 选择适当的步长参数 γ_{θ_t},执行更新

$$\theta_{t+1}=\theta_t-\gamma_{\theta_t}\Big(\Big(\partial\Big(f_\alpha(\overline{X_k},d)+\beta_\alpha\sum_j^M p_\alpha(\overline{X_k},j,d)\,\tilde{g}_\alpha^{v(\theta_k)}(j)\Big)\Big)/\partial d\Big)\Big|_{d=v(\overline{X_k},\theta_t)}\cdot\frac{\partial v(\overline{X_k},\theta)}{\partial\theta}\Big|_{\theta=\theta_t}.$$

Step4: 若 $\|\theta_{t+1}-\theta_t\|\le\varepsilon$,则令 $\theta_{k+1}=\theta_{t+1}$,转 Step5;否则令 $t:=t+1$,转 Step3.

Step5: 判断停止条件是否满足. 若满足,则跳出;否则,令 $k:=k+1$,转 Step2.

实际上,算法 2 的每次迭代中,可对 Φ_k 中的所有状态连续执行 Step3 和 Step4,后面的例子中将给出有关实验结果. 下面是另外一种策略更新算法:

算法 3　样本训练法

Step1: 与算法 2 的 Step1 相同.

Step2: 利用当前参数 θ_k,调用算法 1,得到 $\tilde{g}_\alpha^{v(\theta_k)}$,并记录所经历的样本状态,构成集合 Φ_k.

Step3: 对任意 $i\in\Phi_k$,根据式(9)计算得到状态-行动对集合 $V_k=\{(i,d_i)\mid i\in\Phi_k\}$.

Step4: 将 V_k 中的元素作为样本,采用最小二乘等方法,对神经元网络进行训练,得到更新的网络权系数 θ_{k+1}.

Step5: 判断停止条件是否满足. 若满足,则跳出;否则,令 $k:=k+1$,转 Step2.

算法 2 或算法 3 的停止条件可以判断 $\|\bar{\eta}^{(\theta_{k+1})}(\alpha)-\bar{\eta}^{(\theta_k)}(\alpha)\|$,$\|\tilde{g}_\alpha^{(\theta_{k+1})}-\tilde{g}_\alpha^{(\theta_k)}\|$ 或 $\|f_a^{v(\theta_{k+1})}+\beta_\alpha P_a^{v(\theta_{k+1})}\tilde{g}_\alpha^{v(\theta_k)}-f_a^{v(\theta_k)}-\beta_\alpha P_a^{v(\theta_k)}\tilde{g}_\alpha^{v(\theta_k)}\|$ 是否小于一个正常数 ε,并同时判断是否达到固定的迭代步数,以便算法能够正常有效退出.

3.3　Actor 算法的最优性能误差界

线性逼近结构的 NDP 方法,其收敛性已有证明[9]. 但对于一般逼近结构的 NDP 方法,尚无法保证其最优收敛性,故只能对算法进行误差界分析. 文献[10]已给出了 Critic 模式下 SMDP 基于性能势 Monte-Carlo 仿真估计的性能误差界. 同样,对于 Actor 模式下的 NDP 优化,这里存在性能势的学习误差,式(9)的计算误差和神经元网络的训练误差或算法 3 中的策略参数更新误差等. 其中式(9)的计

算误差和神经元网络的训练误差最后又等效为策略参数的更新误差. 假设对任意 $k \geq 0$,基于 TD(λ) 学习的逼近策略评估误差为 $\varepsilon, \varepsilon > 0$,即 $\tilde{g}_{\alpha}^{v(\theta_k)}$ 满足 $\| \tilde{g}_{\alpha}^{v(\theta_k)} - \bar{g}_{\alpha}^{v(\theta_k)} \| \leq \varepsilon$,逼近策略更新步骤中的参数改进误差为 $\delta, \delta > 0$,即对任意 θ, k,参数 θ_{k+1} 满足

$$f_{\alpha}^{v(\theta_{k+1})} + \beta_{\alpha} P_{\alpha}^{v(\theta_{k+1})} \tilde{g}_{\alpha}^{v(\theta_k)} \leq$$

$$f_{\alpha}^{v(\theta)} + \beta_{\alpha} P_{\alpha}^{v(\theta)} \tilde{g}_{\alpha}^{v(\theta_k)} + \delta e.$$

于是,对任意 $\alpha > 0$,文中讨论的 Actor 模式下的神经元策略迭代算法产生的参数序列 θ_k 满足

$$\limsup_{k \to \infty} \| \bar{\eta}_{\alpha}^{v(\theta_k)} - \bar{\eta}_{\alpha}^{v(\theta^*)} \| =$$

$$\limsup_{k \to \infty} \| \eta_{\alpha}^{v(\theta_k)} - \eta_{\alpha}^{v(\theta^*)} \| \leq$$

$$(\delta + 2\beta_{\alpha}\varepsilon)/(1 - \beta_{\alpha}) =$$

$$[(\mu + \alpha)\delta + \mu\varepsilon]/\alpha,$$

这里 θ^* 为最优的策略参数,即对应的策略 $v(\theta^*)$ 是最优的. 其证明此处不再赘述.

4 数值例子

考虑文献[11]中的 SMDP 数值例子,设计一个拓扑结构为 5 ×3 ×1 的 BP 神经元网络,且网络输出映射到区间 [0.5, 3.5]. 在 NDP 算法的实现中,首先将 SMDP 转换成其一致链,通过仿真该链的一条样本轨道来进行性能势的 TD(λ) 学习;然后根据性能势的学习值进行策略参数的改进. 根据系统精确模型,利用数值计算理论方法,如标准的数值迭代或标准的策略迭代,可以求出最优平均代价为 3.717 563 24,或求出给定折扣因子 α 时的最优折扣代价[10, 11].

图 1 是 $\alpha = 0$,$\lambda = 0.1$,且 $L = 30$ 的情况下,采用算法 2 的一次优化曲线,其执行时间为 11.73 s,算法停止策略 $v(\theta)$ 对应的平均代价为 $\bar{\eta}^{v(\theta)} = \eta^{v(\theta)} = 3.858\ 501\ 32$. 在仿真中,当 λ 参数变大接近于 1 时,曲线的波动变大,优化效果变差.

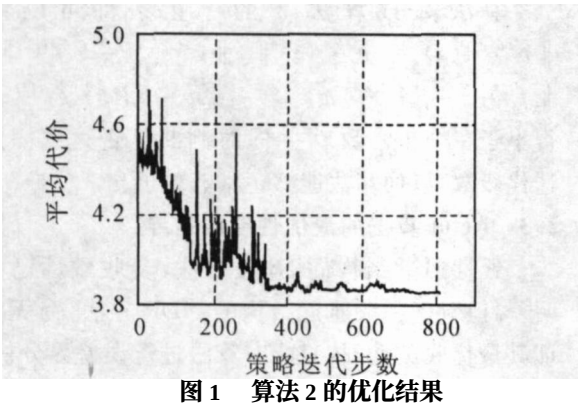

图 1 算法 2 的优化结果

正如第 3.2 节所述,算法 2 的每次迭代中可对 Φ_k 中的所有状态连续执行参数更新. 图 2 便是 $\alpha = 0$,$\lambda = 0.1$,且 $L = 30$ 的情况下采用多状态更新的一次优化曲线,算法执行时间为 4.41 s,$\bar{\eta}^{v(\theta)} = \eta^{v(\theta)} =$ 3.840 190 99. 与图 1 相比,多状态参数改进的算法,其曲线波动较小,优化速度明显变快,总体优化精度有所提高. 而且,当性能势的学习步数变大时,曲线的波动将更小,更为平滑.

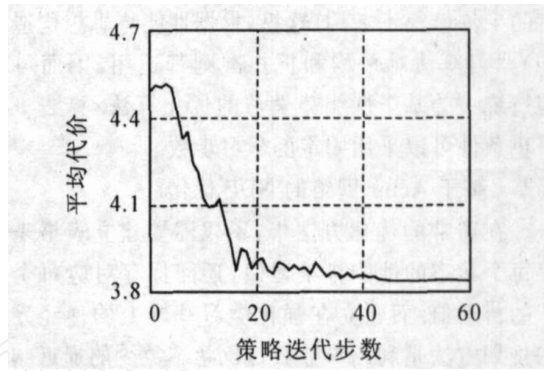

图 2 多状态改进的优化曲线

图 3 是算法 2 取 $\lambda = 0$ 和 $L = 25$ 且进行多状态参数改进时,对应不同折扣因子情况下的优化结果曲线. 可以看到,当折扣因子 α 趋向零时,所有优化曲线将收敛于一点,即趋近平均准则情况. 这说明了平均准则下基于性能势学习的神经元策略迭代是折扣准则情况下的一个特殊形式.

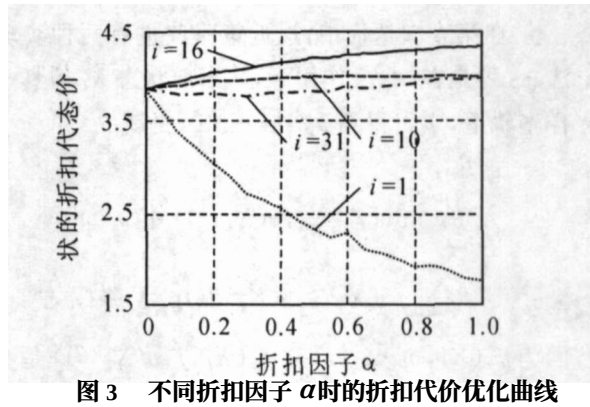

图 3 不同折扣因子 α 时的折扣代价优化曲线

图 4 和图 5 是采用算法 3 时的优化曲线. 其中图 4 是在 $\alpha = 0$,$\lambda = 0$ 且 $L = 20$ 时得到的一次优化曲线,算法执行时间为 7.75 s,$\bar{\eta}^{v(\theta)} = \eta^{v(\theta)} = 3.871\ 096\ 98$;图 5 是 $\alpha = 0.05$,$\lambda = 0$ 且 $L = 20$ 时进行多状态参数改进的优化曲线. 可见该算法收敛速度也很快,只需较少的策略迭代步数就可取得较

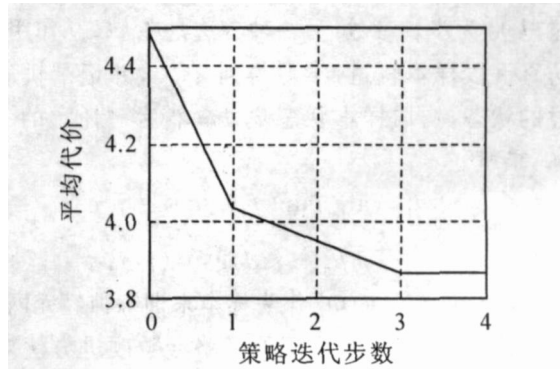

图 4 平均代价优化曲线

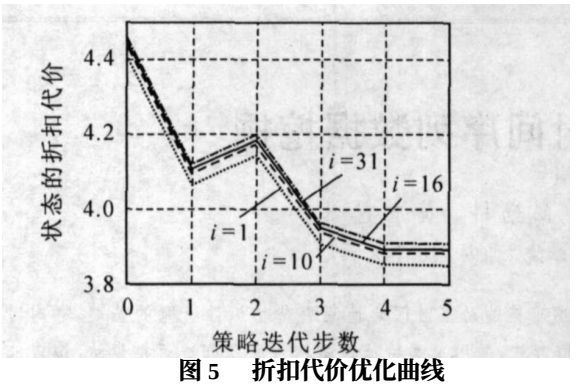

图 5　折扣代价优化曲线

好的优化效果.

5　结　　论

综上所述,通过 SMDP 的等价 α 一致 Markov 链的单个样本轨道,并以性能势的 TD(λ) 学习作为策略评估手段,可以构造平均和折扣两种性能准则下统一的神经元动态规划 Actor 算法.算法实现中,选择不同的折扣因子,可得到不同折扣准则或平均准则下的(次)最优策略参数,且计算机保存的数据相对较少,起到了克服"维数灾"的作用.进一步,可以建立两种准则下统一的性能势参数化即时差分公式,并运用两个不同的神经元网络逼近结构分别表示性能势和策略参数,构造基于性能势参数学习的 Actor-Critic 算法.

参考文献(References)

[1] Cao X R, Chen H F. Perturbation realization, potentials and sensitivity analysis of Markov processes[J]. IEEE Trans on Automatic Control, 1997, 42(10): 1382-1393.

[2] Cao X R. The relations among potentials, perturbation analysis, and Markov decision processes[J]. Discrete Event Dynamic Systems: Theory and Applications, 1998, 8(1): 71-78.

[3] Cao X R. Single sample path-based optimization of Markov chains [J]. J of Optimization Theory and Applications, 1999, 100(3): 527-548.

[4] Tang H, Xi H S, Yin B Q. Performance optimization of continuous-time Markov control processes based on performance potentials[J]. Int J of Systems Science, 2003, 34(1), 63-71.

[5] Puterman M L. Markov decision processes: Discrete stochastic dynamic programming [M]. New York: Wiley, 1994.

[6] 胡奇英,刘建庸. 马尔可夫控制过程引论[M]. 西安:西安电子科技大学出版社,2000.
(Hu Q Y, Liu J Y. An introduction to Markov decision processes[M]. Xi'an: Xidian University Press, 2000.)

[7] Cao X R. Semi-Markov decision problems and performance sensitivity analysis [J]. IEEE Trans on Automatic Control, 2003, 48(5): 758-769.

[8] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT Press, 1998.

[9] Bertsekas D P, Tsitsiklis J N. Neuro-dynamic programming[M]. Belmont: Athena Scientific, 1996.

[10] Tang H, Yuan J B, Lu Y, et al. Performance potential-based neuro-dynamic programming for SMDPs[J], Acta Automatic Sinica, 2005, 31(4): 642-645.

[11] 唐昊,周雷,袁继彬. 折扣平均准则 MDP 基于 TD(0)学习的统一 NDP 方法[J]. 控制理论与应用,2006, 23(2):292-296.
(Tang H, Zhou L, Yuan J B. Unified NDP method based on TD(0) learning for both average and discounted Markov decision processes [J]. Control Theory and Applications, 2006, 23(2):292-296.)

[12] 殷保群,奚宏生,周亚平. 排队系统性能分析与 Markov 控制过程[M]. 合肥:中国科学技术大学出版社,2004.
(Ying B Q, Xi H S, Zhou Y P. Queueing system performance analysis and Markov control processes [M]. Hefei: Press of University of Science and Technology, 2004.)

[13] Singh S, Sutton R S. Reinforcement learning with replacing eligibility traces [J]. Machine Learning, 1996, 22(1/2/3):123-158.

(上接第 154 页)

[12] Du C, Xie L, Soh Y C. H_∞ filtering of 2-D discrete systems [J]. IEEE Trans on Signal Processing, 2000, 48(6): 1760-1768.

[13] Gao H, Lam J, Wang C, et al. H_∞ mode reduction for uncertain two-dimensional discrete systems [J]. Optimal Control Applications Methods, 2005, 26(3): 199-227.

[14] Gao W B, Wang Y F, Homaifa A. Discrete-time variable structure control system [J]. IEEE Trans on Industrial Electronics, 1995, 42(2): 117-122.