

一种求解动态及不确定性优化问题的新方法

刘 晓

(中航工业西安航空计算技术研究所, 陕西 西安 710065)

摘 要: 动态和不确定性优化问题, 是许多智能优化算法面临的巨大挑战. 介绍了一种可求解这类问题的连续动作学习自动机(Continuous-Action Learning Automaton, CALA). 该自动机利用一个可变区间作为其动作集, 并依照均匀分布方式产生输出动作. 根据一个滑动窗口内的最佳的历史动作, 对区间的两个端点进行更新. 通过两个仿真实验, 演示了该算法在时变的随机环境下的优异性能. 相对于三种传统的 CALA 算法, 新算法的学习精度、反应速度以及在最坏情况下的行为表现都非常出色.

关键词: 学习自动机; 连续动作学习自动机; 动态环境; 不确定性环境; 在线学习

中图分类号: TP18; TP202.7

文献标识码: A

文章编号: 1000-7180(2016)07-0083-06

A New Method for Solving Dynamic and Uncertain Optimization Problems

LIU Xiao

(AVIC Xi'an Aeronautics Computing Technique Research Institute, Xi'an 710065, China)

Abstract: Dynamic and uncertain optimization problems pose a serious challenge to many intelligent optimization algorithms. A continuous action learning automaton (CALA) is introduced that can solve such problems. The automaton uses a variable interval as its action set, and generates actions with uniform distribution over this interval. The endpoints of the interval are updated according to the best historical action within a sliding window. Simulation results are presented to show the performance of the automaton in two time-varying stochastic environments. It is shown that the new algorithm exceeds three of the traditional CALA algorithms in the accuracy of learning, the rapidity of response, and the behaviors in the worst case.

Key words: learning automata; continuous-action learning automata; dynamic environments; uncertain environments; on-line learning

1 引言

工程技术领域的许多问题, 都可以归结为某种数值优化问题. 目前, 各种新型优化算法, 尤其是基于自然计算的智能优化算法, 如模拟退火遗传算法、蚁群算法、粒子群优化等进化类算法, 相关的研究非常活跃^[1-2]. 这些算法在解决静态、确定性优化问题方面十分有效. 然而, 实际应用中的问题常常是动态的, 或带有一定程度的不确定性. 与静态优化问题不同, 在动态优化中, 由于目标函数会随时间变化, 优化算法的任务不再是寻找一个固定的最优解, 而是

需要连续监视和跟踪最优解的变化. 对于不确定性优化问题, 因为目标函数含有噪声, 即使同一个备选解, 在不同的时间也可能有不同的函数值. 这就要求优化算法必须具有足够的抗干扰能力, 以发现在平均或数学期望意义上表现最好的解.

学习自动机(Learning Automata, LA)^[3-4]属于一种随机优化方法, 非常适合处理非线性、不确定性优化问题. 在任一时刻, 自动机根据某种概率分布, 从其动作集里选择一个动作(相当于备选解), 并输出给环境; 环境则反馈回一个标量信号(相当于目标函数), 作为对所选动作的评价. 根据环境的评价, 自

动机对其动作概率进行调整,以增大表现好的动作的被选概率.按照动作集的性质,LA 可以分为有限动作学习自动机(Finite-Action Learning Automata, FALA)和连续动作学习自动机(Continuous-Action Learning Automata, CALA)两大类^[4].对于实值参数优化问题,采用 CALA 最为合适,因为不必对动作空间进行离散化.传统的 CALA 共有四种:由 Santharam, Sastry 和 Thathachar^[4-6]提出的 CALA-SST;由 Beigy 和 Meybodi^[7]提出的 CALA-BM;由 Vlachogiannis^[8]提出的 R-CALA;由 Frost, Howell, Gordon 和吴青华^[9]提出的 CARLA(Continuous Action Reinforcement Learning Automata, 连续动作强化学习自动机).其中,前三种算法都采用基于高斯分布的参数化概率模型, CARLA 则采用一种非参数化的概率模型.

CALA-SST 和 CALA-BM 的动作集均为实数轴,算法根据所选动作及环境评价,对高斯分布的均值进行递推式更新. R-CALA 的动作集是一个有限区间,算法总保存所选过的全程最优动作,并以此作为高斯分布的均值.由于全程最优动作仅在其第一次被选择时评价一次,故该算法容易受噪声误导,也难以发现和跟踪环境的变化.关于高斯分布的标准差, CALA-SST 是根据所选动作及环境评价进行更新,并通过投影操作使之不小于一个预定值,以适应环境的动态变化. CALA-BM 和 R-CALA 则根据学习时间(迭代步数)来确定高斯分布的标准差,其数值单调减小并趋于 0,故不适合对时变环境的跟踪.与 R-CALA 类似, CARLA 的动作集也是一个有限区间.一开始,动作选择概率服从均匀分布;在学习过程中,算法利用一个高斯型的“邻近函数”,将表现好的动作的奖赏散布给相邻的动作.由于采用非参数化概率模型, CARLA 的实现非常复杂,需要进行大量的数值积分运算,空间开销也很大.

本文将介绍一种新的参数化 CALA,并研究其在求解动态、不确定性优化问题方面的潜力.我们将从解的精确度、反应的快速性以及最坏情况下的性能表现等方面,对该算法和传统的 CALA 算法进行实验比较.

2 新的 CALA

为克服传统 CALA 存在的不足,最近,我们提出一种新的参数化 CALA^[10-11],并将其应用于被噪声污染的多模态函数的全局优化以及系统辨识人工神经网络的训练.假设有一个未知的目标函数(如某

种损失函数或代价函数) $J(x)$.我们的任务是寻找使得 $J(x)$ 最小的最佳参数 x^* .该算法总保持一个可变区间 $I = [x_L, x_R]$ 和一个滑动窗口(或称历史信息记忆器) $W = \{(x(i), J(i)) \mid i = k-M+1, \dots, k\}$,其中 x_L 和 x_R 分别为区间的左右端点, $(x(i), J(i))$ 为最近的自动机与环境交互的历史记录(输出的动作参数和相应的损失函数), $x(i)$ 代表在 i 时刻自动机选出的动作参数, $J(i)$ 为 $J(x(i))$ 的简写.为高效实现, W 采用环形队列的方式存储最近的 M 对 x 和 J , M 为窗口的大小.

在任一时刻 k ,自动机以均匀分布方式在当前区间 I 上随机选择一个动作 $x(k)$,并观测其损失函数 $J(k)$.将 $x(k)$ 和 $J(k)$ 放入 W 中(若 W 内位置已满,淘汰掉最旧的一对记录).然后,找出窗口中具有最小 $J(x)$ 值的动作,记其为 x_b .最后,根据 x_b 对动作区间的端点(从而概率分布)进行更新.

假设 c_L 和 c_R 分别为将当前区间三等分的两个分割点,即 $c_L = x_L + \Delta$, $c_R = x_R - \Delta$, $\Delta = (x_R - x_L)/3$,则区间端点的更新规则如下:

$$\begin{aligned} x_L &= x_L - \lambda_1(c_L - x_b), \text{若 } x_b < c_L \\ &= x_L + \lambda_2(1 - \delta/\Delta)(x_b - c_L), \text{若 } x_b \geq c_L; \\ x_R &= x_R + \lambda_1(x_b - c_R), \text{若 } x_b > c_R \\ &= x_R - \lambda_2(1 - \delta/\Delta)(c_R - x_b), \text{若 } x_b \leq c_R \end{aligned}$$

式中, λ_1 , λ_2 和 δ 均为大于 0 的算法参数. λ_1 控制左端点向左移或右端点向右移(扩张)的步长, λ_2 控制左端点向右移或右端点向左移(收缩)的步长. λ_2 一般可取为 $\lambda_1/3$,以使收缩比扩张更谨慎一些. δ 则用于控制解的精度,并防止动作区间的长度收缩为 0,以使自动机永远保持对环境变化进行跟踪的能力.

该算法通过对动作区间进行“变焦”、“对焦”式调整,以追踪表现最好的动作,将其包围在区间的中心.故被称之为“聚焦区间学习自动机(Focused Interval Learning Automaton)”,简记为 FILA^[10-11].因学习算法始终关注一个滑动窗口,并将动作区间朝该窗口内的最佳动作 x_b 的方向调整(相当于对 x_b 进行“奖励”),故进一步将该算法记做“FILA/WR”,其中的“WR”代表“窗口(Window)”和“奖励(Reward)”^[10-11].

3 仿真实验

现有的进化算法的相关文献中,一般都是将动态环境和不确定性环境分开处理.本文研究的环境,则兼具动态性和不确定性.我们设计了两个测试问题,其中待最小化的目标函数(损失函数)均取二次

函数的形式.

问题1: $\min J(x, k) = (x - 10k/10\ 000)^2 + w_k$;

问题2: $\min J(x, k) = f(x, k) + w_k$, 其中

$f(x, k) = (x - 10)^2$, 当 $0 < k \leq 2\ 500$ 时

$= (x + 10)^2$, 当 $5\ 000 < k \leq 7\ 500$ 时

$= x^2$, 其他情况

在上述函数中, k 代表离散时间(仿真步数), 两个问题的最大仿真时间都取 10 000 步; w_k 为 0 到 1 之间均匀分布的随机数. 仿真开始前($k = 0$), 两个问题的最优参数都在 $x^* = 0$ 处. 问题1 相当于一个缓慢而连续变化的环境, 在仿真过程中(k 从 0 开始增加到 10 000), 其最优参数由 0 开始线性增大到 10. 问题2 则对应一个突变环境, 在仿真过程中其最优参数按照 $0 \rightarrow 10 \rightarrow 0 \rightarrow -10 \rightarrow 0$ 的顺序阶跃式变化.

分别用 FILA/WR 和三种传统的参数化 CALA, 即 CALA-SST, CALA-BM 和 R-CALA, 对上述两个问题进行仿真实验.

各算法都有一些内部参数, 其设置对仿真结果有很大的影响. 对每一种算法, 分别做了若干次尝试, 最后取综合效果最好的设置. 对 FILA/WR, 取 $M = 10$, $\lambda_1 = 0.15$, $\lambda_2 = 0.05$, $\delta = 0.2$. 对 CALA-SST, 取 $\lambda = 0.002$, $K = 20$, $\sigma_L = 0.3$. 对 CALA-BM, 用于更新高斯分布均值的步长因子 α , 取 0.3. 原始的 CALA-BM 没有针对高斯分布标准差的控制参数, 算法根据公式 $\sigma_k = 1/[\text{floor}(k/10)]^{1/3}$ 来计算 k 时刻高斯分布的标准差, 其中 floor 为下取整函数. 但这里存在两个问题, 一是当 $k < 10$ 时该公式会出现“除法溢出”, 二是其 σ_k 衰减过快. 为此, 我们将 floor 替换为上取整函数 ceil , 并引入一个新的参数 σ_0 , 按照 $\sigma_k = \sigma_0/[\text{ceil}(k/10)]^{1/3}$ 计算 σ_k , 使 σ_k 的大小可以被控制. R-CALA 可设置的内部参数只有两个, 即动作参数的上下限 $x_{\max}$ 和 $x_{\min}$. 这两个参数有两方面的作用: 一是限定动作参数的取值范围; 二是计算高斯分布的标准差, 具体算式为 $\sigma_k = [\Delta/(6k)]^{1/3}$, 其中 $\Delta = x_{\max} - x_{\min}$. 对于本文的问题, 这种“同一组参数, 两种作用”的体制并不合适: 为了得到较高的学习精度, σ_k 计算公式中的 Δ 不能太大, 否则无异于漫无边际的随机搜索; 另一方面, 这里的目标函数是动态变化的, 其理论最优解的变化范围很宽, 这又要求动作参数的取值范围不能太窄. 因此, 将两种用途的参数分离开来, 其中用于控制动作参数取值范围的上下限, 在问题1 中取 11 和 -1, 在问题2 中取 11 和 -11; 用于计算 σ_k 的 Δ , 则另行设置. FILA/WR 的初始动作区间的两个端点, 取 $x_L = -0.4$, $x_R = 0.4$. CALA-SST 的高斯分布的初

始均值和标准差, 分别取 $\mu_0 = 0$, $\sigma_0 = 0.5$. 对于 CALA-BM, 高斯分布的初始均值 μ_0 取 0, 我们新引入的、控制高斯分布标准差的参数 σ_0 取 2. 对于 R-CALA, 用于计算 σ_k 的参数 Δ 取 1.5.

对每一个问题, 分别用上述四种算法各做 100 次仿真实验. 图 1 显示了针对问题1 的, 仿真过程中自动机输出的动作参数 $x(k)$ 随仿真时间 k 的变化关系. 其中每一张图, 都是将相应算法 100 次仿真结果“叠加”在一起的效果. 为便于参照比较, 图中还给出了理论最优参数的变化轨迹(虚线).

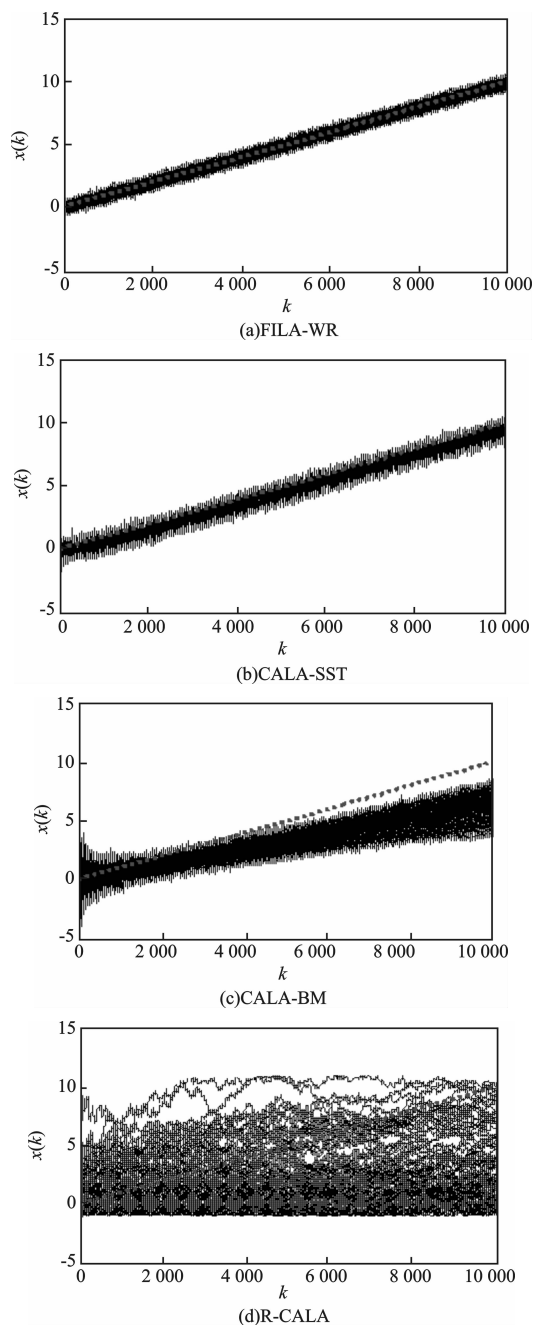


图1 渐变环境下的参数跟踪轨迹

由图 1 可以看出,对于环境连续而缓慢的变化, FILA/WR 和 CALA-SST 都能够跟踪. 不过, CALA-SST 的参数轨迹要粗一些,即选出的动作波动范围较大. 另外, FILA/WR 的动作轨迹几乎将理论最优动作(虚线)围在中心,而 CALA-SST 的动作轨迹则明显偏下,这表明其对最优动作的跟踪较为滞后. 相比之下, CALA-BM,尤其是 R-CALA,表现则非常差. CALA-BM 有跟踪的趋势,但其动作轨迹未能与理论最优动作很好地重合,甚至不平行,“轨迹带”也相当宽. 增大算法参数 α 和 σ_0 , 可以加快跟踪速度,但轨迹带将会更宽,导致性能进一步恶化;减小 α 和 σ_0 , 轨迹带会变窄,但方向却更倾向于水平,跟踪效果更差. 表现最差的是 R-CALA,其动作轨迹十分杂乱.

静态优化是一种离线优化,一般只需关心最终找到的解的质量,搜索过程中遇到过的解的表现无关紧要. 动态优化则是一种在线优化,优化过程中的每个解都应该考虑,对于某些实时控制问题,还应当考虑系统的稳定性、可靠性和安全性. 算法在探索和尝试时,不能产生太差的解(即使偶尔产生一次),因为一次危险的、太差的尝试,有可能导致灾难性的后果. 目前,在进化算法的相关研究中(如文献[12]),几乎都是采用平均离线性能指标,来评价动态环境下优化算法的性能表现的. 这种指标,一方面仅考虑每一代群体中最好的解,另一方面是将多次仿真结果取平均,忽略或回避了在线应用时的安全性风险. 与此不同的是,本文则关注算法运行中所产生的所有尝试解,尤其是其中最差的解. 图 2 给出的是针对问题 1 的,四种算法各 100 次仿真中每个时间步(k)上出现过的损失函数的最大值(即最坏情况).

由图 2 可以看出, FILA/WR 和 CALA-SST 的最大损失函数所形成的轨迹都呈水平走向. FILA/WR 的轨迹基本上处于 1.05 到 1.5 之间,最大值为 1.80. CALA-SST 的轨迹则更高,包络也更宽,除开始的一小段外基本上都在 1.2 之上,最大值则达到 4.31. CALA-BM 和 R-CALA 的最大损失函数都随时间上升,幅度也高得多,前者最高达 43.73,后者更是达到 121.45.

在理想的、完美跟踪的情况下,损失函数应始终是 0,但由于加入了 0 到 1 之间的随机噪声,故实际值达到 1.0 也是有可能的. 因此可以将 1.0 作为一个基准,根据超出它的程度来评价算法的性能. 这样, FILA/WR 的最大损失函数超出基准值的最大

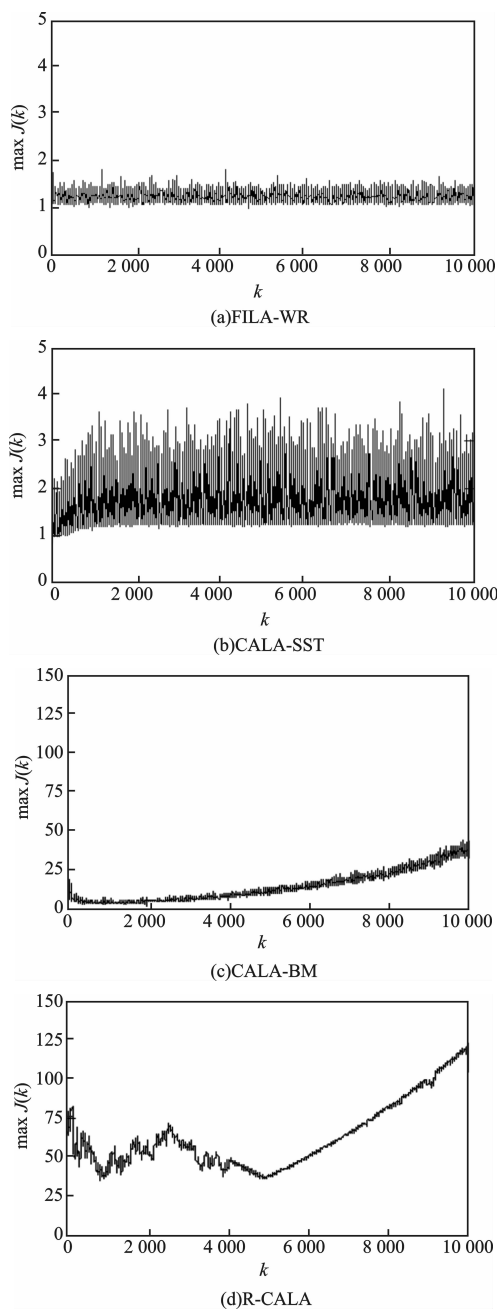


图 2 渐变环境下的最大损失函数

幅度为 0.80, CALA-SST, CALA-BM 和 R-CALA 的最大超出幅度则分别为 3.31, 42.73 和 120.45, 分别为 FILA/WR 的 4.1 倍、53.4 倍和 150.6 倍.

再来看环境阶跃式变化的情形(即问题 2). 图 3 给出的是针对问题 2 的,四种算法各 100 次仿真所产生的 $x(k)$ 曲线的叠加效果.

由图 3 可以看出,对于环境的阶跃式变化, FILA/WR 和 CALA-SST 基本上都能正确反应,但 CALA-SST 的效果要差一些. 首先, CALA-SST 跟踪速度缓慢,其参数轨迹与理论最优值(虚线)重合的部分较短;其次, CALA-SST 的轨迹较粗,上面还

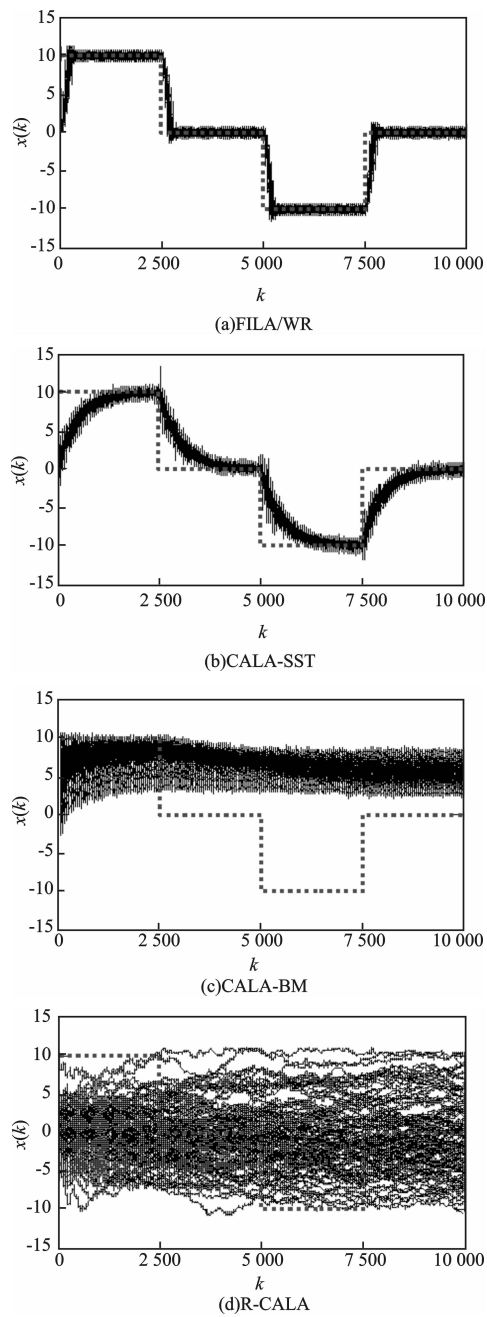


图 3 突变环境下的参数跟踪轨迹

有一些毛刺,尤其是在环境突变处出现幅度很高的尖刺. FILA/WR 则没有这些问题,其参数跟踪迅速,轨迹陡直,上面也没有过大的尖刺. CALA-BM 对环境的第一次跳变($0 \rightarrow 10$)有反应,但跟踪不到位,轨迹上沿刚到 10.0,中轴线只到 5.0 多一点;对于后面的三次跳变,反应则不明显,动作参数基本上分布于 2.5 到 8.5 之间,轨迹包络很宽.表现最差的仍是 R-CALA,其基本上不能感知环境的变化,动作轨迹膨胀而杂乱,介于 -10.0 到 11.0 之间.

图 4 是针对问题 2 的,四种算法各 100 次仿真中最大损失函数随时间变化的情况.

由图 4 可以看出,对应环境的四次跳变,FILA/WR 和 CALA-SST 的最大损失函数各有四次短暂的起伏,不过具体细节不大一样. FILA/WR 形成四个很窄的“小齿”,其下降速度非常快(375 步之前就已下降到最小值),CALA-SST 的“齿”要大一些,下降速度越到后期越缓慢(1 250 步之后才下降到最小值). CALA-BM 和 R-CALA 的表现则截然不同. CALA-BM 对第一次跳变有一定的反应,但其最大损失函数的下降并不彻底(最低都在 28.0 之上).此后因动作参数基本固定在 5.0 附近,故对于第二次和第四次跳变(理论最优动作都是 0),其最大损

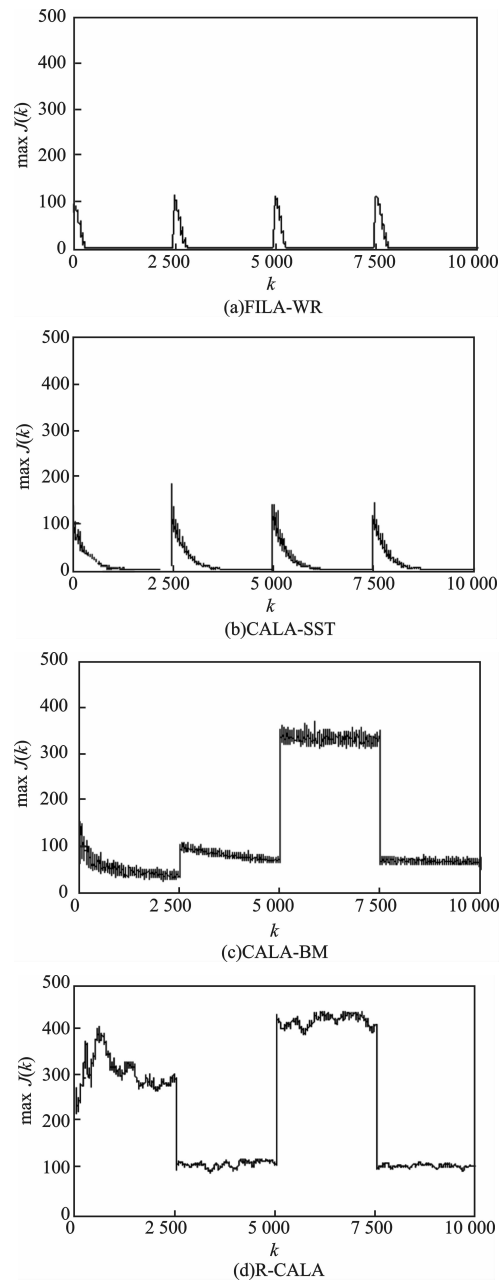


图 4 突变环境下的最大损失函数

失函数侥幸地处于相对较低的水平;但对于第三次跳变(理论最优动作为-10),最大损失函数则阶跃式上升并形成-一个明显的“高台”。至于 R-CALA,相应于环境的四次跳变,其最大损失函数基本上也呈阶跃式变化,其中第一次和第三次跳变也形成了两个高台,两个较低的“平台”(对应第二次和第四次跳变),高度也在 100.0 左右。

在环境跳变瞬间,损失函数突然升至 100 显然是难以避免的,因为自动机不可能预知环境要变化,只有当发现反馈信号有变化后才能做出反应。由于目标函数叠加了 0 到 1 之间的随机噪声,故其最大值实际上可以达到 101.0。因此,可以将 101.0 作为性能恶化的基准。在图 4 中,FILA/WR 的四个小齿的顶端都不是很高,最高的一个为 116.10,超出基准 14.95%。CALA-SST 的齿则更高,最高的一个尖峰达到 186.45,超出基准 84.60%。CALA-BM 和 R-CALA 的最大损失函数最高分别为 369.51 和 441.98,超出基准 265.85%和 337.60%。

综上分析,可以得出结论:不论对渐变随机环境,还是突变随机环境,FILA/WR 的性能都远远优于三种传统的参数化 CALA。其跟踪速度快、精度高,即使在最坏情况下也有相当不错的表现。

4 结束语

动态及不确定性优化问题,是目前优化和学习算法领域中-一个热点研究课题。本文介绍了一种新的参数化 CALA,即基于对一个滑动窗口内最佳的历史动作进行奖励的聚焦区间学习自动机(FILA/WR),并将其应用于时变随机环境下的优化问题。通过两个仿真实验,对该算法和三种传统的参数化 CALA 进行了比较研究。结果表明,不论是连续渐变的随机环境,还是阶跃式突变的随机环境,FILA/WR 都能够很好地跟踪。在跟踪速度、跟踪精度以及最坏情况下的表现等方面,FILA/WR 均显著优于传统 CALA。下一步研究的方向是,对 FILA/WR 和进化算法在动态不确定性环境下的性能进行仿真比较,以及将新算法应用于 PID 参数的在线整定等实际工程问题。

参考文献:

- [1] 杨久红,王小增.改进的混合粒子群算法[J].微电子学与计算机,2012,29(5):170-173.
- [2] 谢朝政,樊晓光,嵯真福.引入模式搜索算子的粒子群优化算法[J].微电子学与计算机,2015,32(2):97-99.
- [3] Narendra K S,Thathachar M A L. Learning automata: An introduction[M]. Englewood Cliffs, NJ: Prentice Hall,1989:35-58.
- [4] Thathachar M A L,Sastry P S. Varieties of learning automata: An overview[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2002, 32(6):711-722.
- [5] Santharam G,Sastry P S,Thathachar M A L. Continuous action set learning automata for stochastic optimization[J]. Journal of the Franklin Institute, 1994, 331(5):607-628.
- [6] Sastry P S,Nagendra G D,Manwani N. A team of continuous-action learning automata for noise-tolerant learning of half-spaces[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2010, 40(1):19-28.
- [7] Beigy H,Meybodi M R. A new continuous action-set learning automaton for function optimization[J]. Journal of the Franklin Institute, 2006, 343(1):27-47.
- [8] Vlachogiannis J G. Probabilistic constrained load flow considering integration of wind power generation and electric vehicles[J]. IEEE Transactions on Power Systems, 2009, 24(4):1808-1817.
- [9] Howell M N, Frost G P, Gordon T J, et al. Continuous action reinforcement learning applied to vehicle suspension control [J]. Mechatronics, 1997, 7(3):263-276.
- [10] 刘晓.一种鲁棒的连续动作学习自动机[C]//全国抗恶劣环境计算机第二十五届学术年会论文集. 银川, 2015:266-271.
- [11] 刘晓.基于智能计算的非线性系统辨识与建模[C]//全国抗恶劣环境计算机第二十五届学术年会论文集. 银川, 2015:251-255.
- [12] 刘子坤,李枚毅,张晓.动态环境下运用对称位移映射的 PSO 算法[J]. 计算机工程, 2014, 40(11):200-204.

作者简介:

刘 晓 男,(1965-),高级工程师。研究方向为动力控制和智能计算。E-mail: xiaoliu@163.com.