

粒距反馈的 S 函数粒子群权值调整策略

左旭坤^{1*}, 苏守宝^{1,2}

(1. 皖西学院 信息工程学院, 安徽 六安 237012; 2. 哈尔滨工业大学 卫星技术研究所, 哈尔滨 150080)

(* 通信作者电子邮箱 zsk_land@163.com)

摘要:针对标准粒子群优化(PSO)算法把惯性权值作为全局参数,很难适应复杂的非线性优化的问题,提出了一种基于粒距和 S 型函数的粒子群权值调整策略(SFIW)。利用 S 型函数能够在非线性和线性之间平滑过渡的特性,构造了基于 Logistic 方程的惯性权值函数。在优化过程中根据每个粒子的粒距大小,调整每个粒子的惯性权值函数的非线性系数,使得粒距较大的粒子获得较大的惯性权值、粒距较小的粒子获得较小的惯性权值,从而平衡算法的局部开发和全局探测能力。最后,通过对基准函数的仿真并与其他 PSO 算法比较,验证了算法的有效性和可行性。

关键词:粒子群优化;惯性权值;粒距;S 型函数;局部开发;全局探测

中图分类号: TP301.6 **文献标志码:** A

Sigmoid inertia weight adjustment strategy with particle spacing feedback for PSO

ZUO Xu-kun^{1*}, SU Shou-bao^{1,2}

(1. Faculty of Information Engineering, West Anhui University, Lu'an Anhui 237012, China;

2. Research Center of Satellite Technology, Harbin Institute of Technology, Harbin Heilongjiang 150080, China)

Abstract: Concerning the problem that the standard Particle Swarm Optimization (PSO) algorithm in which inertia weight is global parameter, and cannot adapt to the complex and nonlinear optimization process, a Spacing Feedback Inertia Weight (SFIW) was proposed. Taking advantage of the characteristic that Sigmoid function can make the smooth transition between linear and nonlinear, an inertia weight function based on Logistic equation was constructed. In the process of optimization, the nonlinear coefficient of inertia weight function was adjusted according to the particle spacing to make the particle with longer particle spacing get larger inertia weight and make the particle with shorter particle spacing get smaller inertia weight. Therefore, the local exploitation and global exploration get balanced. Finally, the experimental results on several benchmark functions and the comparison with other algorithms show the effectiveness and feasibility of the SFIW-PSO.

Key words: Particle Swarm Optimization (PSO); inertia weight; particle spacing; Sigmoid function; local exploitation; global exploration

0 引言

粒子群优化(Particle Swarm Optimization, PSO)算法的思想来源于鸟群等社会群体生物的觅食现象,最初由 Kennedy 等^[1]于 1995 年提出。由于 PSO 具有算法简单、易于实现、不需要求解问题的梯度信息、参数少等优点^[2],因此被广泛应用于科学研究、工程实践和管理等诸多领域^[3]。

然而,PSO 算法有易陷入局部极值、过早收敛和收敛性能差等缺点^[4]。不少学者为解决这一问题,从 PSO 的参数调整策略入手,提出了很多有效的改进算法^[5-6]。其中,惯性权值是 PSO 算法中非常重要的一个参数,通过它控制粒子速度可以调整算法的局部开发和全局探测能力^[7]。较大的权值有利于 PSO 的全局探测,较小的权值有利于 PSO 的局部开发^[8]。为了较好地平衡这两种能力,先后出现了惯性权值的线性调整策略^[9]、非线性调整策略^[10]、S 型函数调整策略^[11]和自适应 S 型函数调整策略^[12]等。借鉴上述文献思想,本文提出了一种基于粒距反馈的 S 型函数权值调整策略(Spacing Feedback Inertia Weight, SFIW),并通过基准测试函数验证了算法的有效性和可行性。

1 PSO 算法及常见权值调整策略

1.1 PSO 算法介绍

PSO 算法首先初始化种群随机粒子,每个粒子都代表着优化问题的可能解。粒子 i 的信息可以用 D 维向量表示,位置表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$,速度表示为 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 。每一次进化,粒子通过跟踪两个“极值”来更新自己。一个是粒子本身所找到的最优解,用 $pbest$ 表示其坐标。另一个是整个种群目前找到的最优解,用 $gbest$ 表示其坐标。找到这两个最优值时,粒子根据如下的公式来更新自己的速度和位置:

$$v_{ij}(t+1) = \omega \times v_{ij}(t) + c1 \times r_1 \times [pbest_{ij}(t) - x_{ij}(t)] + c2 \times r_2 \times [gbest_j(t) - x_{ij}(t)] \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t) \quad (2)$$

其中: $v_{ij}(t)$ 是粒子 i 在第 t 次进化中第 j 维的速度; $x_{ij}(t)$ 是当前粒子的位置; $r_{1,2}$ 是 $(0,1)$ 的随机数,用来模拟自然群体中的轻微扰动; $c1, c2$ 是学习因子,分别表示粒子对自身记忆的依赖程度和其他粒子对它的影响; ω 是惯性权值,体现粒子对自身速度的保留程度。

收稿日期:2012-04-05;修回日期:2012-05-23。

基金项目:国家自然科学基金资助项目(61075049);安徽高校省级自然科学基金项目(KJ2010B467)。

作者简介:左旭坤(1978-),男,安徽六安人,讲师,硕士,主要研究方向:计算机测控、智能控制; 苏守宝(1965-),男,安徽六安人,教授,博士,主要研究方向:群体智能、模式识别。

1.2 线性递减权值和非线性递减权值

文献[9]提出了惯性权值线性调整策略 (Linearly Decreased Inertia Weight, LDIW), 也称标准 PSO。 ω 按式(3)线性递减:

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \frac{t_{\max} - t}{t_{\max}} \quad (3)$$

其中: t 为当前进化次数; t_{\max} 为进化总次数; ω_{\max} , ω_{\min} 分别为最大、最小权值, 并建议分别取值 0.95 和 0.4。在式(3)的控制下, 算法前期有较好的全局搜索能力, 后期加强局部开发能力。

文献[10]提出了惯性权值非线性调整策略 (Nonlinear Decreased Inertia Weight, NDIW), 该算法将式(3)改为:

$$\omega(t) = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \left(\frac{t_{\max} - t}{t_{\max}} \right)^n \quad (4)$$

其中 n 是非线性指数, 可以更好地平衡算法的全局搜索和局部开发, 提高算法的收敛速度和精度。 n 的值可以自适应调节。作者通过大量仿真实, 指出当 $n = 1.2$ 时算法精度最高, 若无法确定 n 的取值, 建议在 $[0.9, 1.3]$ 上随进化次数按 0.1 步进取值。

文献[11]利用神经网络中常用的 S 型激活函数在线性和非线性之间良好平衡能力, 提出了基于 Sigmoid 函数的非线性权值调整策略 (Sigmoid Inertia Weight, SIW)。该策略在群聚集度较大时采用 LDIW, 否则利用式(5)调节惯性权值。

$$\omega(t) = \frac{1}{1 + \exp \left[\frac{(\ln 1.5 - \ln 19) \times t - \ln 19}{t_{\max}} \right]} \quad (5)$$

文献[12]提出了一种自适应 Sigmoid 函数的非线性权值调整策略 (Adaptive Sigmoid Inertia Weight, ASIW)。该策略在 S 型函数中增加了两个调节算子, 从种群进化速度和聚合度两方面出发, 调节 S 型函数的平滑性, 在一定程度上解决了收敛性和全局性两者之间的矛盾, 如式(6)所示:

$$\begin{cases} \alpha = f_{\text{best}}(t) / f_{\text{best}}(t-1) \\ \beta = f_{\text{best}}(t) / f_{\text{avg}}(t) \\ \omega(t) = \frac{1}{1 + \exp \left[\frac{1.2\alpha(\ln 2.2 - \ln 19)t - \ln 19}{\beta t_{\max}} \right]} \end{cases} \quad (6)$$

其中 f_{best} 和 f_{avg} 分别表示当前的全局最优值和适应度平均值, 则 α 和 β 可作为衡量种群进化速度和聚合度的两个指标。

2 粒距反馈的 S 型函数权值调整策略

2.1 粒距与权值调整

本文利用 S 型曲线在线性和非线性之间平滑过渡的特性, 通过 Logistic 函数构造以下的权值调整公式:

$$\omega(t) = \frac{1}{1 + \exp \left[\ln 1.5 - \ln 19 \times \left(\frac{t_{\max} - t}{t_{\max}} \right)^{\varepsilon} \right]} \quad (7)$$

该公式能够使得 $\omega \in (0.95, 0.4)$, 满足文献[9]的结论。式中的 ε 用来调节权值的下降速度, 如图 1 所示。

由图 1 可知, ε 越大, 惯性权值下降越快, 进化同期的权值越小, 对应的粒子速度也就越小。

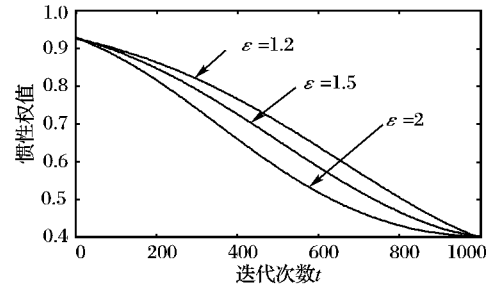


图 1 不同 ε 下的惯性权值变化曲线

规定 t 次进化时第 i 个粒子的粒距为该粒子在各维上与当前最优点距离的均方根, 如式(8)所示:

$$d_i(t) = \sqrt{\frac{1}{D} \sum_{j=1}^D (x_{ij}(t) - gbest_j(t))^2} \quad (8)$$

由 PSO 算法本质可知, 粒距小的粒子可取较小的惯性权值, 减速进行局部开发; 粒距大的粒子取较大的惯性权值, 加速进行全局探测。本文利用式(9)根据每个粒子的粒距选择 ε 从而控制粒子速度:

$$\varepsilon_i(t) = \varepsilon_{\min} + (\varepsilon_{\max} - \varepsilon_{\min}) \times \left(\frac{d_{\max}(t) - d_i(t)}{d_{\max}(t) - d_{\min}(t)} \right) \quad (9)$$

其中: ε_{\max} 和 ε_{\min} 是 ε 的取值范围, 本文中分别取 2 和 1.1。 $d_{\max}(t)$ 和 $d_{\min}(t)$ 是当前最大和最小粒距。从式(7)~(9)可看出, 当 $d_i(t)$ 越大时, $\varepsilon_i(t)$ 越小, 该粒子的惯性权值越大, 其速度越大; 反之亦然。每个粒子根据自身粒距选择合适的惯性权值从而平衡整个种群的开发和探测能力。

2.2 变异操作

尽管根据粒距反馈调整惯性权值能够增强探测能力, 但经过一定次数进化后, 仍会发生早熟收敛^[13]。主要原因是此时粒子的飞行速度越来越小, 粒子的有序性在很大程度上增强了, 这样不利于全局搜索^[14]。因此, 本文引入变异操作来增加粒子的混乱度, 以保持种群的多样性, 避免陷入局部最优。具体操作如下。

$$\begin{cases} a = f_{\text{best}}(t) / f_{\text{best}}(t-1) \\ \text{if } \text{rand}(1) < a \\ gbest = gbest \times (1 + \text{Gauss}(0, 1)) \end{cases} \quad (10)$$

其中 a 是种群的变异概率, 由 PSO 算法原理可知 $f_{\text{best}}(t) \leq f_{\text{best}}(t-1)$, 因此 $a \in [0, 1]$ 。 a 变大, 说明算法的进化速度变慢, 算法陷入局部最优的风险增大, 此时种群变异的概率增大。变异时, 对全局最优值选择一个服从 $\text{Gauss}(0, 1)$ 分布的随机数进行扰动, 提高其带领种群逃离局部最优的可能性。

2.3 SFIW 算法流程

基于粒距反馈的 S 函数权值调整 PSO 算法可归纳如下。

- 步骤 1 确定种群规模 N 和维数 D , 初始化粒子群 $pbest$ 和 $gbest$ 。
- 步骤 2 根据式(10)进行变异操作。
- 步骤 2 根据式(7)~(9)计算各粒子惯性权值。
- 步骤 4 根据式(1)和式(2)更新粒子速度和位置。

步骤 5 评价粒子适应度,更新 $pbest$ 和 $gbest$,判断是否满足停止进化的条件。若是则转步骤 7。

步骤 6 进化次数加 1,转步骤 2。

步骤 7 输出 $gbest$ 的适应度值,算法结束。

3 实验分析

3.1 标准测试函数

选取 Benchmark 的 3 个常用函数对 LDIW、NDIW、SIW、ASIW 和 SFIW 策略进行对比测试,如表 1 所示。其中 f_1 函数是一个单峰函数,只有一个全局最优解,整个优化过程是平滑过渡到最优解; f_2 函数是多级函数,优化过程呈螺旋向四周扩散; f_3 函数是一个多峰函数,在唯一的全局最优解四周有多个半球形的局部最优解^[15]。基准函数如下所示。

Sphere 函数:

$$f_1(x) = \sum_{i=1}^D x_i^2; x \in [-30, 30]$$

Ackley 函数:

$$f_2(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}\right) - \exp\left[\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)\right] + 20 + e;$$

$x \in [-30, 30]$

Rastrigin 函数:

$$f_3(x) = \sum_{i=1}^D [x_i^2 - 10\cos(2\pi x_i) + 10];$$

$x \in [-5.12, 5.12]$

所有算法参数均一致,即: $\omega_{\min} = 0.4, \omega_{\max} = 0.95, c1 = c2 = 1.7, t_{\max} = 1000$,维数 D 分别取 10 和 30,种群规模 S 取 20,所有实验均重复 50 次。

3.2 测试结果及分析

从表 1~3 可以看出,低维数时 NDIW 和 SIW 算法性能相当,精度和一致性均略差于 ASIW,略优于 LDIW。高维度时四种算法差异减小,性能均大幅下降,说明它们处理复杂优化过程的能力较差。SFIW 不论在低维还是高维,算法精度和一致性都明显优于上述四种算法。实验结果表明,SFIW 有较强的全局探测和处理复杂优化过程的能力。

表 1 Sphere 函数测试结果

维数	算法	最优值	平均值	标准差
10	LDIW	0.006 2	0.660 9	0.937 4
	NDIW	0.006 5	0.520 4	0.630 4
	SIW	0.007 9	0.455 7	0.597 5
	ASIW	0.002 8	0.293 8	0.419 2
	SFIW	0	3.927 2E-34	2.654 9E-33
30	LDIW	64.966 6	476.711 3	500.662 1
	NDIW	61.307 2	439.242 8	447.283 2
	SIW	63.805 1	414.162 9	440.768 6
	ASIW	23.120 5	220.005 0	303.222 5
	SFIW	9.290 4E-19	8.569 7E-14	2.806 6E-13

限于篇幅,本文只给出各函数 10 维的优化过程,如图 2~4 所示,30 维优化过程类似。从图中可看出,五种算法在进化的初期性能相当。进化 200 次后,其他四种算法很快陷入局

部最优早熟收敛,只有 SFIW 在整个优化过程中一直保持较好的全局探测能力,寻优精度较高。

表 2 Ackley 函数测试结果

维数	算法	最优值	平均值	标准差
10	LDIW	0.740 0	2.289 0	1.063 2
	NDIW	0.585 5	2.477 4	0.979 4
	SIW	0.842 1	2.234 7	0.910 6
	ASIW	0.598 5	2.000 8	0.634 3
	SFIW	0	2.897 1E-9	1.449 0E-8
30	LDIW	6.987 2	11.313 3	2.703 1
	NDIW	5.824 6	11.095 0	2.452 3
	SIW	5.606 5	10.128 1	2.427 6
	ASIW	4.455 4	8.088 6	3.063 7
	SFIW	2.296 3E-9	0.332 0	0.741 0

表 3 Rastrigin 函数测试结果

维数	算法	最优值	平均值	标准差
10	LDIW	4.432 0	20.265 8	10.760 2
	NDIW	3.012 9	17.531 8	10.516 5
	SIW	1.424 3	17.035 9	9.673 2
	ASIW	7.199 8	16.115 2	8.603 1
	SFIW	0	8.109 2	4.838 1
30	LDIW	109.645 3	163.101 2	27.310 0
	NDIW	94.806 0	157.528 4	28.894 8
	SIW	83.201 3	153.322 0	31.458 5
	ASIW	96.541 1	145.364 6	20.148 3
	SFIW	40.853 7	74.659 0	21.859 5

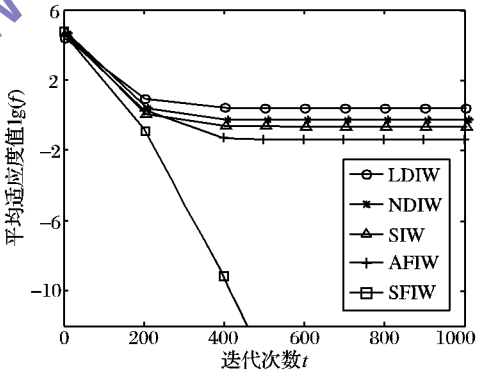


图 2 Sphere 函数优化过程(10 维)

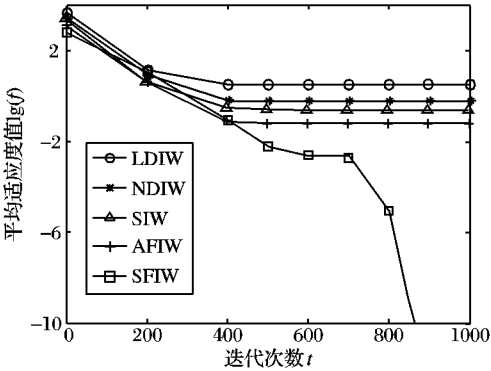


图 3 Ackley 函数优化过程(10 维)

4 结语

本文提出了一种粒距反馈的 S 型函数 PSO 权值调整策略 SFIW。针对每个粒子粒距的不同,选取不同下降速度的 S 型权值函数,自适应地调整每个粒子的速度,从而形成了一种

闭环 PSO 控制。从仿真结果来看, SFIW 算法能有效地逃离局部最优, 提高寻优精度。与标准 PSO 相比, SFIW 算法只增加了粒距计算、单个粒子变异等操作, 并未过分加大算法的复杂度, 因此该算法是一种有效、可行的新型 PSO 全局算法。

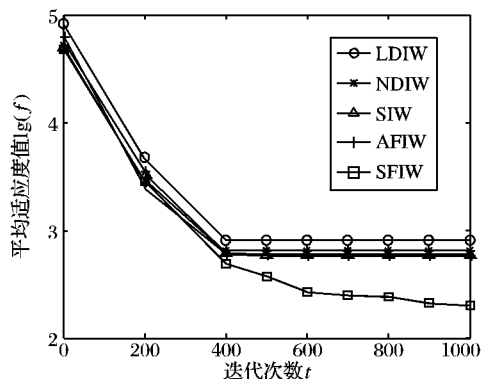


图 4 Rastrigin 函数优化过程(10 维)

参考文献:

- [1] KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proceedings of 1995 IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Service Center, 1995: 1942 - 1948.
- [2] 杨雪榕, 梁加红, 陈凌, 等. 多邻域改进粒子群算法[J]. 系统工程与电子技术, 2010, 32(11): 2453 - 2458.
- [3] 吴宪祥, 郭宝龙, 王娟. 基于 Lotka-Volterra 模型的双群协同竞争粒子群优化算法[J]. 控制与决策, 2010, 25(11): 1619 - 1624.
- [4] 孙越泓, 魏建香, 夏德深. 一种基于粒子对称分布多样性的 PSO 算法[J]. 模式识别与人工智能, 2010, 23(2): 137 - 144.
- [5] 金义雄. 基于粒子群算法的输电网优化规划研究[D]. 上海: 上海交通大学, 2006.
- [6] 刘志雄, 梁华. 粒子群算法中随机数参数的设置与实验分析[J]. 控制理论与应用, 2010, 27(11): 1489 - 1496.
- [7] SHI Y, EBERHART R C. A modified particle swarm optimizer [C]// Proceedings of IEEE International Conference on Evolutionary Computation. Washington, DC: IEEE, 1998: 69 - 73.
- [8] 黄翀鹏, 熊伟丽, 徐保国. 惯性权值对粒子群算法收敛性的影响及改进[J]. 计算机工程, 2008, 34(12): 31 - 33.
- [9] SHI Y, EBERHART R C. Empirical study of particle swarm optimization[C]// Proceedings of the 1999 Congress on Evolutionary Computation. Washington, DC: IEEE, 1999: 1945 - 1950.
- [10] CHATTERJEE A, SIARRY P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers & Operations Research, 2006, 33(3): 859 - 871.
- [11] 田东平, 赵天绪. 基于 Sigmoid 惯性权值的自适应粒子群优化算法[J]. 计算机应用, 2008, 28(12): 3058 - 3061.
- [12] 黄利, 杜伟伟. 基于 Sigmoid 惯性权值自适应调整的粒子群优化算法[J]. 计算机应用研究, 2012, 29(1): 32 - 34.
- [13] 陈民铨, 张聪誉, 罗辞勇. 自适应进化多目标粒子群优化算法[J]. 控制与决策, 2009, 24(12): 1851 - 1855.
- [14] 任子晖, 王坚. 加速收敛的粒子群优化算法[J]. 控制与决策, 2011, 26(2): 201 - 206.
- [15] 李旭芳, 王士同. 一种自适应粒子群算法[J]. 系统仿真学报, 2008, 21(9): 2582 - 2586.
- [16] GONG C, WENBO H, JIE L, et al. Energy-aware server provisioning and load dispatching for connection-intensive Internet services [C]// Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2008: 337 - 350.
- [17] MARCIO F, ALAN S, RICHARD W, et al. Adaptive performance prediction for distribute data-intensive applications [C]// Proceedings of the 1999 ACM/IEEE Conference on Supercomputing (CDROM). New York: ACM, 1999: 36 - es.
- [18] AMAN K, FENG Z. Fine-grained energy profiling for power-aware application design [J]. ACM SIGMETRICS Performance Evaluation Review, 2008, 36(2): 26 - 31.
- [19] BARATH R, KASHI V, SRIRAM R, et al. Cloud control with distributed rate limiting [C]// Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. Kyoto, Japan: ACM Press, 2007: 337 - 348.
- [20] NORMAN W P, MARCELO A T, KEVIN L, et al. Optimizing utility in cloud computing through autonomic workload execution [J]. IEEE Data Engineering Bulletin, 2009, 32(1): 51 - 58.
- [21] SHEKHAR S, AMAN K, FENG Z. Energy aware consolidation for cloud computing [C]// Proceedings of the 2008 Conference on Power Aware Computing and Systems. Berkeley, CA, USA: USENIX Association, 2008: 10 - 10.
- [22] MARTIN F A, CAREY L W. Web server workload characterization: the search for invariants [C]// Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. Philadelphia, Pennsylvania, United States: ACM Press, 1996: 126 - 137.
- [23] Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles [EB/OL]. [2012 - 03 - 10]. <http://www.broadinstitute.org/gsea/index.jsp>.
- [24] BLAST [EB/OL]. [2012 - 03 - 10]. <http://blast.ncbi.nlm.nih.gov/Blast.cgi>.
- [25] HWANG K, XU Z. Scalable parallel computing: Technology, architecture, programming [M]. New York: McGraw-Hill, 1998.
- [26] 程朝辉, 李伟, 涂亚明. 软件静态度量和预测模型研究[J]. 计算机工程与应用, 2005, 41(24): 118 - 120.
- [27] HONG L T, SCHAHRAM D. Composible cost estimation and monitoring for computational applications in cloud computing environments [J]. Procedia Computer Science, 2010, 1(1): 2175 - 2184.
- [28] 胡亮, 车喜龙, 唐阔. 网络应用程序的性能预测策略[J]. 吉林大学学报: 理学版, 2005, 43(6): 794 - 798.
- [29] LIONEL L, LAGAR-CAVILL H, DAVID L. Computer meteorology: Monitoring compute clouds [C]// Proceedings of the 12th Conference on Hot Topics in Operating Systems. Monte Verità, Switzerland: ACM Press, 2009: 4 - 4.
- [30] JEROME B, ANDY K, RUNPING Q, et al. Chukwa: A large-scale monitoring system [C]// CCA'08: Cloud Computing and its Applications. Chicago, IL: [s. n.], 2008: 1 - 5.
- [31] 葛君伟, 张博, 方义秋. 云计算环境下的资源监测模型研究[J]. 计算机工程, 2011, 37(11): 31 - 33.

(上接第 2723 页)