

求解非光滑优化问题的改进大洪水算法

张清叶, 高 岩, 马 良

(上海理工大学 管理学院, 上海 200093)

摘要: 应用启发式算法求解非光滑优化问题, 解决基于次梯度信息的确定性算法在求解时困难较大的问题. 首先分析了基本大洪水算法的优化机理及特征并给出其求解步骤, 然后针对无约束及盒子约束问题分别设计了改进的大洪水算法, 将基本大洪水算法所依赖的参数 up 省去. 对于无约束情形, 提出了进行邻域搜索的随机行走法; 对于盒子约束情形, 提出了选择初始可行点的方法和进行邻域搜索的混沌优化算法. 最后通过算例进行测试并与其他算法进行对比, 测试结果表明了改进的大洪水算法在求解非光滑优化问题时的有效性与优越性, 故其可作为求解非光滑优化问题的一种实用方法.

关键词: 大洪水算法; 非光滑优化; 随机行走; 混沌

中图分类号: O 224 **文献标志码:** A

Improved Great Deluge Algorithm for Nonsmooth Optimization Problems

ZHANG Qingye, GAO Yan, MA Liang

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Since nonsmooth optimization problems are difficult to solve by deterministic algorithms based on subgradient information, the heuristic algorithm was considered. The optimization mechanism and characteristics of the basic great deluge algorithm (GDA) were analyzed and the solving steps were given as well. Then improved GDAs for unconstrained and box constrained problems were proposed respectively, where the parameter up was omitted. For the unconstrained case, the random walk algorithm with respect to neighborhood search was proposed. For the box constrained case, the method of choosing a feasible initial point and a chaos optimization algorithm with respect to neighborhood search were proposed. The improved GDAs were tested by taking several typical nonsmooth optimization problems as examples and were compared with other algorithms. The test results show that the improved GDAs are efficient and superior to other algorithms mentioned in the paper. So it can be used as a practical method for solving nonsmooth optimization problems.

Keywords: great deluge algorithm; nonsmooth optimization; random walk; chaos

收稿日期: 2014-12-01

基金项目: 国家自然科学基金资助项目(11171221); 高等学校博士学科点专项科研资助项目(20123120110004); 上海市一流学科建设资助项目(XTKX2012)

第一作者: 张清叶(1978-), 女, 博士研究生. 研究方向: 非光滑优化、投资组合优化. E-mail: zhangqingye123@163.com

通信作者: 高 岩(1962-), 男, 教授. 研究方向: 非光滑优化、投资组合优化、混杂系统控制. E-mail: gaoyan@usst.edu.cn

非光滑优化广泛应用于经济、力学、工程和最优化控制等众多领域,然而由于不具有连续可微性质,基于梯度信息的经典优化方法不再适用.目前,次梯度法和束方法是求解非光滑优化问题最常用的两种方法^[1-2],但由于凸函数的次梯度与局部 Lipschitz 函数的广义梯度计算困难较大,从而严重影响了非光滑优化算法的应用.鉴于非光滑优化问题的普遍性及计算的困难性,本文拟引进启发式算法对其进行处理.

大洪水算法(great deluge algorithm, GDA)是一种受自然界启发的优化算法,它通过模拟洪水的上涨过程来进行全局寻优,该算法最早由 Dueck 于 1993 年提出^[3]. Dueck 在利用 threshold accepting (TA)算法进行实验时,发现了两个行之有效的启发式算法,其中之一便是大洪水算法.它与模拟退火算法^[4](simulated annealing, SA)有着相似的结构,但在迭代过程中对劣解的接受规则不同,且比模拟退火算法依赖于更少的参数.由于其遍历过程与圣经中诺亚方舟的大洪水有相似之处,故称之为大洪水算法.其算法思想可描述为:设想有一块崎岖不平的空地,在这块空地上一直不间断地下雨从而产生洪水,空地上有一个人,他可以向任意方向移动,然而随着水平面不断上涨,为了生存,此人必须不断地移至更高处,一段时间之后,此人最终将到达空地的最高点,即最大化问题的最优解.对于最小化问题,有两种处理方法:其一是将目标函数取负值,从而将最小化问题转化成最大化问题;其二是修改大洪水算法,不妨称为干旱算法,寻找高处的人为寻找水的鱼类代替.

Dueck 首先将大洪水算法用于旅行商问题(traveling salesman problem, TSP)的求解,随后国内外一些学者对大洪水算法进行了研究^[5-10],如将其应用于二次分配、电力调度、复杂系统可靠性等问题的求解中.然而,截至目前尚未有学者将其用于求解非光滑优化问题.

本文接下来给出基本大洪水算法的求解步骤,然后针对无约束及盒子约束非光滑优化问题分别设计改进的大洪水算法,同时给出相应的伪代码,最后通过算例对所给算法进行测试.

1 基本大洪水算法的求解步骤

以最大化目标函数 $f(\mathbf{x})$ 为例,其中 $f(\mathbf{x})$ 是 $\mathbb{R}^n$ 上的实值函数,即求解 $\max_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x})$.

算法 1 基本大洪水算法(BGDA)

Step 1 初始化

给定初始可行解 $\mathbf{x}_0$, 计算 $f(\mathbf{x}_0)$;

设置初始水平面高度 $water_level$, 令 $water_level := f(\mathbf{x}_0)$;

设置水平面上升速度 $up > 0$;

设置迭代次数上界 $count$, 置迭代因子 $i := 1$;

for $i = 1: count$, 执行 Step 2.

Step 2 优化

在 $\mathbf{x}_0$ 附近进行邻域搜索, 得新解 $\mathbf{x}$, 计算 $f(\mathbf{x})$.

执行 if 判断语句:

if $f(\mathbf{x}) > water_level$, 则

置 $\mathbf{x}_0 := \mathbf{x}$, $water_level := water_level + up$.

End if

迭代因子 i 增加 1, 即令 $i := i + 1$.

End for

Step 3 输出 $\mathbf{x}_0, f(\mathbf{x}_0)$.

大洪水算法由于每轮仅允许从当前水平面以上的点中进行搜索, 故算法效率较高. 此外, Dueck 指出, 即使空地很快被洪水分割成几块不连续的部分, 算法依然能很快找到最优解^[3]. 与其他优化算法相比, 大洪水算法最明显的优势是仅依赖于唯一的参数 up . 一般来说, up 的值越大, 算法运行越快但解的质量较差; 反之, up 的值越小, 算法产生的解质量越高但运行的时间也越长. Dueck 通过实验发现, 最好的 up 应满足 $up \leq |f(\mathbf{x}) - water_level| \times 1\%$ ^[3], 其中 $f(\mathbf{x})$ 为当前目标函数值, $water_level$ 为当前水平面高度.

2 针对无约束优化问题的算法

考虑问题

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad (1)$$

其中 $f(\mathbf{x})$ 不一定连续可微. 由于无约束, 可在 $\mathbb{R}^n$ 中任选一点作为初始解 $\mathbf{x}_0$, 然后利用随机行走法^[11]进行邻域搜索. 不妨设迭代进行到第 k 步, 当前解为 $\mathbf{x}_k$, 设置迭代格式, 令 $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$, 其中 λ_k 为邻域半径, $\mathbf{d}_k$ 为规划后的搜索方向. 为使 $\mathbf{d}_k$ 能遍历所有方向, 可随机生成多个 $[-1, 1]$ 之间的 n 维向量, 其中 n 为决策变量 $\mathbf{x}$ 的维数. 注意到在基本大洪水算法的求解步骤 Step 2 中, 对 $water_level$ 的更新准则依赖于参数 up , 改进算法将参数 up 省去, 更新准则采用 $water_level = f(\mathbf{x})$, 这样一方面

可以减少参数,另一方面通过大量测试发现运行效率可得到较大提高.下面给出求解问题(1)的具体步骤,按伪码形式叙述如下.

算法 2 无约束优化问题的改进大洪水算法, (UCGDA)

Begin

给定 $\mathbf{x}_0$, 计算 $f(\mathbf{x}_0)$, 设置初始水平面高度 $water_level$, 令 $water_level := f(\mathbf{x}_0)$;

设置迭代次数上界 $count$, 设置邻域搜索时生成搜索方向个数 m , 置迭代因子 $i := 1$;

进入 for 循环, for $i = 1 : count$.

生成 m 个在 $[-1, 1]$ 之间服从均匀分布的 n 维随机变量 $\mathbf{u}_k$, 将 $\mathbf{u}_k$ 单位化, 令 $\mathbf{d}_k = \mathbf{u}_k / \|\mathbf{u}_k\|$.

设置邻域半径 λ , 令 $\mathbf{x}_k = \mathbf{x}_0 + \lambda \mathbf{d}_k$, 计算 $f_k = f(\mathbf{x}_k)$. 令 $a := \min\{f_k \mid 1 \leq k \leq m\} = f_I, \mathbf{x} := \mathbf{x}_I$.

执行 if 判断语句: if $a < water_level$, 则

置 $\mathbf{x}_0 := \mathbf{x}, water_level := a$.

End if

迭代次数增加 1, 即令 $i := i + 1$.

End for

输出最优解 $\mathbf{x}_0$ 和最优值 $water_level$.

End

算例测试

为验证算法的有效性,下面选几个测试函数进行求解,并将求解结果与用模拟退火算法、Matlab 优化工具箱命令 `fminsearch` 所得结果进行对比.表 1 列出了当前能找到的最优解 $\mathbf{x}^{**}$ 和对应的最优值 $f^{**} = f(\mathbf{x}^{**})$,表 2 列出了各种算法求得的最优解 $\mathbf{x}^*$ 与对应的最优值 $f^* = f(\mathbf{x}^*)$.上述 3 种算法均在 Win7 系统 Matlab2012a 环境下实现.参数设置如下:算法 2(UCGDA)迭代上界 $count$ 取 1 000,邻域半径 λ 取 0.1,邻域搜索时搜索方向个数 m 取 100;模拟退火算法与 `fminsearch` 的参数设置取工具箱默认值,初始点为相同初始点.

例 1 wolfe 函数

$$f(x_1, x_2) = \begin{cases} 5(9x_1^2 + 16x_2^2)^{\frac{1}{2}}, & x_1 \geq |x_2| \\ 9x_1 + 16|x_2|, & 0 < x_1 < |x_2| \\ 9x_1 + 16|x_2| - x_1^9, & x_1 \leq 0 \end{cases}$$

初始点设为 $\mathbf{x}_0 = (3, 2)^T$.

例 2 Rosen-Suzuki 函数

$$f(\mathbf{x}) = \max_{1 \leq i \leq 4} f_i(\mathbf{x})$$

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 21x_3 + 7x_4$$

$$f_2(\mathbf{x}) = f_1(\mathbf{x}) + 10(x_1^2 + x_2^2 +$$

$$x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8)$$

$$f_3(\mathbf{x}) = f_1(\mathbf{x}) + 10(x_1^2 + 2x_2^2 +$$

$$x_3^2 + 2x_4^2 - x_1 - x_4 - 10)$$

$$f_4(\mathbf{x}) = f_1(\mathbf{x}) + 10(2x_1^2 + x_2^2 +$$

$$x_3^2 + 2x_1 - x_2 - x_4 - 5)$$

初始点设为 $\mathbf{x}_0 = (0, 0, 0, 0)^T$.

例 3 Spiral 函数

$$f(\mathbf{x}) = \max\{f_1(\mathbf{x}), f_2(\mathbf{x})\}$$

$$f_1(\mathbf{x}) = (x_1 - \sqrt{x_1^2 + x_2^2} \cos \sqrt{x_1^2 + x_2^2})^2 + 0.005(x_1^2 + x_2^2)$$

$$f_2(\mathbf{x}) = (x_2 - \sqrt{x_1^2 + x_2^2} \sin \sqrt{x_1^2 + x_2^2})^2 + 0.005(x_1^2 + x_2^2)$$

初始点 $\mathbf{x}_0 = (1.411\ 831, -4.794\ 62)^T$.

表 1 当前最优解

Tab.1 Current optimal solution

算例	$\mathbf{x}^{**}$	f^{**}
1	$(-1, 0)^T$	-8
2	$(0, 1, 2, -1)^T$	-44
3	$(0, 0)^T$	0

表 2 求解结果

Tab.2 Calculated results

算例	算法	$\mathbf{x}^*$	f^*	$ f^* - f^{**} $
1	UCGDA	$(-0.999\ 2, 0.000\ 1)^T$	-7.997 8	0.002 2
	SA	$(-0.991\ 0, 0.000\ 0)^T$	-7.997 1	0.002 9
	fminsearch	$(-1.000\ 0, 0.000\ 0)^T$	-7.999 9	0.000 1
2	UCGDA	$(0.015\ 9, 1.015\ 8, 1.984\ 2, -1.014\ 6)^T$	-43.992 8	0.007 2
	SA	$(0.024, 1.046, 1.968, -1.028)^T$	-43.981 1	0.018 9
	fminsearch	$(-0.032, -0.017, 1.294, -1.802)^T$	-32.939 4	11.060 6
3	UCGDA	$(0.016\ 7, 0.004)^T$	1.49×10^{-5}	1.49×10^{-5}
	SA	$(1.412, -4.795)^T$	0.124 9	0.124 9
	fminsearch	$(-1.30, 1.788)^T$	0.024 849	0.024 85

通过对表 2 的求解结果进行分析可以发现, UCGDA 的求解精度均优于 SA. 此外, 由于 `fminsearch` 为局部搜索算法, 其求解结果严重依赖于初始点的选取, 当初始点选取不太合适时, 将产生较大误差, 如例 2. 而在实际问题中, 由于最优解是待求的, 初始点的选取大都是盲目的, 故为得到全局最优解应使用 UCGDA. 综上, 为得到质量较好的全局最优解, 算法 2 不失为求解非光滑无约束优化问题的一种实用方法.

为进一步加快算法收敛速度, 在对 $water_level$ 进行更新时, 亦可令

$water_level = \min\{water_level + up, f(x)\}$,
其中, $up < 0$ 为水平面下降速度.

3 针对盒子约束优化问题的算法

考虑问题

$$\begin{cases} \min f(x) \\ \text{s.t. } a \leq x \leq b \end{cases} \quad (2)$$

式中, x, a, b 均为 n 维向量, $f(x)$ 不一定光滑.

为叙述方便, 记 $G = \{x | a \leq x \leq b\}$ 为一盒子, 则此时最优解须在盒子内. 处理方法常见的有两种, 一种是利用罚函数的思想, 引入指示函数 $\delta_G(x)$. 其中, 当 $x \in G$ 时, $\delta_G(x) = 0$; 当 $x \notin G$ 时, $\delta_G(x) = +\infty$. 于是问题(2)就转化成无约束问题

$$\min_{x \in \mathbb{R}^n} f(x) + \delta_G(x)$$

另一种方法则是不进行转化, 直接对约束优化问题进行求解. 本文拟设计专门针对约束优化问题的改进大洪水算法.

不妨在盒子 G 内任取一点 x_0 作为初始解, 其函数值 $f(x_0)$ 作为初始水平面, 设置 up 值及迭代次数上界 $count$. 在邻域搜索时, 所求解 x 不仅需在 x_0 附近, 还需在 G 内. 为得到函数 $f(x)$ 在盒子 G 内的全局最优解, 希望邻域搜索时能遍历 G 内的所有点且最好不重复, 而混沌恰好具有此性质, 故在求解约束优化问题时, 可考虑采用 Logistic 映射产生混沌序列.

混沌是一种普遍的非线性现象, 其行为复杂且类似随机, 但有精致的内在规律性. 混沌优化算法是一种全局优化算法^[9,12], 其理论基础是混沌的遍历性. 混沌优化是通过混沌变量来实现的, 混沌变量的产生有多种方法, 如应用广泛的 Logistic 映射, 其方程为 $t_{i+1} = \mu t_i(1 - t_i)$, $i = 1, 2, \dots$, 其中 μ 为控制参数, $0 \leq t_i \leq 1$, 当 $\mu = 4$ 时处于完全混沌.

设函数优化变量 $x(a \leq x \leq b)$ 为 n 维向量, $T = \{t_1, t_2, \dots\}$ 为 Logistic 映射函数产生的混沌序列, 其中 $t_i (i = 1, 2, \dots)$ 亦为 n 维向量且满足

$$\begin{aligned} t_{i+1} &= 4t_i \cdot (1 - t_i), \\ i &= 1, 2, \dots, 0 < t_i < 1 \end{aligned} \quad (3)$$

式中: $\cdot$ 表示两个向量的对应元素相乘, 其结果为一 n 维向量; 0 表示元素全为 0 的 n 维向量; 1 表示元素全为 1 的 n 维向量. x 与 T 取值的映射与逆映射公式为

$$t_i = (x_i - a) / (b - a), i = 1, 2, \dots \quad (4)$$

$$x_i = a + t_i \cdot (b - a), i = 1, 2, \dots \quad (5)$$

即 $x_i \Leftrightarrow t_i \rightarrow t_{i+1} \Rightarrow x_{i+1} \rightarrow t_{i+2} \Rightarrow x_{i+2} \rightarrow t_{i+3} \rightarrow \dots$, 其中, $\cdot$ 表示两个向量的对应元素相除, 其结果为一 n 维向量.

至此可以给出求解问题(2)的具体算法步骤. 为进一步加快算法收敛速度, 不妨在 G 内随机产生 m 个点 $x_{01}, x_{02}, \dots, x_{0m}$, 计算其对应的目标函数值 $f_{0i} = f(x_{0i}), 1 \leq i \leq m$, 记 $f_0 = \min\{f_{0i} | 1 \leq i \leq m\} = f_{0I}$, 取 $x_0 = x_{0I}$. 下面按伪码形式叙述求解问题(2)的改进大洪水算法.

算法 3 (盒子约束优化问题的改进大洪水算法, BCGDA)

Begin

在 G 内随机产生 m 个点 $x_{01}, x_{02}, \dots, x_{0m}$, 计算其对应函数值 $f_{0i} = f(x_{0i}), 1 \leq i \leq m$, 记 $f_0 = \min\{f_{0i} | 1 \leq i \leq m\} = f_{0I}$, 取 $x_0 = x_{0I}$. 将 x_0 作为初始可行点, 按式(4)将 x_0 映射成混沌变量 t_0 ;

设置初始水平面高度为 $water_level$, 令 $water_level := f_0$;

设置迭代次数上界 $count$, 置迭代因子 $i := 1$;

进入 for 循环, for $i = 1 : count$.

按式(3)由 t_i 生成 t_{i+1} , 按式(5)将 t_{i+1} 映成 x_{i+1} , 计算 $f(x_{i+1})$. 执行 if 判断语句:

if $f(x_{i+1}) < water_level$, 则

置 $x_0 := x_{i+1}, water_level := f(x_{i+1})$.

End if

迭代次数增加 1, 即置 $i := i + 1$.

End for

输出最优解 x_0 和最优值 $water_level$.

End

算例测试

同算法 2 一样, 为验证其有效性, 下面选几个测试函数进行求解, 并将求解结果与用模拟退火算法、Matlab 优化工具箱命令 `fmincon` 所得结果进行对比. 表 3 列出了目前能找到的最优解 x^{**} 和最优值 $f^{**} = f(x^{**})$, 表 4 列出了各种算法的求解结果 x^* 和 $f^* = f(x^*)$. 参数设置如下: 算法 3 (BCGDA) 迭代上界 $count$ 取 1 000, 为加快收敛速度, 在算法初始化阶段, 在可行域 G 内随机生成 100 个点, 即取

表 3 当前最优解

Tab.3 Current optimal solution

算例	x^{**}	f^{**}
4	$(-2, -2)^T$	-1.5
5	$(-488.632, 511.999)^T$	511.731 9

表4 求解结果
Tab.4 Calculated results

算例	算法	x^*	f^*	$ f^* - f^{**} $
4	BCGDA	$(-2, -2)^T$	-1.500 0	0
	SA	$(-2, -2)^T$	-1.499 9	0.000 1
	fmincon	$(1, -2)^T$	-1.499 99	0.000 1
	BCGDA	$(-488.587\ 0, -511.999\ 7)^T$	511.725 5	0.006 4
5	SA	$(-488.131\ 7, 511.149\ 7)^T$	511.072 2	0.659 7
	fmincon	$(-512.000\ 0, 486.908\ 1)^T$	488.843 1	22.888 8

$m = 100$; 模拟退火算法与 fmincon 的参数设置取工具箱默认值, 初始点为相同初始点.

例4

$$\min f(x, y) = 2|y - x| - \left| \frac{1}{2} - y \right| + 1$$

$$\text{s.t. } -2 \leq x \leq 2, -2 \leq y \leq 2$$

曲面图如图1所示.

例5

$$\max f(x, y) = -x \sin \sqrt{|y+1-x|} \cdot$$

$$\cos \sqrt{|y+1+x|} - (y+1) \cdot$$

$$\cos \sqrt{|y+1-x|} \sin \sqrt{|y+1+x|}$$

$$\text{s.t. } -512 \leq x \leq 512, -512 \leq y \leq 512$$

曲面图如图2所示.

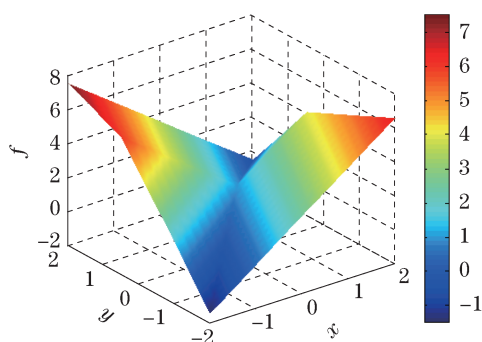


图1 例4曲面图

Fig.1 Surface figure of example 4

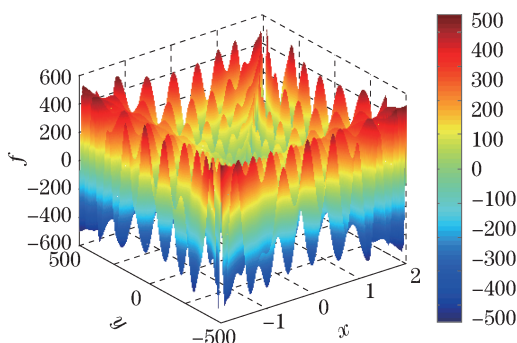


图2 例5曲面图

Fig.2 Surface figure of example 5

这里需要指出的是, 模拟退火算法与 fmincon 的运行结果与初值关系非常密切, 给定不同的初值将得到不同的结果. 为了得到较好的结果, 首先在 G 内任取 100 个点, 从中筛选出最优点作为初始可行点, 例 4 中取 $x_0 = (0, 0)^T$, 例 5 取 $x_0 = (-502.582\ 9, 482.484\ 5)^T$. 算法 3 对初值不敏感, 给定不同初值, 均能得到最优解.

4 结束语

本文针对无约束及盒子约束非光滑优化问题提出了改进的大洪水算法, 去掉了大洪水算法所依赖的唯一参数 up , 对无约束情形在进行邻域搜索时提出了随机行走算法; 对约束优化问题为加快算法收敛速度, 给出了选择初始可行点的方法, 在进行邻域搜索时引入了混沌优化算法. 通过对大量算例进行测试, 发现本文提出的改进大洪水算法是求解非光滑优化问题的一种切实可行的方法. 此外, 虽然本文所提算法是针对非光滑函数给出的, 但在实际应用中, 任何非线性规划问题均可用之求解. 为进一步验证算法的可行性及优越性, 对文献[9]所给算例进行了测试, 结果发现本文所给算法能较快收敛到最优解. 如何将大洪水算法推广至带复杂非线性约束非光滑问题的求解上, 以便进一步扩大其应用范围, 是一个值得考虑的问题.

参考文献:

- [1] 高岩. 非光滑优化[M]. 北京: 科学出版社, 2008.
- [2] 王伟伟, 高岩. 凸可行问题的一种次梯度投影算法[J]. 上海理工大学学报, 2009, 31(5): 422-426.
- [3] Dueck G. New optimization heuristics: the great deluge algorithm and the record-to-record travel[J]. Journal of Computational Physics, 1993, 104(1): 86-92.
- [4] Eglese R W. Simulated annealing: a tool for operational research [J]. European Journal of Operational Research, 1990, 46(3): 271-281.
- [5] 魏欣, 马良, 张惠珍. 二次分配问题的大洪水算法求解[J]. 运筹与管理, 2011, 20(1): 12-15.
- [6] Lenin K, Reddy B R, Kalavathi M S. An improved great deluge algorithm (IGDA) for solving optimal reactive power dispatch problem[J]. International Journal of Electronics and Electrical Engineering, 2014, 2(4): 321-326.

(下转第 61 页)

- [3] Tamg C. Perspectives in supply chain risk management[J]. International Journal of Production Economics, 2006, 103(2): 451 - 488.
- [4] 秦桂英, 刘宇熹. 基于云计算的供应链运营模式研究[J]. 上海理工大学学报, 2013, 35(3): 245 - 250.
- [5] Kacen J, Hess J D, Chiang W Y K. Bricks or clicks? consumer attitudes toward traditional stores and online stores[J]. Global Economics and Management Review, 2013, 18(1): 12 - 21.
- [6] Tang O, Musa S N. Identifying risk issues and research advancements in supply chain risk management[J]. International Journal of Production Economics, 2011, 133(1): 25 - 34.
- [7] Kermack W O, McKendrick A G. A contribution to the mathematical theory of epidemics[J]. Proceedings of the Royal Society of London, Series A, 1927, 115 (772): 700 - 721.
- [8] 张芳, 司光亚, 罗批. 谣言传播模型研究综述[J]. 复杂系统与复杂性科学, 2009, 6(4): 1 - 11.
- [9] 赵来军, 吴盼. 考虑传播率和移出率变化的谣言传播规律研究[J]. 上海理工大学学报, 2014, 36(4): 345 - 350.
- [10] Huang J Y, Jin X G. Preventing rumor spreading on small-world networks[J]. Journal of Systems Science and Complexity, 2011, 24(3): 449 - 456.
- [11] Zheng M H, Lü L Y, Zhao M. Spreading in online social networks: the role of social reinforcement[J]. Physical Review E, 2013, 88(1): 2252 - 2279.
- [12] Bass F M. A new product growth for model consumer durables[J]. Management Science. 1969, 15 (5): 215 - 227.
- [13] 陶晓波, 宋卓昭, 张欣瑞, 等. 网络负面口碑对消费者态度影响的实证研究——兼论企业的应对策略[J]. 管理评论, 2013, 25(3): 101 - 110.

(编辑: 丁红艺)

(上接第 47 页)

- [7] Ravi V. Optimization of complex system reliability by a modified great deluge algorithm[J]. Asia-Pacific Journal of Operational Research, 2004, 21(4): 487 - 497.
- [8] Al-milli N. Hybrid genetic algorithm with great deluge to solve constrained optimization problems[J]. Journal of Theoretical and Applied Information Technology, 2014, 59(2): 385 - 389.
- [9] 盛虹平, 马良. 混沌大洪水算法求解函数优化问题[J]. 计算机应用研究, 2011, 28(5): 1626 - 1627.
- [10] Kifah S, Abdullah S. An adaptive non-linear great deluge algorithm for the patient-admission problem[J]. Information Sciences, 2015, 295: 573 - 585.
- [11] 吴鹏. Matlab 高效编程技巧与应用: 25 个案例分析[M]. 北京: 北京航空航天大学出版社, 2010.
- [12] 王秀梅, 秦体恒. 基于混沌优化算法的最小二乘圆参数估计[J]. 武汉理工大学学报, 2008, 30(8): 101 - 104.

(编辑: 丁红艺)