

记忆原理的元胞自动机优化算法及其收敛性证明

陆秋琴 牛倩倩 黄光球

(西安建筑科技大学管理学院 西安 710055)

摘要 为了解决大规模优化问题,根据记忆原理与元胞自动机的特点构造了求解优化问题的全局收敛算法。在该算法中,将优化问题的理论搜索空间划分为离散搜索空间,该空间定义为元胞空间,其中的每个元胞对应着一个候选解。将记忆原理的记忆、遗忘规律用于控制每个元胞的状态转移;元胞的状态由其空间位置、位置修正量以及记忆残留值构成,该值分为瞬时记忆、短时记忆和长时记忆3种状态类型,并依据元胞接受刺激的强度被加强或衰减;记忆残留值低于某个阈值的元胞时被遗忘,不再被处理。在元胞演化过程中,元胞从一个状态转移到另一个状态实现了元胞空间对理论搜索空间的搜索。应用可归约随机矩阵的稳定性条件证明了本算法具有全局收敛性。测试结果表明本算法是高效的。

关键词 优化,元胞自动机,记忆原理,全局收敛性

中图分类号 TP18 **文献标识码** A

Cellular Automata Algorithm for Solving Optimization Problems Based on Memory Principles and its Global Convergence Proof

LU Qiu-qin NIU Qian-qian HUANG Guang-qiu

(School of Management, Xi'an University of Architecture & Technology, Xi'an 710055, China)

Abstract To solve large-scale optimization problems(OP), the algorithm with global convergence was constructed for solving OP based on the characteristics of memory principles(MP) and cellular automata(CA). In the algorithm, the theoretical search space of OP is divided into the discrete space, and the discrete space is defined as cellular space where each cell is an alternative solution of OP; the memorizing and forgetting rules of MP are used to control transition of states of each cell; a cellular state consists of position, increment of position and residual memory which is divided into three kinds of memory state such as instantaneous, short and long-term memory, each of which is strengthened or weakened by accepted stimulus strength. A cell is forgotten and then discarded when its residual memory is lower than a threshold. During evolution process, a cell's transferring from one state to another realizes the search of cellular space on the theoretical search space. The stability condition of a reducible stochastic matrix was applied to prove the global convergence of the algorithm. The case study shows that the algorithm is efficient.

Keywords Optimization, Cellular automata, Memory principles, Global convergence

1 引言

考虑优化问题:

$$\begin{aligned} & \min f(\mathbf{X}) \\ & \text{s. t. } \begin{cases} g_i(\mathbf{X}) \geq 0, i=1, 2, \dots, W \\ \mathbf{X} \in G \subset \mathbb{R}^n \end{cases} \end{aligned} \quad (1)$$

式中, G 为搜索空间; $f(\mathbf{X})$ 为目标函数, $g_i(\mathbf{X}) \geq 0$ 为第 i 个约束条件, $i=1, 2, \dots, W$, W 为约束条件个数; $\mathbf{X}=(x_1, x_2, \dots, x_n)$ 是一个 n 维决策向量, 通常每个自变量 x_i 满足边界约束条件: $l_i \leq x_i \leq u_i, i=1, 2, \dots, n; \mathbb{R}^n$ 为 n 维实空间。

目标函数 $f(\mathbf{X})$ 和约束条件 $g_i(\mathbf{X}) (i=1, 2, \dots, M)$ 往往非线性、不连续、不可微, 甚至无明确的数学表达式, 因此, 利用

传统的数值算法^[1]很难得到理想的优化结果。为了解决优化问题(1), 人们已发展出了进化优化算法, 这类算法对目标函数和约束条件一般不需要特殊的限制条件, 具有较广泛的适用性。已有的进化优化算法有: 遗传算法^[2]、蚁群算法^[3]、鱼群算法^[4]、粒子群算法^[5]、模拟退火算法^[6]和生物地理学算法^[7]等。

在元胞自动机优化方法研究方面, 文献[8]提出了一种新的优化算法——元胞蚂蚁算法, 该算法将元胞自动机的邻居和规则引入传统的蚂蚁算法, 实验结果证明该算法可行且有效, 有良好的全局优化能力; 文献[9]研究了元胞遗传算法在动态函数优化问题上的性能表现, 设计出了一种动态环境下基于种群多样性引入灾难的元胞遗传算法; 文献[10]将人工

到稿日期: 2012-06-11 返修日期: 2012-09-05 本文受陕西省科学技术研究发展计划项目(2011K06-08), 陕西省教育厅科技计划项目(12JK0789)资助。

陆秋琴(1966—), 女, 博士, 教授, 主要研究方向为先进计算, E-mail: luqiuqin88@yahoo.cn(通信作者); 牛倩倩(1987—), 女, 硕士生, 主要研究方向为先进计算; 黄光球(1964—), 男, 博士, 教授, 主要研究方向为计算智能。

鱼群算法与元胞自动机的邻域模型和状态迁移规则相结合,设计了元胞鱼群算法;文献[11]利用元胞自动机的离散空间与并行计算特性,通过对元胞的抽象和局部规则的设计,并借助于元胞状态的动态演化,解决了 AOE 网络中多源点多汇点关键路径的求解;文献[12]运用元胞自动机理论,针对无向图剖分优化问题进行了分析和建模,提出了一种元胞自动机模型以及基于该模型的无向图剖分优化算法;文献[13]提出了基于元胞自动机的小生境微粒群算法,指出了针对不同的函数虽然应该选用不同的邻域,但存在一些好的邻域结构,它们往往更适合广泛范围函数的优化;文献[14]将遗传算法和元胞自动机相结合应用到水网的优化计算中;文献[15,16]提出采用遗传算法来提高元胞自动机的运算速度,并进行实际的应用;文献[17]将元胞自动机和模拟退火算法相结合来解决优化问题;文献[18]用进化元胞自动机来解决森林规划问题;文献[19]用蚁群算法、马尔科夫链和元胞自动机相结合的方法来建立土地使用变化空间模型;文献[20]用元胞自动机和粒子群优化规则来建立都市动态增长模型。

将记忆的思想用于优化问题的求解,目前已有一定的研究,如文献[21,22]等。但这些算法中的记忆是通过将已发生过的历史信息直接引入表达式中,例如,文献[21,22]采用如下表达式就可描述记忆因素:

$$x(t+1) = a_0 x(t) + a_1 x(t-1) + a_2 x(t-2) + \dots$$

式中,若当前时刻为 t ,当前取值为 $x(t)$,未来值为 $x(t+1)$,则 $a_1 x(t-1) + a_2 x(t-2) + \dots$ 就表示对历史 $t-1, t-2, \dots$ 的记忆。很显然,用这种方式描述的记忆与人类记忆原理的真实机制相差甚远。

为了快速解决大规模优化问题,本文将人类记忆原理引入到优化问题的求解中,提出了基于记忆原理的元胞自动机优化算法。该算法具有如下特色:

(1)采用元胞自动机模型,将优化问题的候选解空间定义成元胞空间,一个候选解就是一个元胞,通过增加元胞空间中的元胞数量达到增加候选解多样性的目的;通过元胞状态的演化实现对优化问题最优解的搜索,且元胞状态的演化可并行执行。

(2)每个元胞的状态信息除了包括与解(包括候选解及其变化量)相关的属性信息之外,还包括与记忆相关的属性信息。前一部分称为解状态,后一部分称为记忆状态。

(3)解状态一直向好的方向转移的元胞,其记忆状态值会不断被加强,且不易被遗忘,若幸运,可以快速求得全局或局部最优解;解状态一直向坏的方向转移的元胞,其记忆状态值会不断被衰减,且很快被遗忘,此举可以有效防止无效迭代和溢出。

(4)处于局部最优状态的元胞,因解状态很难发生转变,其记忆状态值不断衰减直至被遗忘,此举可以有效跳出局部最优状态。

(5)CA 演化过程中,元胞的解状态向好或向坏的方向转移随机交替出现,可使搜索过程丰富多彩。

(6)当优化问题(1)存在大量局部最优解时,CA 演化到一定阶段后,因处于大量局部最优解状态的元胞和状态不断变坏的元胞被遗忘,可大幅减少元胞处理数量,从而减少搜索时间。因此,随着演化的不断进行,本算法的收敛速度将越来越快。故本算法对存在大量局部最优解的优化问题效果更

佳,同时在不增加搜索时间的条件下可增大元胞数量。

2 基于记忆原理的元胞自动机优化算法

2.1 元胞自动机优化模型定义

定义 1^[23] 一个元胞自动机可用如下四元组描述:

$$CA = (L_d, S, N, F)$$

式中,CA 代表一个元胞自动机系统; L_d 代表一个 d 维元胞空间, d 理论上可以是一个任意正整数; S 代表各个元胞状态类型的离散有限集合; N 代表邻域元胞的组合(包括中心元胞); F 代表一个将 S_i 映射到 S 上的一个局部转换函数,又称为局部规则或状态转换函数,也就是 t 时刻根据某个元胞的邻居的状态组合来更新 $t+1$ 时刻该元胞的状态。演化规则描述如下:

$$S_i^{t+1} = f(S_1^t, \dots, S_j^t, \dots, S_{|N|}^t)$$

式中, S_i^{t+1} 为第 i 个元胞在 $t+1$ 时刻的状态, $\{S_1^t, S_2^t, \dots, S_{|N|}^t\}$ 为第 i 个元胞所有邻域的集合,包括其本身;中心元胞的邻居个数为 $|N|$ 。

基于定义 1,可以构造出 CA 优化模型的基本框架,如定义 2 所述。

定义 2 设求解优化问题(1)时,在 t 时刻(即第 t 次演化)的 $\prod_{i=1}^d M_i$ 个候选解为:

$$x^s(t) = (x_1^s(t), x_2^s(t), \dots, x_n^s(t)), s=1, 2, \dots, \prod_{i=1}^d M_i$$

则元胞自动机模型如下:

(1)每个候选解即为一个元胞,元胞总个数为 $\prod_{i=1}^d M_i$ 。

(2)元胞空间 L_d 采用 d 维元胞空间表示,即

$$L_d = \{(i_1, i_2, \dots, i_d) | j=1, 2, \dots, d; i_j=1, 2, \dots, M_j\}$$

其形成方法为:将 $\prod_{i=1}^d M_i$ 个元胞排列成 $M_1 \times M_2 \times \dots \times M_d$ 维矩阵,此时所有元胞可以用 $c(i_1, i_2, \dots, i_d, k)$ 来表示,即

$$c(i_1, i_2, \dots, i_d, k) = x_k \left(\sum_{k=3}^d (i_k - 1) \prod_{s=1}^{k-1} M_s + (i_1 - 1) M_1 + i_2 \right), k=1, 2, \dots, n; j=1, 2, \dots, d; i_j=1, 2, \dots, M_j$$

这样,坐标 (i_1, i_2, \dots, i_d) 表示了元胞在 d 维空间的位置,但元胞的空间位置与优化问题(1)的解的空间位置没有关系,也就是说,位置上相邻的元胞并不表明其对应的候选解在空间上也是相邻的。

(3) t 时刻元胞状态空间 $S(t)$ 由所有元胞的当前空间位置 $\{x^s(t) | s=1, 2, \dots, \prod_{i=1}^d M_i\}$ 、空间位置修正量 $\{\Delta x^s(t) | s=1, 2, \dots, \prod_{i=1}^d M_i\}$, $\Delta x^s(t) = (\Delta x_1^s(t), \Delta x_2^s(t), \dots, \Delta x_n^s(t))$ 和元胞状态累积记忆值 $\{m^s(t) | s=1, 2, \dots, \prod_{i=1}^d M_i\}$ 构成,即

$$S(t) = \{(x^s(t), \Delta x^s(t), m^s(t)) | s=1, 2, \dots, \prod_{i=1}^d M_i\} \quad (2)$$

显然,任意时刻 t ,都有 $|S(t)| = |L_d| = \prod_{i=1}^d M_i$ 。

(4)每个元胞的邻居 N 可以根据实际情形定义,常用的邻居可使用半径为 r 的 Moore 型邻域来表示。

(5)元胞状态演化规则如下:

$$\Delta x^s(t+1) = F(\Delta x^s(t), x^s(t), \mathbf{gBest}(t), \mathbf{lBest}(t)) \quad (3)$$

$$x^s(t+1) = x^s(t) + \Delta x^s(t+1) \quad (4)$$

式中, $\mathbf{gBest}(t)$ 为截止到时刻 t 为止,元胞状态空间内已达到的优化问题(1)的全局最优解。

$$gBest(t) = \mathbf{x}^s(t) = (x_1^s(t), x_2^s(t), \dots, x_n^s(t)), s_g \in \{1, 2, \dots, \prod_{i=1}^d M_i\}$$

s_g 是使下式成立的元胞空间位置:

$$F(s) = \begin{cases} F(\mathbf{x}^s(t)), & \forall i \in \{1, 2, \dots, W\}, g_i(\mathbf{x}^s(t)) \geq 0, \mathbf{x}^s(t) \in G \\ F_{\max}, & \exists i \in \{1, 2, \dots, W\}, g_i(\mathbf{x}^s(t)) < 0, \mathbf{x}^s(t) \in G \end{cases} \quad (5)$$

$$F(s_g) = \min_{s_g} \{F(s) \mid \forall s \in L_d\} \quad (6)$$

式中, F_{\max} 为非常大的实数; $\mathbf{IBest}(t)$ 为时刻 t , 元胞邻域内使优化问题(1)达到最优的局部最优解, $\mathbf{IBest}(t) = \mathbf{x}^i(t) = (x_1^i(t), x_2^i(t), \dots, x_n^i(t)), s_i \in N_r(s), N_r(s)$ 为元胞 s 的半径为 r 的邻域, 简称 r 邻域. s_i 是使下式成立的位于元胞 s 的邻域中的元胞空间位置:

$$F(s_i) = \min_{s_i} \{F(v) \mid \forall v \in N_r(s)\} \quad (7)$$

式(3)可以根据问题的性质选取合适的形式, 但小心选择式(3)的表达形式可使优化问题(1)的求解过程收敛更快. 下面给出一个式(3)常用的形式.

$$\Delta \mathbf{x}^s(t+1) = w(t) \Delta \mathbf{x}^s(t) + c_1 r_1(t) (gBest(t) - \mathbf{x}^s(t)) + c_2 (t) r_2(t) (\mathbf{IBest}(t) - \mathbf{x}^s(t)) \quad (8)$$

式中, $w(t)$ 为惯性系数; $c_1(t)$ 和 $c_2(t)$ 为学习系数.

2.2 基于记忆原理的 CA 优化算法的工作原理

2.2.1 记忆的基本模型^[24, 25]

现代认知心理学认为, 记忆是对客观世界信息识记、保持、再认和回忆的过程. 记忆包括瞬时记忆、短时记忆、长时记忆 3 个阶段, 其相互关系如图 1 所示.

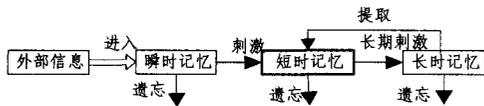


图 1 记忆的基本模型

瞬时记忆是通过感觉器官进入瞬时记忆存贮空间, 是外界刺激的原始呈现, 完成对信息的过滤. 短时记忆是信息传递给长时记忆的容量有限的缓冲器, 组成了当前头脑中的信息, 对决策起着重要的作用. 长时记忆是短时记忆长期刺激的结果, 能够反映客观世界相对稳定的信息, 它使主体能回忆过往信息, 并据此进行推理和解决未知问题. 遗忘是指对已识记过的材料的不能再认.

本文引入记忆值描述优化系统对事件发生的记忆的强弱, 记忆值越大, 则对该事件的记忆越强, 反之越弱; 若记忆值小于指定值, 该事件及其相关信息就从记忆库中被删除, 这时就认为优化系统对该事件实行了遗忘操作.

2.2.2 遗忘数学模型

本算法中涉及的瞬时记忆、短时记忆、长时记忆 3 种记忆都会随着时间的流逝而逐渐减弱. 根据艾宾浩斯遗忘曲线可得元胞 s 的记忆衰减函数^[25]:

$$m_i^s(t) = m_i^s(t_0) e^{-a_c(t-t_0)} \quad (9)$$

式中, $m_i^s(t)$ 表示系统对元胞 s 的状态在时刻 t 的记忆残留值, t_0 为最后一次更新记忆值的时刻, $t > t_0$. 系统演化最初时刻 $t_0 = 0$; a_c 为记忆衰减速度调节系数 $a_c > 0$, $c = I, S, L$, 分别表示瞬时记忆、短时记忆、长时记忆 3 种记忆类型. 由于瞬时记忆衰减迅速, 因此 a_I 的值较大, 短时记忆的 a_S 次之, 长时间记忆的 a_L 最小. 结合人类记忆衰减的实际情况, 该函数必需

满足如下的限制条件:

$$\lim_{t-t_0 \rightarrow 0} m_i^s(t) e^{-a_c(t-t_0)} = m_i^s(t_0)$$

$$\lim_{t-t_0 \rightarrow \infty} m_i^s(t) e^{-a_c(t-t_0)} = 0$$

2.2.3 记忆的更新数学模型

考虑到优化问题(1)的特殊性, 本算法中涉及的瞬时记忆、短时记忆、长时记忆 3 种记忆类型的记忆残留值均保存在 $m^s(t)$ 中, 并将其保存在元胞的状态信息数据结构中. 若 $m^s(t) < M_S$, $M_S = (1 - a_S) M_0$, 则认为元胞 s 的状态处在瞬时记忆(IM)中; 若 $M_S \leq m^s(t) \leq M_L$, $M_L = (1 - a_L) M_0$, 则认为元胞 s 的状态处在短时记忆(SM)中; 若 $m^s(t) > M_L$, 则认为元胞 s 的状态处在长时记忆(LM)中, 即

$$MS^s(t) = \begin{cases} \text{IM}, & m^s(t) < M_S \\ \text{SM}, & M_S \leq m^s(t) \leq M_L \\ \text{LM}, & m^s(t) > M_L \end{cases} \quad (10)$$

式中, $MS^s(t)$ 为元胞 s 的状态所处的记忆类别. 经过式(10)处理后, 无需再对记忆进行分类. 因此, 式(9)可改写为

$$m^s(t) = m^s(t_0) e^{-a(t-t_0)}, a = \begin{cases} a_I, & MS^s(t) = \text{IM} \\ a_S, & MS^s(t) = \text{SM} \\ a_L, & MS^s(t) = \text{LM} \end{cases} \quad (11)$$

时刻 t 优化系统会根据元胞 s 所受的刺激的大小计算其记忆值 $\Delta m^s(t)$. $F(s) - F(s_i)$ 越大, 认为元胞 s 所受的刺激越大, $\Delta m^s(t)$ 的值越大, 反之, $\Delta m^s(t)$ 的值就越小. $\Delta m^s(t)$ 的计算公式如下:

$$\Delta m^s(t) = h |F(s) - F(s_i)| \quad (12)$$

式中, h 为刺激调节系数, 可根据优化问题的具体情况取值. 对元胞 s 来说, 某事件出现, 则对元胞 s 的状态的记忆会增强, 即

$$m^s(t) = m^s(t) + \Delta m^s(t) \quad (13)$$

由于每个元胞上都保存有有关该元胞状态的记忆残留值, 因此对以往相关的事件的回忆是瞬间完成的. 记忆值会随时间衰减, 也即出现越早的事件其记忆值衰减得越多, 出现越晚的事件其记忆值衰减得越少. 包含回忆的记忆衰减函数表达如下:

$$m^s(t) = m^s(t_0) e^{-a(t-t_0)} + \lambda m^s(t_0) f(t-t_0) \quad (14)$$

式中, $f(t-t_0)$ 称为回忆衰减函数; λ 为衰减调整系数, $\lambda > 0$, 其取值参考式(11). 在不同条件下, $f()$ 可有多种表示形式, 用于控制回忆衰减速度. 本模型中的衰减速度与时间间隔 $(t-t_0)$ 成反比, 与记忆残留值成正比, 因回忆产生的记忆增长应相对原始记忆衰减得较快, 且满足如下条件: $\lim_{t-t_0 \rightarrow \infty} f(t-t_0) = 0$.

函数 $f(x) = e^{-bx}$ (b 常数, $b > a_L$, 其取值参考式(11)) 具有很好的平滑性, 且满足上述条件, 采用该函数作为长时记忆的回忆衰减函数. 因此, 式(14)可改写为:

$$m^s(t) = m^s(t_0) (e^{-a(t-t_0)} + \lambda e^{-b(t-t_0)}) \quad (15)$$

综合考虑记忆的增强、遗忘和回忆衰减等因素后, 可得对元胞 s 状态的记忆更新数学模型为:

$$m^s(t) = m^s(t_0) (e^{-a(t-t_0)} + \lambda e^{-b(t-t_0)}) + \Delta m^s(t) \quad (16)$$

2.3 基于记忆原理的 CA 优化算法工作原理

基于记忆原理的 CA 优化算法是将人类记忆原理应用到 CA 优化模型的元胞状态转移某些特殊特事件的处理中, 其工作原理如下所述.

算法 1 记忆原理的工作原理

- (1) 时刻 t 元胞 s 进入优化系统的瞬时记忆后,按式(5)计算 $F(s)$,记录其状态信息 $(x^s(t), \Delta x^s(t))$;确定该元胞的 r 邻域内元胞 $N_r(s)$,按式(7)找出 $N_r(s)$ 内具有最佳 $F_g(s_i)$ 值的元胞 s_i ,记录其状态信息 $(x^{s_i}(t), \Delta x^{s_i}(t))$ 。
- (2) 比较元胞 s 的 $F(s)$ 是否优于其 r 邻域内最佳元胞 s_i 的 $F(s_i)$,进行如下处理:
 - (2.1) 若 $F(s) > F(s_i)$,则系统认为事件“元胞 s 当前状态可以向更好的状态转移”已发生,此信息被认定为特殊刺激,对元胞 s_i 的记忆即被加强,该加强值按式(12)计算,元胞 s 的累积记忆值 $m^s(t)$ 按式(16)计算。若 $m^s(t) > M_g$,则信息“ r 邻域内某最佳元胞 s_i 的状态 $(x^{s_i}(t), \Delta x^{s_i}(t))$ ”连同其累积记忆值 $m^s(t)$ 进入到元胞 s 的短时记忆中;若元胞 s 的累积记忆值 $m^s(t) > M_l$,则元胞 s 进入到其长时记忆中。进入长时记忆中的元胞,遗忘速度很慢。
 - (2.2) 若 $F(s) \leq F(s_i)$,则优化系统认为“元胞 s 当前状态不能向好的状态转移”,此信息被认定为普通刺激,记忆值不会被加强,元胞 s 的累积记忆值出现衰减,其累积记忆残值 $m^s(t)$ 按式(11)计算。若元胞 s 的累积记忆值 $m^s(t)$ 低于阈值 M_0 ,元胞 s 要被遗忘掉,即元胞 s 处在哪个记忆状态(瞬时记忆、短时记忆或长时记忆),就从哪个记忆状态中删除。被遗忘的元胞不再被处理。
- (3) 不断重复(1)–(2),直到每轮将元胞空间中的所有元胞处理完为止。

基于记忆原理的 CA 优化算法的工作原理的优势在于:

- (1) 步骤(2.1)表明,状态一直向好的方向转移的元胞,其累积记忆值不断加强,且不易被遗忘。若幸运,可以快速求得全局或局部最优解。
- (2) 步骤(2.2)表明,状态一直向坏的方向转移的元胞,其累积记忆值不断衰减,且很快被遗忘,可以有效防止溢出。
- (3) 步骤(2.2)还表明,当处于局部最优状态的元胞,其状态很难发生转变时,不断衰减该元胞状态的累积记忆值直至被遗忘,此举可以有效跳出局部最优状态。
- (4) 步骤(2.1)和步骤(2.2)的随机交替出现,可使搜索过程丰富多样。
- (5) 当优化问题(1)存在大量局部最优状态时,处于这些状态的元胞和状态不断变坏的元胞被遗忘,可大幅减少元胞处理数量,从而减少搜索时间。

因此,本算法对存在大量局部最优状态的优化问题效果更佳,同时在不增加搜索时间的条件下可增大元胞空间数量。

3 算法步骤、时间复杂度与边界元胞处理方法

3.1 算法步骤

算法 2 基于记忆原理的 CA 优化算法

- (1) $t=0$,确定 d, M_1, M_2, \dots, M_d ,从而确定 L_d ;初始化由式(2)表示的元胞状态空间 $\{(x^s(t), \Delta x^s(t), m^s(t)) | s=1, 2, \dots, \prod_{i=1}^d M_i\}$;按式(5)计算元胞空间每个元胞的函数值;按式(6)计算 $gBest(t) = x_g^s(t)$;令 $forget(s) = \text{false}$, $s=1, 2, \dots, \prod_{i=1}^d M_i$, $forget(s)$ 为元胞 s 的遗忘状态。
- (2) 元胞空间内各元胞的演化过程如下:
 - FOR $t=1$ TO $G // G$ 为元胞空间演化的最大代数。
 - FOR $s \in L_d$ AND $forget(s) = \text{false}$ // 对元胞空间中的每个元胞 s 进行处理,但处于遗忘状态的元胞除外。

确定当前元胞 s 的 r 邻域内的所有元胞 $N_r(s)$ 。若当前元胞 s 为边界元胞,则按 3.3 节介绍的边界元胞处理方法选取邻域元胞 $N_r(s)$;

对于所有 $v \in N_r(s)$,按式(5)计算元胞 v 的函数值 $F(v)$,按式(7)确定 r 邻域内最佳元胞 s_i 及其对应的 $IBest(t) = x^{s_i}(t)$;

IF $F(s) > F(s_i)$ THEN

当前元胞 s 的状态以概率 $p_a = 1$ 转移到 r 邻域内最佳元胞 s_i 所对应的状态上,即 $y^s(t+1) = IBest(t)$, $\Delta y^s(t+1) = 0$;按算法 1 的(2.1)对元胞 s 的记忆属性进行处理。

ELSE

当前元胞 s 执行以下操作:当前元胞 s 的状态以概率 p_b 转移保持不变,或以概率 $1 - p_b$ 转移一个新状态。若当前元胞 s 状态不发生转移,则按算法 1 的(2.2)对元胞 s 的记忆属性进行处理。若元胞 s 的累积记忆值 $m^s(t) < M_0$,则 $forget(s) = \text{true}$;若当前元胞 s 状态发生转移,则新状态计算方法如下:

a) 根据 $gBest(t)$ 、 $IBest(t)$ 和当前元胞 s 的状态值 $(x^s(t), \Delta x^s(t))$ 按式(3)或式(8)计算元胞当前元胞 s 状态中的位置修正量并保存到 $\Delta y^s(t+1)$ 中;

b) 按式(4)计算当前元胞 s 状态中的空间位置并保存到 $y^s(t+1)$,若 $y^s(t+1)$ 超越预定的界限,则应根据问题的具体情况将 $y^s(t+1)$ 缩回到预定的界限内;

c) 检查当前元胞 s 的新状态 $(y^s(t+1), \Delta y^s(t+1))$ 所对应的函数值 F_{new} (按式(5)计算)是否小于 $F(s)$ 。若 $F_{new} < F(s)$ (设此概率为 p_c),则当前元胞 s 的状态调整到新状态,同时按算法 1 的(2.1)对元胞 s 的记忆属性进行处理;若 $F_{new} \geq F(s)$,则反复试探 q 次;若反复试探 q 次后还不成功,则当前元胞 s 的状态不发生改变,按算法 1 的(2.2)对元胞 s 的记忆属性进行处理。若元胞 s 的累积记忆值 $m^s(t) < M_0$,则 $forget(s) = \text{true}$ 。

END IF

调整 $gBest(t)$; // 保存当前时刻 t 下粒子空间内存在的全局最好解的相关信息。

END FOR

FOR $s \in L_d$ AND $forget(s) = \text{false}$

$\Delta x^s(t+1) = \Delta y^s(t+1)$, $x^s(t+1) = y^s(t+1)$; // 更新元胞空间

所有元胞的状态。

END FOR

IF 误差满足最低要求 ϵ THEN 转(3)

END FOR

(3) 保存最优解,演化结束。

3.2 时间复杂度

因每个元胞的邻域空间大小是一样的,不妨令 $|N_r| = |N_r(s)|$,故算法 2 的时间复杂度 $T = G |L_d| (|N_r| + |N_r|n + 3n) + n |L_d| = |L_d| (G(|N_r|(1+n) + 3n) + n)$ 。对于 d 维 Moore 型 r 邻域,则 $|N_r| = (1+2r)^d$,于是 $T = |L_d| (G((1+2r)^d(1+n) + 3n) + n)$ 。显然,算法 2 的时间复杂度与演化次数 G 、元胞空间大小 L_d 、元胞邻域大小 N_r 以及变量个数 n 相关。

在算法 2 中,每个元胞可与周围 $(2r+1)^d - 1$ 个邻域元胞交流信息。邻域半径 r 越大,每个元胞与周围邻域元胞交流信息的个数越多,从而可提高元胞空间中信息的传播速度、减少陷入局部解的可能性,并可提高算法的收敛速度。

3.3 边界元胞处理

元胞自动机模型的边界元胞处理方法均可用于算法 2。

常用的边界处理规则有以下几种^[23]：

(1) 周期型边界。该边界处理规则指元胞空间边界相对连接的处理方法，对二维空间来讲，元胞空间左右边界相接，上下边界相接，使边界循环使用，使元胞空间达到无限延展的目的。周期型边界处理方法模拟出的空间与无限空间最为接近，是目前比较常用的边界处理方法。

(2) 反射型边界。该边界处理规则是指以边界为轴，对元胞空间做镜面反射，使边界内的元胞反射到边界外，使边界处的元胞在边界外存在邻接元胞，这种处理规则也称为对称型边界规则。

(3) 定值型边界。该边界处理规则指给所有边界元胞在边界外设置固定的元胞状态类型。

(4) 随机型边界。随机型边界处理规则指在每次对边界处元胞进行计算时，给边界外设置随机值作为边界元胞的邻域条件。

4 算法 2 的全局收敛性证明

由定义 2 知，元胞空间是一个离散的空间，但每个元胞 s 的状态 $a = (x^s, \Delta x^s)$ 是在连续的实数或复数空间取值。实际搜索是在离散的元胞空间中进行，元胞总个数为 $|L_d|$ ，每个元胞状态 a 即为优化问题(1)的一个候选解，其目标函数值为 $F(a)$ (按式(5)计算)，则所有元胞状态或候选解所形成的集合为

$$F = \{F(a) | a \in L_d\}$$

进一步令

$$F = \{F_1, F_2, \dots, F_{|L_d|}\}, F_1 < F_2 < \dots < F_{|L_d|}$$

$\forall X \in L_d$ 有 $F_{|L_d|} \geq F(X) \geq F_1$ ，将元胞空间 L_d 划分为非空子集：

$$X_i^s = \{X | X \in L_d \text{ 且 } F(X) = F_i\}, i = 1, 2, \dots, |L_d|$$

$$\sum_{i=1}^{|L_d|} |X_i^s| = |L_d|; \forall i \in \{1, 2, \dots, |L_d|\}, X_i^s \neq \phi; \forall i \neq j,$$

$$X_i^s \cap X_j^s = \phi, \bigcup_{i=1}^{|L_d|} X_i^s = L_d$$

令 $X^{i,j} (i=1, 2, \dots, |L_d|, j=1, 2, \dots, |X_i^s|)$ 表示 X_i^s 中第 j 个元胞的状态。在元胞演化过程中，从一个状态转移到另外的状态可表示为 $X^{i,j} \rightarrow X^{k,l}$ ，则从 $X^{i,j}$ 到 $X^{k,l}$ 的转移概率为 $p_{ij,kl}$ ，从 $X^{i,j}$ 到 X_i^s 中任一元胞状态的转移概率为 $p_{ij,k}$ ，从 X_i^s 中任一元胞状态到 X_i^s 中任一元胞状态的转移概率为 $p_{i,k}$ ，则有：

$$p_{ij,k} = \sum_{l=1}^{|X_i^s|} p_{ij,kl}; \sum_{k=1}^{|L_d|} p_{ij,k} = 1; p_{i,k} \geq p_{ij,k}$$

引理 1 在算法 2 中， $\forall X^{i,j} \in X_i^s, i=1, 2, \dots, |L_d|$ ，满足： $j=1, 2, \dots, |X_i^s|$

$$\forall k > i, p_{i,k} = 0 \quad (17)$$

$$\exists k < i, p_{i,k} > 0 \quad (18)$$

(1) 式(17)的证明。设 $X^{i,j}$ 为第 t 次演化后的元胞状态，记为 X^t ，设 X^t 中状态最好的元胞为 $gBest^t = X^* \sqrt{a^2 + b^2}$ ，即有 $F(gBest^t) = F_i$ 。算法 2 在每次演化过程中对当前元胞最好状态的更新可知：

$$F(X^{t+1}) \geq F(X^t) \Rightarrow \forall k > i, p_{ij,kl} = 0 \Rightarrow$$

$$\forall k > i, p_{ij,k} = \sum_{l=1}^{|X_i^s|} p_{ij,kl} = 0 \Rightarrow \forall k > i, p_{i,k} = 0$$

(2) 式(18)的证明。在算法 2 中，设 $gBest^{t+1}$ 为 X^{t+1} 中最

优元胞状态，此时当前元胞探索邻域范围随机选择状态，有两种情况：

情形 1 若当前元胞的邻域范围内某个元胞 v 的状态对应的状态优于当前元胞 s 对应的状态，则当前元胞 s 的状态将以概率 $p_a = 1$ 更新到元胞 v 所对应的状态，显然 $p_a > 0$ ，命题得证。

情形 2 若当前元胞的邻域范围内某个元胞 v 的状态对应的状态次于当前元胞 s 对应的状态，此时当前元胞 s 面临如下两种选择：

a) 当前元胞 s 的状态以概率 p_b 转移保持不变。

b) 或以概率 $1 - p_b$ 转移一个新状态，新状态的计算方法如下：重新随机调整当前元胞的状态，以期更好的解出现，设此概率为 p_r ，这种试探共进行 q 次。当前元胞 s 的状态随机调整后所对应的状态优于调整前所对应的状态的概率为 $P_c = (1 - p_b)(1 - p_r)^{q-1} p_r \geq 0$ ；如果 $P_c > 0$ ，命题得证；如果 $P_c = 0$ ，说明此时元胞演化已经到达低精度解状态或已求得一组真实的精确解状态。

由算法 2 知当前元胞 s 选择上述两种情形的总概率为 1，综合上述两种情况可得 $\exists k < i, p_{i,k} > 0$ ，证毕。

定理 2^[26] 设 P' 是一 n 阶可归约随机矩阵，也就是通过相同的行变换和列变换后可以得到 $P' = \begin{bmatrix} C & \dots & 0 \\ R & \dots & T \end{bmatrix}$ ，其中 C 是 m 阶本原随机矩阵并且 $R \neq 0, T \neq 0$ ，则有：

$$P'^{\infty} = \lim_{k \rightarrow \infty} P'^k = \lim_{k \rightarrow \infty} \begin{bmatrix} C^k & \dots & 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & \dots & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & \dots & 0 \\ R^{\infty} & \dots & 0 \end{bmatrix}$$

上述矩阵是一个稳定的随机矩阵且 $P'^{\infty} = 1' P'^{\infty}$ ， $P'^{\infty} = P^0 P'^{\infty}$ 唯一确定并且与初始分布无关， P'^{∞} 满足如下条件：

$$P'^{\infty} = [p_{ij}]_{n \times n}, \begin{cases} p_{ij} > 0, 1 \leq i \leq n, 1 \leq j \leq m \\ p_{ij} = 0, 1 \leq i \leq n, m < j \leq n \end{cases}$$

该定理的证明可参见文献[26]。

定理 3 算法 2 具有全局收敛性。

证明：对于每个 $X_i^s, i=1, 2, 3, \dots, |L_d|$ 可看为是有限 Markov 链上的一个状态，根据引理 1 中式(17)的结论可得，该 Markov 链的转移矩阵为：

$$P' = \begin{bmatrix} p_{1,1} & 0 & \dots & 0 \\ p_{2,1} & p_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{|L_d|,1} & p_{|L_d|,2} & \dots & p_{|L_d|,|L_d|} \end{bmatrix} = \begin{bmatrix} C & 0 \\ R & T \end{bmatrix}$$

根据引理 1 中式(18)结论得：

$$p_{2,1} > 0, R = (p_{2,1}, p_{3,1}, p_{4,1}, \dots, p_{|L_d|,1})^T$$

$$T = \begin{bmatrix} p_{2,2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ p_{|L_d|,2} & \dots & p_{|L_d|,|L_d|} \end{bmatrix} \neq 0, C = (p_{1,1}) = (1) \neq 0$$

由以上可知转移矩阵 P' 是 $|L_d|$ 阶可归约随机矩阵，满足上述定理 2 中的条件，所以下式成立：

$$P'^{\infty} = \lim_{k \rightarrow \infty} \begin{bmatrix} C^k & \dots & 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & \dots & T^k \end{bmatrix} = \begin{bmatrix} C^{\infty} & \dots & 0 \\ R^{\infty} & \dots & 0 \end{bmatrix}$$

$$C^{\infty} = (1), R^{\infty} = (1, 1, 1, \dots, 1)^T$$

因此有:

$$P^\infty = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \text{是稳定随机矩阵,得}$$

$$\lim_{t \rightarrow \infty} p\{F(X^t) \rightarrow X^*\} = 1$$

因此,算法 2 具有全局收敛性,证毕。

5 例子研究

应用本文提出的算法求解优化问题(19)和(20)。这两个优化问题的全局最优分别为 $(0, 0, \dots, 0)$ 和 $(100, 100, \dots, 100)$;但两个优化问题均存在大量局部最优解,图 2 为 $n=2$ 时局部最优解的分布情况。当维数更高时,局部最优解的分布将更复杂。

$$Y_1 = \min \left[\sum_{i=1}^n (x_i^2 - 10 * \cos(2\pi x_i) + 10) \right] \quad (19)$$

$$Y_2 = \min \left(1 + \frac{\sum_{i=1}^n (x_i - 100)^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) \right) \quad (20)$$

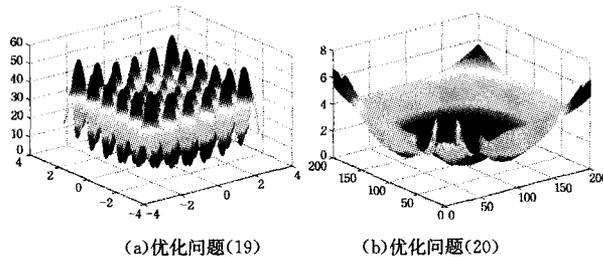


图 2 优化问题(19)和(20)的局部最优解分布情况

计算时,各参数取值方法及其依据如下:

(1) 元胞空间维数 $d=2$, 元胞邻域半径 $r=1$ 。尽管 d 和 r 取较大值可扩大搜索空间,但由 3.2 节知,算法时间复杂度 T 与 $(1+2r)^d$ 成正比,因此, d 和 r 不能取得太大, r 取最小值 1 即可,元胞空间维数 d 取 2 即可。

(2) 刺激调节系数 h 的取值应不使记忆值增长太快为宜;瞬时记忆、短时记忆和长时记忆衰减速度调节系数 a_I, a_S, a_L 的取值应不使瞬时记忆、短时记忆和长时记忆衰减得太快为宜。这些参数的取值无需太高的精确性,只需要依据具体的优化问题而定,对于本文优化问题取 $h=150, a_I=0.09, a_S=0.05, a_L=0.01$ 即可;参数如 b, λ, M_0 以及递增时间步距 Δt 的取值依据也类似,对于本文优化问题取 $b=0.4, \lambda=2.5, M_0=3000, \Delta t=t-t_0=0.2$ 即可。

(3) 式(8)中的惯性系数为^[5]: $w(t) = w_{\max} - \frac{(w_{\max} - w_{\min})t}{G}$, $w_{\max}=1.2, w_{\min}=0.1$;学习系数 $c_1(t)=2.8, c_2(t)=1.2$;其取值依据可参加文献[5]。

(4) 收敛条件 $\epsilon=1 \times 10^{-8}$,其取值依据是我们希望获得的最优解的精度高低。最大迭代代数 $G=2000$,其取值依据是防止迭代过程不满足收敛条件 ϵ 时出现无限迭代。

取维数 $n=2$ 时的优化问题(19)和(20)的最优解分别为 $(1.9132E-08, 1.8125E-08)$ 和 $(99.9997, 100.0012)$ 。其收敛过程如图 3 所示。图 3 说明本文算法可以求出最优解,收敛速度很快且稳定。

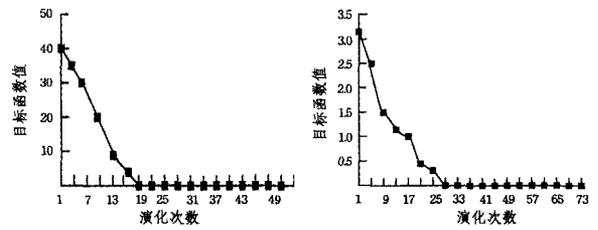


图 3 求解优化问题(19)和(20)时的收敛过程

表 1 本文算法与标准 PSO 算法的性能对比

优化问题	维数 n	本文算法		标准 PSO 算法	
		CA 的规模 d=2	演化代数	演化次数	演化次数
式(19)	10	20×20	68	172	
	20	30×30	128	246	
	30	40×40	113	582	
式(20)	10	20×20	72	337	
	20	30×30	137	634	
	30	40×40	154	817	

结束语 人类记忆原理对于特殊事件的记忆和遗忘具有天然的优势;元胞自动机模型作为一种离散模型系统,对复杂系统演化行为的处理具有天然的优势。本文将这两种理论有机地结合起来用于复杂优化问题的求解,具有创新性。与标准 PSO 相比,本文提出的算法对求解大规模优化问题具有明显的优势。本文给出了算法的时间复杂度表达式,并给出了该算法的全局收敛性证明。该证明过程为本算法性能的进一步提升提供了理论依据。本文今后的工作以及需要进一步改进的地方如下:(1)对不同类型的优化问题,确定更加合理的元胞状态演化规则;(2)将记忆原理用于函数优化问题求解的基本原理标准化,并给出相关标准化算法;(3)在记忆原理中纳入联想记忆功能,从而实现在类似刺激出现时采取回避操作,以提高迭代速度。

参考文献

- [1] 王梓坤. 常用数学公式大全[M]. 重庆:重庆大学出版社,1991:1028-1040
- [2] 黄光球,王国政,周静. 用遗传算法求解物流运输中多级中转定位优化问题[J]. 微电子学与计算机,2006,23(3):47-50
- [3] 黄光球,何星. 基于蚁群算法的随机 Petri 网最优路径序列寻找[J]. 系统仿真学报,2008,20(17):4555-4560
- [4] 李晓磊,邵之江,钱积新. 一种基于动物自治体的寻优模式:鱼群算法[J]. 系统工程理论与实践,2002,22(11):32-38
- [5] 崔志华,曾建潮. 微粒群优化算法[M]. 北京:科学出版社,2011:33-35
- [6] Leung S C H, Zhang De-fu, Zhou Chang-le, et al. A Hybrid Simulated Annealing Metaheuristic Algorithm for the Two-Dimensional Knapsack Packing Problem[J]. Computed Operation Research,2012,39(1):64-73
- [7] Simon D. Biogeography-Based Optimization[J]. IEEE Transactions on Evolutionary Computation,2008,12(6):702-713
- [8] 朱刚,马良,高岩. 元胞蚁群算法的收敛性分析[J]. 系统仿真学报,2007,19(7):142-1459
- [9] 万成. 动态环境下的元胞遗传算法研究[D]. 南昌:南昌航空大

[10] 柳毅,沈勤.带时间窗可回程取货车车辆路径问题的元胞鱼群算法[J].系统管理学报,2011,20(6):739-743

[11] 钱鑫,吴晓军,张甜甜,等.求解关键路径的元胞自动机算法[J].陕西师范大学学报:自然科学版,2009,37(6):19-22

[12] 孙凌宇,冷明,彭宣戈.一种基于元胞自动机的无向图剖分优化算法[J].计算机工程与应用,2008,44(24):46-48

[13] 夏小翔.基于元胞自动机的微粒群算法研究[D].太原科技大学,2007

[14] Sidiropoulos E, Tolikas P. Genetic algorithms and cellular automata in aquifer management[J]. Applied Mathematical Modelling, 2008, 32(6): 617-640

[15] D' Ambrosio D, Spataro W, Iovine G. Parallel genetic algorithms for optimising cellular automata models of natural complex phenomena: An application to debris flows [J]. Computers & Geosciences, 2006(32): 861-875

[16] Karafyllidis I. Acceleration of cellular automata algorithms using genetic algorithms [J]. Advances in Engineering Software, 1999 (30): 419-437

[17] Fadaei A H, Setayeshi S, Kia S. An optimization method based on combination of cellular automata and simulated annealing for VVER-1000 NPP loading pattern[J]. Nuclear Engineering and Design, 2009(239): 2800-2808

[18] Mathey A-H, Krcmar E, Tait D, et al. Forest planning using co-evolutionary cellular automata [J]. Forest Ecology and Management, 2007(239): 45-56

[19] Yang Xin, Zheng Xin-qi, Lv Li-na. A spatiotemporal model of land use change based on ant colony optimization Markov chain and cellular automata[J]. Ecological Modelling, 2012(233): 11-19

[20] Feng Yong-jiu, Liu Yan, Tong Xiao-hua, et al. Modeling dynamic urban growth using cellular automata and particle swarm optimization rules[J]. Landscape and Urban Planning, 2011(102): 188-196

[21] 段其昌, 黄大伟, 雷蕾, 等. 带扩展记忆的粒子群优化算法仿真分析[J]. 控制与决策, 2011, 26(7): 1087-1090

[22] 汤京永, 时贞军. 一类全局收敛的记忆梯度算法及其线性收敛性[J]. 数学进展, 2007, 36(1): 67-75

[23] Chopard B, Droz M. Cellular Automata Modeling of Physical Systems[M]. London: Cambridge University Press, 1998

[24] 黄光球, 石昌文, 孙周军. 基于记忆原理的 Web 入侵预警系统[J]. 系统工程与电子技术, 2006, 28(12): 1940-1944

[25] 艾森克 M W, 基恩 M T. 认知心理学(第 5 版)[M]. 上海: 华东师范大学出版社, 2009

[26] Iisufescu M. Finite Markov Processes and Their Applications [M]. Wiley: Chichester, 1980

(上接第 226 页)

的一些性质。然后在模糊近似空间中,结合直觉模糊等价关系,构造了新的粗糙近似算子,并在 γ 算子的基础之上,讨论了该近似算子的一些重要性质。还基于截集给出 λ 上(下)近似,证明了其一些性质。最后在模糊近似空间上定义粗糙度,并讨论了其性质。

参 考 文 献

[1] Atanassov K. Intuitionistic fuzzy sets[J]. Fuzzy Sets and Systems, 1986, 20: 87-96

[2] 张文修,吴伟志,梁吉业,等.粗糙集理论与方法[M].北京:科学出版社,2007

[3] Dubois D, Prade H. Rough fuzzy sets and fuzzy rough set [J]. International Journal of General Systems, 1990, 17(2): 191-208

[4] Lin L, Yuan X H, Xia Z Q. Multicriteria fuzzy decision-making methods based on intuitionistic fuzzy sets[J]. Journal of Computer and System Sciences, 2007, 73(1): 84-88

[5] Jena S P, Ghosh S K. Intuitionistic fuzzy rough sets[J]. Notes on Intuitionistic Fuzzy Sets, 2002, 8(1): 1-18

[6] Samanta S K, Mondal T K. Intuitionistic fuzzy rough sets and rough intuitionistic fuzzy sets[J]. Journal of Fuzzy Mathematics, 2001, 9(6): 561-582

[7] Zhou L, Wu W Z. On generalized intuitionistic fuzzy rough approximation operators[J]. Information Sciences, 2008, 178(11): 2448-2465

[8] Szmidi E, Kacprzyk J. Distances between intuitionistic fuzzy sets and their applications in reasoning [J]. Studies in Computational Intelligence, 2005, 2: 101-116

[9] 殷宏燕,雷英杰,雷阳.基于海明距离的直觉模糊粗糙集相似度量方法[J].计算机应用,2008,25(12):15-16

[10] 范成礼,雷英杰,张戈.改进的直觉模糊粗糙集相似度量方法[J].计算机应用,2011,31(5):1344-1346

[11] Zhou Lei, Wu Wei-zhi. Characterization of rough set approximations in Atanassov intuitionistic fuzzy set theory[J]. Computers and Mathematics with Applications, 2011, 62: 282-296

[12] Huang Bing, Li Hua-xiong, Wei Da-kuan. Dominance based rough set model in intuitionistic fuzzy information systems[J]. Knowledge-Based Systems, 2012, 28: 115-123

[13] Wu Wei-zhi, Zhou Lei. On intuitionistic fuzzy topologies based on intuitionistic fuzzy reflexive and transitive relations[J]. Soft Computing A Fusion of Foundations, Methodologies and Applications, 2011, 15(6): 1183-1194

[14] Tripathy B K. Rough Sets on Fuzzy Approximation Spaces and Intuitionistic Fuzzy Approximation Spaces[J]. Studies in Computational Intelligence, 2009, 174: 3-44

[15] Lin Meng-lei. A Characterization for Intuitionistic Fuzzy Sets Based on the Assistant Sets Generated by S-Rough Sets[J]. Advances in Soft Computing, 2009, 54: 627-631

[16] Jiang Yun-cheng, Tang Yong, Wang Ju, et al. Reasoning within intuitionistic fuzzy rough description logics[J]. Information Sciences, 2009, 179(14): 2362-2378

[17] Zhou Lei, Zhang Wen-xiu, Wu Wei-zhi. Roughness Measures of Intuitionistic Fuzzy Sets[J]. Lecture Notes in Computer Science, 2008, 5009: 308-315

[18] Thomas K V, Nair, Latha S. Rough intuitionistic fuzzy sets in a lattice[J]. International Mathematical Forum, 2011, 6(27): 1327-1335

[19] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Sciences, 1982, 11: 341-356

[20] Banerjee M, Sankar K P. Roughness of a fuzzy set[J]. Information Science, 1996, 93: 235-246

[21] Shen Yong-hong, Wang Fa-xing. Rough approximations of vague sets in fuzzy approximation space[J]. International Journal of approximate reasoning, 2011, 52: 281-296

[22] 苗夺谦,李道国.粗糙集理论、算法与应用[M].北京:清华大学出版社,2008