

# 混沌算法在节水灌溉制度优化设计中的应用<sup>\*</sup>

段春青<sup>1</sup>, 邱林<sup>2</sup>, 陈晓楠<sup>1</sup>, 黄强<sup>1</sup>

(1 西安理工大学 水电学院, 陕西 西安 710048;

2 华北水利水电学院 水利系, 河南 郑州 450008)

**[摘要]** 基于目前各种节水灌溉制度优化方法存在的不足, 本研究将混沌优化算法运用于作物灌溉制度优化设计中。由于混沌优化算法的叠代具有不重复性和遍历性, 因此该算法可以避免陷入局部最优点而获得全局最优。实例计算分析表明, 与遗传算法、动态规划逐次逼近法相比, 混沌优化算法具有原理简单、计算简便、结果精度高的优点。

**[关键词]** 灌溉制度优化; 节水灌溉; 混沌优化算法

**[中图分类号]** S274.1

**[文献标识码]** A

**[文章编号]** 1671-9387(2005)09-0133-04

随着我国水资源的不断短缺, 我国农业面临着严重的水危机。如何提高灌溉水资源的利用效益, 即在向农田供给总灌溉水量一定的条件下, 通过灌溉水量的合理分配, 获得最大的效益, 使单位灌溉水量的产值最大, 是节水灌溉农业研究中亟待解决的问题。因此, 近年来研究者对此进行了大量的相关研究, 其中付强等<sup>[1,2]</sup>将基于实码加速遗传算法的投影寻踪分类模型应用于水稻灌溉制度优化设计, 并使用实码加速遗传算法求解了水稻的灌溉制度优化模型; 马建琴等<sup>[3]</sup>、邱林等<sup>[4]</sup>应用模糊数学对作物的灌溉制度进行优化设计, 并对灌区干旱概率分布作了研究; 朱成立等<sup>[5]</sup>通过建立二维动态规划模型, 对冬小麦的灌溉制度进行了详细的研究; 张祖莲等<sup>[6]</sup>利用动态规划对中稻的水分生产函数和优化灌溉制度进行了研究, 均取得了一定的阶段性成果。

传统方法求解最优灌溉制度是根据作物的生长过程, 将全生育期划分为 $N$ 个生长阶段, 以各阶段初的可用水量作为状态变量, 各生长阶段的实际灌水量和实际腾发量作为决策变量, 利用动态规划逐次逼近法(DPSA)进行计算。该方法计算复杂, 且不能保证在任何情况下都能收敛得到全局最优解。遗传算法(GA)虽被广泛应用, 但其局部寻优能力差, 容易出现早熟现象, 且交叉概率和变异概率的选择对最优解有较大影响<sup>[7]</sup>。

混沌优化算法(CA)利用其叠代不重复性和遍历性, 可按“自身规律”不重复地遍历所有状态。因

此, 该算法可以避免陷入局部最优点而获得全局最优。为此, 本研究运用混沌优化算法的优势, 进行了灌溉制度优化设计研究, 旨在为提高优化计算的精度与速度, 从而为作物灌溉制度的制定及灌水量的合理分配提供技术支持。

## 1 混沌优化算法

### 1.1 混沌优化算法的思路

混沌是非线性系统所独有且广泛存在的一种非周期的运动形式, 表现出介于规则和随机之间的一种行为, 其现象几乎覆盖了自然科学和社会科学的每一个分支<sup>[8]</sup>。混沌具有精致的内在结构, 能把系统的运动吸引并束缚在特定的范围内, 按其“自身规律”不重复地遍历所有状态, 因此, 利用混沌变量进行优化搜索, 无疑能跳出局部最优的羁绊, 最终取得满意的结果。

非线性规划处理约束条件下的极值问题, 一般可表示为:

$$\begin{cases} \min f(X) \\ s.t. \quad g_i(X) \geq 0, i = 1, 2, \dots, m \\ \quad h_j(X) = 0, j = 1, 2, \dots, l \end{cases} \quad (1)$$

式中,  $X \in E^n$ ;  $f(X)$  为目标函数;  $g_i(X)$ ,  $h_j(X)$  为其约束条件。这些函数中至少有一个为非线性函数。约束条件有时用集合形式表示, 令  $S = \{X | g_i(X) \geq 0, i = 1, 2, \dots, m; h_j(X) = 0, j = 1, 2, \dots, l\}$ , 称  $S$  为可行

<sup>\*</sup> [收稿日期] 2004-12-27

[基金项目] 河南省高校创新人才培养工程项目(HNCX2003-17)

[作者简介] 段春青(1979-), 女, 河南 阳人, 在读博士, 主要从事水资源系统管理研究。

集或可行域,  $S$  中的点称为可行点。

Logistic 模型是混沌研究中的最典型模型之一, 其方程为:

$$x_{k+1} = \lambda \times x_k \times (1 - x_k), x_k \in [0, 1] \quad (2)$$

式中,  $\lambda$  为控制参数,  $\lambda = 1.0 \sim 4.0$ 。Logistic 的映射是  $[0, 1]$  上的不可逆映射, 当  $\lambda = 4.0$  时, 系统处于混沌状态, 任意取初始点, 可以得到  $[0, 1]$  上的遍历点列。用 Logistic 方程来生成混沌序列, 此序列也叫混沌变量, 将其转化成优化问题解空间中混沌遍历的变量, 则可通过搜索寻优寻找问题的最优解。

本研究解决式(1)的思想是对目标函数不做变动, 考虑 Logistic 方程生成的混沌序列, 将其放大到包含可行域  $S$  的一个区域, 从中搜索属于  $S$  的点, 然后通过比较, 迭代最终求出问题的最优解。

## 1.2 混沌优化算法的求解步骤

设式(1)中  $X$  的维数为  $M$ ,  $X = [x_1, x_2, \dots, x_M]$ ,  $x_i \in [d_i, e_i]$ 。混沌优化方法求解该问题的基本步骤为:

第一步: 算法初始化。置  $k = 1$ , 随机生成  $M$  个初值  $x_{i,k}$ ,  $i = 1, 2, \dots, M$ , 令

$$x'_{i,k} = d_i + (e_i - d_i) \cdot x_{i,k} \quad (3)$$

当其构成的向量  $X'_k \in S$  时, 令  $X^* = X'_k$ ,  $f^* = f(X^*)$ , 否则重复步骤 1 直到获得的  $X'_k$  满足约束条件。

第二步: 置最大迭代步数  $N$ 。将  $X_1$  带入式(2)迭代, 生成混沌向量序列  $\{X_k\}$ ,  $k = 2, 3, \dots, N$ , 将混沌向量序列  $\{X_k\}$  按照式(3)进行载波, 对得到的  $X'_k$  进行可行性检验。如果通过检验, 如  $f(X'_k) < f(X^*)$ , 则  $X^* = X'_k$ ,  $f(X^*) = f(X'_k)$ ; 否则, 令  $k = k + 1$ , 对  $X'_{k+1}$  进行检验, 重复步骤 2 直到  $f(X^*)$  的值只有微小变化或者满足最大迭代次数。

第三步: 设定循环次数。按照  $Z_{k+1} = Z_k + \alpha \cdot Z'_{k+1}$  进行二次载波, 令  $Z_k = X^*$ ,  $Z'_{k+1}$  为由 Logistic 映射生成的混沌序列。  $\alpha$  为很小实数, 如 0.000 1, 可以根据具体问题进行设定。

对  $Z_{k+1}$  进行检验, 如果通过检验, 如  $f(Z_{k+1}) < f(Z_k)$ , 则  $Z_k = Z_{k+1}$ ,  $f(Z_k) = f(Z_{k+1})$ ; 否则, 令  $k = k + 1$ , 重复步骤 3, 直到  $f(Z_k)$  的值满足精度要求或满足最大迭代次数。最终的  $Z_k$  即为所求解, 对应的  $f(Z_k)$  为最优值。

## 2 作物最优灌溉制度的数学模型

混沌优化算法求解的是静态数学模型, 对于旱

作物, 目标函数采用 Jensen 公式, 即:

$$F = \max \left( \frac{Y_a}{Y_m} \right) = \max \prod_{i=1}^N \left( \frac{ET_a}{ET_m} \right)^{\lambda_i} \quad (4)$$

式中,  $Y_a$  为作物实际产量 ( $\text{kg}/\text{hm}^2$ );  $Y_m$  为作物最大产量 ( $\text{kg}/\text{hm}^2$ );  $(ET_a)_i$  为作物第  $i$  个生育阶段的实际腾发量 ( $\text{mm}$ );  $(ET_m)_i$  为作物第  $i$  个生育阶段的潜在腾发量 ( $\text{mm}$ );  $\lambda_i$  为作物第  $i$  个生育阶段的敏感系数;  $N$  为作物生育阶段数。

该函数的约束条件为:

$$\sum_{i=1}^N m_i = Q, m_i \geq 0 \quad (5)$$

式中,  $m_i$  为作物第  $i$  个阶段单位面积上的灌水量 ( $\text{mm}$ );  $Q$  为全生长期单位面积上可供分配的水量 ( $\text{mm}$ )。

$$(ET_{\min})_i \leq (ET_a)_i \leq (ET_{\max})_i, \quad i = 1, 2, \dots, N \quad (6)$$

式中,  $(ET_{\min})_i$ ,  $(ET_{\max})_i$  分别为作物第  $i$  个生育阶段的最小和最大腾发量 ( $\text{mm}$ )。

$$0 \leq S_i \leq S_{\max} \quad (7)$$

式中,  $S_i$  为作物第  $i$  个生育阶段初土壤中可供利用的水量 ( $\text{mm}$ );  $S_{\max}$  为土壤中可供作物利用水量的最大值 ( $\text{mm}$ )。

$$S_{i+1} = S_i + m_i + P_i - CK_i - (ET_a)_i - K_i \quad (8)$$

式中,  $P_i$  为作物第  $i$  个生育阶段的有效降雨量 ( $\text{mm}$ );  $CK_i$  为作物第  $i$  个生育阶段的地下水补给量 ( $\text{mm}$ );  $K_i$  为作物第  $i$  个生育阶段的田间渗漏量 ( $\text{mm}$ )。

$$S_1 = 1000YH(\theta_1 - \theta_0) \quad (9)$$

$$S_{\max} = 1000YH(\theta - \theta_0) \quad (10)$$

式中,  $Y$  为土壤干容重 ( $\text{t}/\text{m}^3$ );  $H$  为计划湿润层深度 ( $\text{m}$ );  $\theta_1$  为第 1 个生育阶段初计划湿润层内的土壤平均含水率, 以占干土重的百分数计;  $\theta$ ,  $\theta_0$  为土壤含水率的上下限, 以占干土重的百分数计。

对于水稻, 需将  $S_i$  换算为第  $i$  生育阶段初田面蓄水深度  $h_i$ 。稻田处于非饱和状态时,  $h_i$  也是土壤含水率的函数, 可由根系层平均土壤含水率与饱和含水率之差折算成相应水深, 定义饱和状态水深为 0, 则此时水深  $h_i$  为负数, 即:

$$h_i = 1000YH(\theta - \theta_0) \quad (11)$$

式中,  $\theta$  为第  $i$  阶段的土壤含水率;  $\theta_0$  为土壤饱和含水率, 以占干土重的百分数计。

$$(H_{\min})_i \leq h_i \leq (H_{\max})_i \tag{12}$$

式中,  $(H_{\min})_i$ ,  $(H_{\max})_i$  分别为第  $i$  个生育阶段田面水深的上下限(mm)。 $(H_{\max})_i$  的值可根据实际灌溉经验确定,  $(H_{\min})_i$  则根据根系层土壤平均含水率的下限值及根系层的深度确定, 一般为负值。

$$h_{i+1} = h_i + P_i + m_i - (ET_a)_i - C_i - K_i \tag{13}$$

式中,  $C_i$  为第  $i$  个生育阶段的排水量;  $K_i$  由下式计算<sup>[2]</sup>:

$$K_i = \overline{K_i} \cdot d_i + \sum_{j=1}^{D_i-d_i} [1\,000 \cdot K_0 / (1 + K_0 \cdot a \cdot t_j / H)] \tag{14}$$

式中,  $\overline{K_i}$  为正常供水情况下稻田的日平均渗漏量(mm);  $D_i$  为第  $i$  个生育阶段的总天数;  $K_0$  为饱和水利传导度(m/d);  $a$  为经验常数, 一般为 50~ 250;  $t_j$  为土壤含水率从饱和状态到第  $j$  天水平所经历的天数;  $H$  为水稻主要根层深度(m)。 $d_i$  为第  $i$  个生育阶段稻田有水层的天数, 可按下式计算:

$$d_i = (h_i + m_i + P_i) / (\overline{ET_i} + \overline{K_i}) \tag{15}$$

式中,  $\overline{ET_i}$  为正常供水情况下稻田日平均腾发量(mm)。其中

$$h_1 = h_0 \tag{16}$$

式中,  $h_0$  为插秧后稻田保持的水深, 一般取 20 mm。

3 实 例

对节水灌溉条件下井灌水稻的优化灌溉制度进行实例计算, 已知作物各生育阶段的敏感系数,  $\lambda$  分别为 0.147 3, 0.325 1, 0.537 0, 0.123 4,  $k_0 = 0.15$  m/d,  $Q = 200$  mm,  $H = 0.4$  m, 其他有关资料见表 1<sup>[2]</sup>。由于水稻黄熟期不需灌水, 故可以不参加计算。用户逐次输入总供水量  $Q$  和  $h_1 = 20$  mm, 则有

目标函数为:

$$F = \max \left( \frac{Y_a}{Y_m} \right) = \max \prod_{i=1}^N \left( \frac{ET_a}{ET_m} \right)^{\lambda_i} = \left( \frac{ET_{a(1)}}{ET_{m(1)}} \right)^{0.147\,3} \cdot \left( \frac{ET_{a(2)}}{ET_{m(2)}} \right)^{0.325\,1} \cdot \left( \frac{ET_{a(3)}}{ET_{m(3)}} \right)^{0.537\,0} \cdot \left( \frac{ET_{a(4)}}{ET_{m(4)}} \right)^{0.123\,4}$$

式中, (1) 为返青- 分蘖期; (2) 为拔节- 孕穗期; (3) 为抽穗- 开花期; (4) 为乳熟期。

其约束条件为:

$$\begin{aligned} 0 \leq m_i \leq Q; i = 1, 2, 3, 4; \sum_{i=1}^4 m_i &= Q \\ 0 \leq ET_{a(1)} \leq 283.5, 0 \leq ET_{a(2)} \leq 81.0, \\ 0 \leq ET_{a(3)} \leq 44.78, 0 \leq ET_{a(4)} \leq 117.5; \\ -50 \leq h_2 \leq 50, -60 \leq h_3 \leq 50, \\ -65 \leq h_4 \leq 40, -65 \leq h_5 \leq 40. \end{aligned}$$

表 1 三江平原富锦试区井灌水稻非充分灌溉条件下优化灌溉制度求解原始数据

Table 1 Original data of optimal irrigation program of crops with well irrigation in non-sufficiency irrigation in test-area of Fujin in Sanjing plain

| 项目<br>Item        | 生育阶段 Procreate phase                 |                               |                             |                            |
|-------------------|--------------------------------------|-------------------------------|-----------------------------|----------------------------|
|                   | 返青- 分蘖<br>Revegetation-<br>Tillering | 拔节- 孕穗<br>Jointing-<br>Tassel | 抽穗- 开花<br>Tassel-<br>Flower | 乳熟期<br>Milky ripe<br>stage |
| $D_i$             | 42                                   | 10                            | 7                           | 20                         |
| $(ET_m)_i$        | 283.5                                | 81.0                          | 44.78                       | 117.5                      |
| $P_i$             | 49                                   | 84.8                          | 37.4                        | 35.7                       |
| $\overline{K_i}$  | 3.33                                 | 3.70                          | 3.10                        | 3.30                       |
| $\overline{ET_i}$ | 6.52                                 | 6.59                          | 5.37                        | 4.36                       |
| $(H_{\max})_i$    | 40                                   | 50                            | 50                          | 40                         |
| $(H_{\min})_i$    | -30                                  | -50                           | -60                         | -65                        |

根据上述详细资料, 假设水稻全生育期可供水量分别为 200 和 250 mm, 使用动态规划逐次逼近法、遗传算法和混沌算法 3 种方法求解, 其中动态规划逐次逼近法将状态变量离散为 11 份。计算结果如表 2 所示。

由表 2 可见, 混沌算法计算精度优于其他两种。

另外, 通过多次试验比较发现, 若要提高 DPSA 的精度, 则需将状态变量离散更多份, 耗时相应更长; GA 的结果不如 CA 稳定, 上述结果是程序运行 10 次的最优结果; 而且, CA 的编程较其他两种方法简单得多。

表 2 3 种算法的计算结果分析

Table 2 Table of data analysis among different algorithm s

mm

| 方法<br>Method | 供水量<br>Water<br>supply | 最优值<br>The<br>optimal<br>value | $m_1$   | $m_2$ | $m_3$  | $m_4$   | $ET_{a(1)}$ | $ET_{a(2)}$ | $ET_{a(3)}$ | $ET_{a(4)}$ |
|--------------|------------------------|--------------------------------|---------|-------|--------|---------|-------------|-------------|-------------|-------------|
| DPSA         | 200                    | 0.883                          | 60      | 20    | 0      | 120     | 151.31      | 81          | 44.78       | 90.438      |
| GA           | 200                    | 0.886                          | 106.916 | 9.806 | 10.343 | 72.935  | 157.334     | 80.237      | 44.78       | 91.019      |
| CA           | 200                    | 0.888                          | 78.644  | 5.639 | 21.673 | 94.044  | 138.567     | 80.999      | 44.764      | 105.761     |
| DPSA         | 250                    | 0.906                          | 100     | 25    | 0      | 125     | 150.8       | 81          | 44.326      | 117.5       |
| GA           | 250                    | 0.913                          | 115.94  | 4.756 | 7.24   | 122.064 | 153.194     | 81          | 44.748      | 117.405     |
| CA           | 250                    | 0.916                          | 130.105 | 7.721 | 6.714  | 105.46  | 173.72      | 80.96       | 44.78       | 103.29      |

## 4 结 论

本研究将混沌优化算法应用于灌溉制度的优化模型求解,并通过实例将该算法与动态优化逐次逼近法以及遗传算法进行了比较。混沌优化算法直接采用混沌变量在允许解空间进行搜索,搜索过程按混沌运动自身规律进行,更容易跳出局部最优解,且搜索效率高,是一种非常有效的寻优方法。与遗传算法及其他方法相比具有以下优点:

- (1) 混沌优化算法原理简明、计算简单、精度高;
- (2) 与动态规划逐次逼近法相比,混沌算法不必

离散状态变量,寻优速度快,且可克服动态规划逐次逼近法需许多次试算初始值的缺点;

(3) 混沌优化算法可克服遗传算法的“早熟现象”,减小交叉概率和变异概率选择对最优解的影响,且精度和速度也优于遗传算法。

综上所述,将混沌优化算法运用到灌溉制度优化设计之中,可以避免其他方法的不足,提高优化结果的精度,减少运算时间。但本模型仅确定了灌水阶段及其水量分配,对具体灌水日期的确定尚未涉及,这还有待今后进一步研究。

## [参考文献]

- [1] 付强,金菊良,梁川.基于实码加速遗传算法的投影寻踪分类模型在水稻灌溉制度优化中的应用[J].水利学报,2002,(10):39-45
- [2] 付强,王立坤,门宝辉,等.推求水稻不充分灌溉下优化灌溉制度的新方法[J].水利学报,2003,(1):123-128
- [3] 马建琴,邱林.作物灌溉制度模糊优化设计[J].华北水利水电学院学报,2000,21(4):6-8
- [4] 邱林,陈晓楠,段春青.农业干旱程度概率分布的研究[J].西北农林科技大学学报(自然科学版),2005,33(3):105-108
- [5] 朱成立,彭世彰,孙景生.冬小麦节水高效优化灌溉制度模型应用研究[J].灌溉排水学报,2003,22(5):77-80
- [6] 张祖莲,李远华.中稻水分生产函数及优化灌溉制度的研究[J].节水灌溉,2001,(6):20-22
- [7] 王凌.智能优化算法及其应用[M].北京:清华大学出版社,2001
- [8] 张伟标.基于混合混沌优化法的BP神经网络算法[J].上海工程技术大学学报,2004,18(1):85-87

## The application of chaos algorithm in optimal design of water-saving irrigation program

DUAN Chun-qing<sup>1</sup>, QIU Lin<sup>2</sup>, CHEN Xiao-nan<sup>1</sup>, HUANG Qiang<sup>1</sup>

(1 Institute of Water Conservancy and Hydroelectric Power, Xi'an University of Technology, Xi'an, Shaanxi 710048, China;

2 North China Institute of Water Conservancy and Hydroelectric Power, Water Conservancy Department, Zhengzhou, Henan 450008, China)

**Abstract:** Based on the deficiency of common optimal methods of water-saving irrigation program, the paper applies chaos algorithm to the optimal design of irrigation program of crops. Based on the congruence non-repetition and ergodicity of chaos, the method will avoid the local optimal solution and find satisfactory globe optimal solution. Finally, the paper compares chaos algorithm with genetic algorithm and dynamic programming successive approximation in detail and the outcome indicates that the method is simple, convenient and accurate.

**Key words:** optimization of irrigation program; water-saving irrigation; chaos optimal algorithm