

# 深基坑支护工程中遗传优化算法的设计\*

## Design of Genetic Optimization Algorithm in Retaining Systems of Deep Foundation Pit

李旻朔 李陶深

Li Minshuo Li Taoshen

(广西大学计算机与信息工程学院 南宁 530004)

(College of Comp. & Info. Engi., Guangxi Univ., Nanning, 530004)

**摘要** 根据深基坑支护工程优化设计的特点,提出一种深基坑支护工程的数学模型和遗传优化算法设计方法。该设计方法充分考虑了遗传算法中的一般处理步骤、个体编码机制、基于可行方案的初始种群的生成、个体的评价及修正算子等因素。运行实验表明,该算法具有较好的有效性和稳定性。

**关键词** 深基坑 遗传算法 适应函数 修正算子

**中图法分类号** TU473.2; TP301.6

**Abstract** A complete and robust genetic algorithm is presented. Through the analysis of the mathematical model of the deep excavations project, the authors put forward the coding mechanism of chromosome, the algorithm of fitness function computation, the key part of the genetic algorithm and the design of the modified operator. The results of our realized project show the effectiveness of the proposed genetic algorithm.

**Key words** deep foundation pit, genetic algorithm, fitness function, modified operator

深基坑支护工程优化设计是一个典型的最优解问题,它以工程造价作为最终优化目标,综合考虑影响基坑工程造价的各个方面的因素。由于存在众多的变量和约束,使深基坑支护工程的求解变得复杂。目前,已提出许多用于求最优解的方法,包括启发式搜索法、神经网络法等,但这些方法适用规模性小,且寻优结果不具备全局特性。遗传算法(简称 GA)作为全局搜索算法已经应用于很多领域。本文给出一种基于遗传算法的深基坑支护工程优化设计方法。

### 1 深基坑支护工程的数学模型

深基坑支护工程的数学模型如下:

$$\min f(X) = \text{基坑支护工程总造价}, X \in U \subset E^n,$$

$$\begin{aligned} \text{s. t. } & h_v(X) = 0, v = 1, 2, \dots, p \\ & g_u(X) \leq 0, u = 1, 2, \dots, m; \end{aligned}$$

其中:  $X$  是由  $x_1, x_2, \dots, x_n$  组成的向量, 是设计过程中要优选的设计变量, 如桩径、桩长、混凝土强度等级等;  $\min$  表示极小化, 即以基坑支护工程总造价最小为优化目标;  $\text{s. t.}$  表示需要满足的约束条件,  $h_v(X)$  为等式约束条件,  $p$  为其数目;  $g_u(X)$  表示不等式约束条件,  $m$  表示其数目。

在  $X$  系统中, 约束条件主要有 3 类:

- (1) 设计变量约束: 指设计变量本身的取值区间, 亦即值域;
- (2) 变量一致性约束: 指各种设计变量之间明显存在的、相互制约的关系;
- (3) 设计准则约束: 指规范、规程中规定必须得到满足的条件。

## 2 深基坑支护工程遗传算法

### 2.1 遗传算法的一般处理步骤

步骤 1 初始化种群. 即在满足约束条件的情况下, 随机、均匀的选择  $N$  个初始解  $X_i$  作为初始种群  $P(0)$ , 其中  $X_i$  称为一条染色体, 变量中的每一个分量叫作基因;

步骤 2  $t := 0$ ;

步骤 3 对种群  $P(0)$  中的每一个个体计算其适应值;

步骤 4 在终止条件不满足情况下, 执行如下操作:

- I. 根据适应值按先前所确定的选择策略计算种群中各个体的选择概率  $P_i$ ;
- II. 计算各种遗传操作进行的概率值;
- III. 根据  $P_i$  从种群  $P(t)$  中选择适应值大的部分个体进行遗传操作, 生成新的种群

$P(t+1)$ ;

IV. 计算种群  $P(t+1)$  中个体的适应值;

V.  $t := t + 1$ ;

步骤 5 返回步骤 4;

步骤 6 输出优化结果;

步骤 7 由抽象的优化结果解码得到要求解的结果。

### 2.2 深基坑支护工程遗传优化算法的设计

#### 2.2.1 编码

编码 (encoding) 是将问题的解空间映射到遗传算法表达空间, 而解码 (decoding) 是遗传算法表达空间映射到问题的解空间。

由于深基坑支护工程解空间的复杂性, 采用常规的二进制编码难以胜任, 作者采用基于值域个数的十进制编码法. 在深基坑支护工程中变量的取值都是一些离散的值, 将每一方案变量的值作为数组  $FA []$  存放, 各个方案变量的取值个数存放在数组  $FAFIELD []$  中, 方案的初始化和演化操作以对应的  $(1-FAFIELD [i])$  为界线. 一个方案对应于长度为  $Length$  的染色体编码串, 每个方案个体的基因又对应一个细部种群. 所有细部变量的值存放在数组  $XB []$ , 每个细部种群对应一个变量取值个数数组  $XBFIELD []$ , 细部的初始化和演化操作以对应的  $(1-XBFIELD [i])$  为界线. 解码时根据方案或细部的基因值, 通过数组  $FA []$  及  $XB []$  还原方案或细部变量的取值。

#### 2.2.2 初始种群的生成

为了保证遗传算法的顺利运行, 必须生成由可行个体组成的初始种群. 相应的代码如下:

```

for (i=1; i<=popsize; i++) {
    for (j=0; j<length; j++)
        faIndParent [i] .chrom [j] =rand ()% (FAFIELD [j]) +1;
        deCode (&faIndParent [i]);
        faIndParent [i] .fitness =faIndParent [i] .restraint ();
        while (faIndParent [i] .fitness ==0)
            { for (j=0; j<length; j++)
                faIndParent [i] .chrom [j] =rand ()% (FAFIELD [j]) +1;
                deCode (&faIndParent [i]);
                faIndParent [i] .fitness =faIndParent [i] .restraint ();
            }
    }
}

```

### 2.2.3 选择操作

本文遗传算法的选择操作采用转盘式选择策略,实施的步骤是:

步骤1 顺序累计种群中各个体的适应值  $f_i$ ,得到适应度的累加值  $S_n = \sum_{i=1}^n f_i$ ;

步骤2 用  $S_n$  去除各个体的适应值  $f_i$ ,得到相对适应度  $P_i$ ,它也就是该个体被选中的概率:

$$P_i = f_i / S_n;$$

步骤3 累计  $P_i$  得累积概率  $g_i = \sum_{j=1}^i P_j$ ;

步骤4 产生  $[0,1]$  内的均匀分布的随机数  $r$ ,满足  $g(i-1) < r \leq g_i$ ,则选择个体  $i$ ;

步骤5 反复执行步骤3、步骤4,直到产生的个体数目等于种群规模。

### 2.2.4 交叉操作

在算法中,运用了整数交叉方法(部分算术杂交法):对染色体的编码  $S = (V_1, V_2, \dots, V_m)$ ,  $V_i \in [a_i, b_i]$ ,  $i = 1, 2, \dots, m$ . 若  $V'_i \in [a_i, b_i]$ ,则  $S' = (V'_1, V'_2, \dots, V'_m)$  都是问题的解。

设  $S_1 = (V_1, V_2, \dots, V_m)$  和  $S_2 = (W_1, W_2, \dots, W_m)$  是父代的2个解向量,  $V_i, W_i \in [a_i, b_i]$ ,  $S_a, S_b$  是杂交后产生的2个后代。在深基坑支护工程中,  $a_i, b_i$  分别表示某一变量的第一个和最后一个取值的标号。通常  $a_i = 1$  或  $0$ ,  $b_i =$  某一变量的取值个数或(个数 - 1)。

交叉操作的步骤如下:

步骤1 确定杂交点,定义函数  $\text{srand}(L1)$ ,取  $1 \sim L$  之间的整数( $L$  为染色体的长度);

步骤2 生成  $n\text{-cropos}$  个  $(0, 1)$  区间的随机数  $a_1, a_2, \dots, a_{n\text{-cropos}}$ , 则子代是改变父代  $\text{cropos}$  后的基因而得,具体生成的两后代  $S_a, S_b$  分别为:

$$S_a = (V_1, V_2, \dots, V_{\text{cropos}}, (\text{int})(a_1 * V_{\text{cropos}+1} + (1-a_1) * W_{\text{cropos}+1}), \dots, (\text{int})(a_{n\text{-cropos}} * V_m + (1-a_{n\text{-cropos}}) * W_m),$$

$$S_b = (W_1, W_2, \dots, W_{\text{cropos}}, (\text{int})(a_1 * W_{\text{cropos}+1} + (1-a_1) * V_{\text{cropos}+1}), \dots, (\text{int})(a_{n\text{-cropos}} * W_m + (1-a_{n\text{-cropos}}) * V_m).$$

在深基坑支护的协同演化模型中,第一步是对方案的演化。同样假设父代为  $S_1 = (V_1, V_2, \dots, V_{L1})$  和  $S_2 = (W_1, W_2, \dots, W_{L1})$ ,假定  $L1$  代表方案变量个数。则交叉的步骤是:

a. 取随机数  $sf = \text{srand}(L1)$ ,对  $V_1 \dots V_{L1}$  基因按步骤2生成  $L1 - sf$  个  $(0, 1)$  区间的随机数,按上述方法改变  $L1 - sf + 1$  到  $L1$  的各基因的值。设生成的后代个体为:

$$C_1 = (V_1, \dots, V_{sf}, M_{sf+1} \dots M_{L1}),$$

$$C_2 = (W_1, \dots, W_{sf}, N_{sf+1} \dots N_{L1}).$$

b. 对某一方案个体, 将细部种群作交叉:

选择细部的交叉点  $sx = \text{rand}(L2)$ . 方案部分不变. 按上述方法改变  $L2 - sx + 1$  到  $L2$  的各基因的值,

$$C_1 = (V_1, \dots, V_{sx}, M_{sx+1} \dots M_{L2}),$$

$$C_2 = (W_1, \dots, W_{sx}, N_{sx+1} \dots N_{L2}).$$

### 2.2.5 变异操作

算法中, 采用均匀变异的方法对个体进行变异. 首先在父代中随机地选择某一个体作为父本, 然后在父本的向量中随机地选择一个分量, 在该分量的取值范围内随机选取一个值, 作为子代该分量的新值, 而其他分量不变. 如从父代种群中随机选出的父本为  $S_1 = (V_1, V_2, \dots, V_m)$ , 变异点是在  $m$  个分量中随机取一个分量, 设为  $k (k \leq m)$ , 若  $S_2 = (W_1, W_2, \dots, W_m)$  是该父本变异后产生的后代. 则是在区间  $[a_i, b_i]$  随机取一个数  $V'_k = \text{rand}(a_i, b_i)$  取代  $V_k$  而得, 即

$$W_i = \begin{cases} V_i & i \neq k, \\ V'_i & i = k. \end{cases}$$

### 2.2.6 修正算子

设定交叉率  $P_c$ , 变异率  $P_m$ , 演化终止所要经历的代数  $\text{gens}$  及每代的种群规模  $\text{popsize}$ . 一般地说, 系统给定的默认值:  $P_c = 0.5 \sim 0.7, P_m = 0.1 \sim 0.2$ . 种群规模、染色体的基因组数 (即变量个数) 满足关系:

$$\text{Popsiz} > 2 \times \text{paranumer}.$$

在执行交叉和变异时, 每生成新一代, 就必须对该新一代实施修正算子. 在演化的初始阶段, 应该有较高的交叉和变异率, 这样可以更快地趋向于最优解; 在演化的后期, 为了保持最优解的稳定性, 应降低交叉和变异率.

算法中相应的语句为:

```
static double delta1 = (x_PM_max - x_PM_min) / maxGen;
.....
if ( (double) (rand () % 101 / 100.0) < f_PM_max - gen * delta1)
    child11 [k] = mutate (k);
if ( (double) (rand () % 101 / 100.0) < f_PM_max - gen * delta1)
    child22 [k] = mutate (k);
k++; gen++;
/* 对第 k 个基因判断是否进行变异, 同时按代数 gen 平均减少变异率
```

## 3 结束语

在深基坑支护工程的优化设计中引入了遗传算法, 所设计的遗传优化设计算法使优化设计不再受设计空间的可微性、连续性的限制, 同时, 增加获得全局最优解的可能性. 实验表明, 算法的运行具有较好的有效性和稳定性, 它为求解像支护工程这类具有离散设计变量和非确定性因素的工程设计问题提供一种新的途径和方法.

### 参考文献

1 周明栋. 遗传算法原理及应用. 北京: 国防工业出版社, 2000.