求解整数规划的混合遗传算法

宁伟华，　陈绍顺，　王凤山
（空军工程大学 导弹学院，陕西 三原　713800）

摘　要：整数规划问题是NP困难问题，在决策变量和约束条件很多时，用常规的求解法效率很低。针对遗传算法存在的缺陷，提出了遗传算法和混沌的结合。在此基础上，设计了用于求解整数规划的混合遗传算法，算例表明，此算法能大幅提高计算效率。

关键词：整数规划；遗传算法；混沌
中图分类号：O221.4　　**文献标识码**：A　　**文章编号**：1009－3516(2004)06－0080－04

整数线性规划(ILP)的标准形式如下：

$$\begin{aligned} &\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.}\quad &\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}=b_i, i\in M \\ &x_j\in \mathbf{Z}^+, j\in N \end{aligned} \tag{1}$$

其中 $\boldsymbol{Z}^+$ 表示全体非负整数的集合，M 表示等式约束的约束指标集合，N 表示非负决策变量的指标集合。

整数规划问题是NP困难问题[1]。常用的求解整数规划问题的分枝界定法、割平面法和枚举法等仅能用于求解规模较小的问题，当 M 和 N 较大时，计算效率很低，为此，本文设计了用于求解标准形整数线性规划的混合遗传算法。

1　求解整数规划的遗传算法设计

1）编码。对于整数规划，宜采用十进制编码，这样既可以省去许多编码解码工作，又可以提高计算的精度。可将约束条件作以下变形：

$$\begin{cases} x_n=\varphi(x_1,x_2,\cdots,x_{n-1}) \\ x_j\in \mathbf{Z}^+,\quad j=1,2,\cdots,n \end{cases} \tag{2}$$

式中，函数 φ 是变量组 $(x_1,x_2,\cdots,x_{n-1})$ 的一个多项式表达式，用来表示 x_n。

2）产生初始群体。初始群体是遗传算法搜索寻优的出发点。群体规模越大，搜索的范围越广，但是每代的遗传操作时间越长。反之，群体规模越小，每代的运算时间越短，然而搜索空间也越小。初始群体的每个个体是随机产生的，个体中的每个整数采用下式计算：

$$x_j=[(x_{j\max}-x_{j\min})\times \mathrm{Rand}+x_{j\min}] \tag{3}$$

式中，[ ]为取整函数，x_j、$x_{j\max}$、$x_{j\min}$ 分别为个体中第 j 个基因的取值和第 j 个基因允许取的最大值和最小值，Rand为[0,1]区间内的随机数。

在整数规划问题中，$x_{j\max}=\max(b_i)$、$x_{j\min}=0$，因而式(3)可简化为 $x_j=[\max(b_i)\times \mathrm{Rand}]$，$j=1,2,\cdots,n-1$，用 $x_n=\varphi(x_1,x_2,\cdots,x_{n-1})$ 求取 x_n。当 $x_n<0$ 时，需重新选取个体中的基因 $x_j(i=1,2,\cdots,n-1)$，直到 $x_n\geqslant 0$ 为止。

3）计算适应度。适应度是衡量个体优劣、执行遗传算法“优胜劣汰”的依据。在求解整数规划问题中，适应度函数 *fitness*(x) 按下式求取：选取一个相对较大的正数 P，当 $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}<p$ 时，取 *fitness*$(x)=p-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$；当 $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$

收稿日期：2004－04－25
作者简介：宁伟华(1976－)，男，山西稷山人，博士生，主要从事防空作战决策分析.

$\geqslant p$ 时,取 $fitness(x)=0$

4) 复制。在遗传算法中通过复制,将优良的个体插入下一代新群体,体现"优胜劣汰"的原则。本文采用轮盘法和最优保留策略相结合的方法。轮盘法的实质是个体被选中复制到下一代的概率取决于该个体的相对适应度 $p_i:p_i=\frac{f_i}{\sum f_i}$,式中 f_i 为个体 的适应度, $\sum f_i$ 为群体的累加适应度。轮盘法的操作步骤为:①把群体中所有个体的适应度按从小到大排成一序列并重新命名。命名规则是如果个体 i 的适应度 p_i 排在第 j 位,则个体 i 的适应度命名为 q_j;②随机生成[0,1]之间的随机数 α ,若满足: $\sum_{i=1}^{j-1}q_i\leqslant\alpha\leqslant\sum_{i=1}^{j}q_i$,则选取 q_j 对应的个体 i 复制到下一代;③若群体的规模为 Q,则重复 Q 次第二步操作。最优保留策略是为了保证群体中最优良的个体能确保遗传到下一代,以加快遗传算法的收敛性。在找出当前群体中适应度最高的个体和适应度最低的个体后,若当前群体中最佳个体的适应度比总的迄今为止最好个体的适应度还要高,则以当前群体中的最佳个体作为新的迄今为止的最好个体,并用迄今为止的最好个体替换掉当前群体中的最差个体。

5) 交换。在遗传算法中,交换是产生新个体的主要手段。在整数规划问题中,采用综合交换的方法。首先,随机确定交换点 ∂ 。为满足约束条件,在[1,$n-1$]区间内产生均匀分布的随机整数,该整数便是交换点的位置。将父代的个体写作:

$$父代个体1:(x_{11},x_{12},\cdots,x_{1\partial},\cdots,x_{1(n-1)},x_{1n})$$
$$父代个体2:(x_{21},x_{22},\cdots,x_{2\partial},\cdots,x_{2(n-1)},x_{2n})$$

按下式计算交换点的新基因:

$$\begin{cases}x'_{1\partial}=[x_{1\partial}-\beta(x_{1\partial}-x_{2\partial})]\\x'_{2\partial}=[x_{2\partial}-\beta(x_{1\partial}-x_{2\partial})]\end{cases}\tag{4}$$

式中,β 为[0,1]区间内的随机数。随后,把交换点以后父代个体基因进行交换,得到两个新的个体:

$$子代新个体1:(x_{11},x_{12},\cdots,x'_{1\partial},\cdots,x_{2(n-1)},x_{2n})$$
$$子代新个体2:(x_{21},x_{22},\cdots,x'_{2\partial},\cdots,x_{1(n-1)},x_{1n})$$

为了保证产生的新个体满足约束条件需重新计算 x_{1n},x_{2n}:

$$\begin{cases}x_{2n}=\varphi(x_{11},x_{12},\cdots,x'_{1\partial},\cdots,x_{2(n-1)},x_{2n})\\x_{1n}=\varphi(x_{21},x_{22},\cdots,x'_{2\partial},\cdots,x_{1(n-1)},x_{1n})\end{cases}$$

若 $x_{2n}<0$ 或 $x_{1n}<0$,须重新执行交换操作,直到 $x_{2n}>0$ 且 $x_{1n}>0$ 为止。

6) 突变。突变是遗传算法产生新个体的另一种方法。在整数规划问题中,为了满足约束条件,突变点的位置限制在第1位到第($n-1$)位之间,并且基因的突变是在该数允许的范围内变动。因为我们总希望在遗传进化的初期加大突变量,使得搜索能够在整个定义域内迅速扩散,而在进化后期到收敛点附近时减少突变量,以加快收敛。这里我们借用了模拟退火的思想。设第 t 代的个体为 $x(x_1,x_2,\cdots,x_i,\cdots,x_{(n-1)},x_n)$,若 x_i 被选为突变点,则新个体为 $x'(x_1,x_2,\cdots,x_i,\cdots,x_{(n-1)},x_n)$,其中 x'_i 按下式计算: $x'_i=x_i+\varepsilon(t,y)$,其中 $y=x_{i\max}-x_{i\min},x_{i\max},x_{i\min}$ 分别为 x_i 允许的最大值和最小值,故 $y=\max(b_i)$。则

$$\varepsilon(t,y)=[y(1-\alpha^{(1-\frac{t}{T})^b})]\tag{5}$$

式中, α 为区间[0,1]内的随机数,T 为最大允许迭代次数,t 为当前的迭代次数,b 为系数。$\varepsilon(t,y)$ 的目的是使迭代后期当时 $t\to T$ 时突变愈来愈小,而在初期 $\varepsilon(t,y)$ 较大,突变在大范围内进行。b 选取的不同,搜索速度也不一样。同理,取 $x_n=\varphi(x_1,x_2,\cdots,x'_i,\cdots,x_{1(n-1)})$,若 $x_n<0$,则须重新执行突变操作直到 $x_n>0$ 为至。

7) 终止。遗传算法是一个反复迭代的过程,每次迭代要执行适应度计算、复制、交换、突变等操作,直至满足终止条件。对于整数规划问题,终止的方法一种是指定最大迭代次数 T,一旦遗传算法的迭代次数达到 T,则停止操作,输出结果;另一种是计算相邻两代输出的最优解之差,等精度达到预定要求时,停止迭代。

2　遗传算法与混沌的结合

遗传算法能保证种群不断变化,使所求的解不断变优,但它往往缺乏产生最优个体的强大能力,导致在搜索全局最优解时速度变慢,甚至陷入局部最优解[2~3],具有初值敏感性、内在随机性、遍历性及规律性等特点,在一定范围内能按其自身的"规律"不重复地遍历所有状态[4]。混沌优化过程分两个阶段进行:首先,

在变量的取值范围内依次遍历经过的各点，接受较好点为最优点;然后，以当前最优点为中心，附加一混沌小扰动，进行细搜索寻找最优点。混沌优化方法在搜索空间小时效果显著，但搜索空间大时其效果却不能令人满意。因此遗传算法与混沌的结合能较好地弥补各自的缺陷,使搜索快速达到全局最优,提高计算效率。

2.1 混沌优化算法

设优化问题的一般形式为：$\min f(x)$

$$a_i \leqslant x_i \leqslant b_i, i=1,2,\cdots,n \qquad x=(x_1,x_2,\cdots,x_n)$$

选取常用的混沌模型,即一维的 Logistic 映射,模型如下：

$$t_{k+1}=ut_k(1-t_k) \tag{6}$$

$k=0,1,2,\cdots,0\leqslant t_k<1$,$u$ 为控制变量,当 $u\in(3.56,4.0)$ 时,式(6)进入混沌状态,具有混沌的一般特性,当 $u=4$ 时,式(6)处于完全混沌状态[5]。混沌优化算法的操作步骤如下：

步骤1 初始化。对式(6)中分别赋予 n 个具有微小差异的初值,得到 n 个轨迹不同的混沌变量 $t_i(k)$，$k=1$,$f^*=f(x^*)$,x^* 为当前最优解。

步骤2 用混沌变量进行搜索。

$$x_i(k)=x_i^*+c_it_i(k)-d_i \qquad c_i,d_i>0 \tag{7}$$

计算性能指标 $f(k)=f(x(k))$,$x(k)=(x_1(k),x_2(k),\cdots x_n(k))$

步骤3 若 $f(k)<f^*$,则 $x_i^*=x_i(k)$,$f^*=f(k)$,如给定的迭代次数已达到,则结束;否则,转步骤2 。

2.2 遗传算法与混沌优化算法的结合

在遗传算法中,当种群完成一次进化(选择、复制、交换、变异)后,利用上述算法对部分个体作混沌搜索,以求适应度更大的最优个体,引导种群进化。由于对适应度更大的几个个体做局部运算,更易得到新的最优个体,为了更快的搜索出最优个体,仅让适应度值较大的部分个体有机会参与混沌搜索算法。具体操作方法是,以遗传算法的适应度函数为混沌算法的目标函数,以参与混沌搜索的个体为当前最优解(即混沌算法步骤1 中 x^*),按照混沌算法流程分别对每个参与混沌搜索的个体进行混沌运算,当所有混沌搜索进化结束后,选出适应度最大的个体作为当前最优解参与遗传算法的下一代进化。

综上所述,求解整数规划的混合遗传算法的操作步骤如下：

步骤1 初始化常量值。确定群体的规模 m 、交叉概率 p_c、变异概率 p_m、混沌模型中的控制变量 u 、遗传算法的最大迭代代数 T 、混沌搜索的给定迭代次数 T'。

步骤2 对整数规划进行十进制编码,生成初始群体,计算各个体适应度。

步骤3 选用轮盘法和保留最优策略相结合的方法选择复制个体。

步骤4 以交换概率 p_c 按综合交换方式对个体进行交换操作。

步骤5 以变异概率 p_m 按文中所述变异方式对个体进行变异操作。

步骤6 对适应度最大的部分个体(2 ~4 个)做混沌优化运算,若新的个体适应度大于原个体,则替换原个体,否则原个体不变。

步骤7 重复步骤2 至步骤6,直到终止条件得以满足。

3 算例

对如下整数规划问题：$\max f(x)=\sum_{k=1}^{5} w_k(1-e^{-px_k})$

$$\text{s. t.} \qquad x_1+x_2+x_3+x_4+x_5=12$$

$$x_i\in z^+ \quad w_1=0.85,w_2=0.5,w_3=0.7,w_4=0.4,w_5=0.3,p=0.7$$

因本例计算规模较小,故确定群体规模 m 为 10,个体为 (x_1,x_2,x_3,x_4,x_5) ,且 $x_5=\varphi(x_1,x_2,x_3,x_4)=12-x_1-x_2-x_3-x_4$,针对本算例,为确保初始群体中不出现非可行解,编码方式如下：

$$x_1=[12\times \text{Rand}]$$

$$x_2=[(12-x_1)\times \text{Rand}]$$

$$x_3 = [(12 - x_1 - x_2) \times \mathrm{Rand}]$$
$$x_4 = [(12 - x_1 - x_2 - x_3) \times \mathrm{Rand}]$$
$$x_5 = 12 - x_1 - x_2 - x_3 - x_4$$

其中,[ ]为取整函数,Rand 为[0,1]区间内的随机数。适应度函数取为目标函数,交叉概率 P_c 为 0.8,变异概率 P_m 为 0.1,混沌模型中的控制变量 u 为 4,指定遗传算法的最大迭代代数 T 为 100,要求的计算精度为 0.001,混沌搜索的给定迭代次数 T' 为 30。在计算机上仿真计算的简要结果见表 1(表中记录的是每一代群体中的最优解)。

表 1　利用混合遗传算法求解优化结果

迭代次数	x_1	x_2	x_3	x_4	x_5	$\max F(x)$
1	5	1	2	3	1	2.105 46
3	4	2	2	1	3	2.167 02
4	4	3	3	1	2	2.216 68
7	3	2	2	3	2	2.227 03
10	4	2	2	2	2	2.229 77
11	3	2	3	3	1	2.238 93
13	4	3	3	2	1	2.2411 67
14	4	3	3	2	1	2.2411 67

从计算结果可看出,此算法收敛较快,当迭代 13 次后,已得出最优解,继续进行迭代,其最优解保持不变。

4　结束语

用现代进化算法求解传统的优化问题具有其独到的优势,遗传算法以其适应性广,无需背景知识,搜索能力强等优点被广泛应用于各类优化问题中。鉴于遗传算法自身存在的一些不足,学术界提出了许多弥补办法。比如,遗传算法与模拟退火的结合,遗传算法与混沌的结合等。本文提出了一种遗传算法与混沌的结合方式,并设计了用于求解整数规划的混合遗传算法,算例表明此方法能大幅提高计算效率,有较大的实用价值。

参考文献:

[1]　马振华,陈景良,刘坤林,等. 现代应用数学手册,运筹学与最优化理论卷[M]. 北京:清华大学出版社,1999.
[2]　邢文训,谢金星. 现代优化计算方法[M]. 北京:清华大学出版社,1999.
[3]　云庆夏. 进化算法[M]. 北京:冶金工业出版社,2000.
[4]　张春慨,王亚英. 混沌在实数编码遗传算法中的应用[J]. 上海交通大学学报,2000,12(12):1658 - 1660.
[5]　雷德明. 利用混沌搜索全局最优解的混合遗传算法[J]. 系统工程与电子技术,1999,21(12):81 - 83.

(编辑:田新华)

A Hybrid Genetic Algorithm for Solving Integer Programming

NING Wei - hua, CHEN Shao - shun, WANG Feng - shan
(The Missile Institute, Air Force Engineering University, Sanyuan, Shaanxi 713800, China)

Abstract: Integer programming problem is NP problem. The efficiency is low by using the routine methods to solve integer programming problem when there are many variables and many restrictions. Aiming at the existing shortcomings in genetic algorithm, the combination between genetic algorithm and chaos is presented. On this basis, the hybrid genetic algorithm for solving integer programming problem is devised. The example shows that the algorithm is very effective in increasing the computing efficiency.

Key words: integer programming; genetic algorithm; chao