一种快速收敛的遗传算法*

周春光 周国芹 李 博 程彦丰 梁艳春

(吉林大学计算机科学系 长春 130023) (吉林大学数学系 长春 130023)

摘要 本文针对传统遗传算法收敛速度慢的缺点,提出了一种改进的快速收敛算法,并以八皇后问题为例进行了数值模拟,实验结果表明这种改进的遗传算法在收敛速度上大大优于传统遗传算法.

关键词 遗传算法,快速收敛,染色体段,段适应度函数,段群体.

遗传算法GA(genetic algorithm)[1]是模拟自然界生物进化过程的计算模型,它依据达尔文的自然选择规则理论,对包含可能解的群体反复使用基于遗传学的操作,使群体不断优化,以求得满足要求的最优解或准最优解.

GA比起其它搜索算法,主要优点是简单、通用、鲁棒性强,但其不足之处是由于搜索空间大,计算量大,因而收敛速度慢.特别是当求解的问题复杂时,使得所构造的染色体(Chromosome)的结构及长度增大,搜索空间增大,收敛速度慢的问题就显得尤其严重.本文针对这一问题,提出了一种改进的快速收敛遗传算法,并以八皇后问题为例加以验证,实验结果表明了这一方法的有效性.

1 快速收敛的遗传算法

传统的遗传算法[2]是依据求解问题来确定染色体编码形式及定义串长度,然后初始化随机生成的N个染色体并组成一个群体(Population),再对群体中的染色体实施遗传操作以搜索代表优化解的染色体.

本文提出的快速收敛遗传算法增加了对染色体的分割(Dividing)与重组(Recombination)操作.分割意指将染色体的定长串分割成若干段,每段内可能包含若干个基因(Gene).该算法首先依据于各段的结构和段长,分别初始化随机生成的N个染色体段(Chromosome Segment)组成的段群体(Segmented Population),然后对各段群体独自实施遗传操作以寻找优化段,最后,再对选出的优化段实施重组操作,重新组合成完整的染色体来搜索优化解.

* 本文研究得到国家自然科学基金资助.作者周春光,1947年生,副教授,主要研究领域为人工神经网络,遗传算法及应用.周国芹,女,1970年生,硕士,主要研究领域为遗传算法及分布式神经计算.李博,1964年生,讲师,主要研究领域为模糊控制,人工神经网络算法.程彦丰,1961年生,讲师,主要研究领域为人工神经网络算法及其应用.梁艳春,1953年生,副教授,主要研究领域为参数识别,人工神经网络算法与应用.

本文通讯联系人:周春光,长春 130023,吉林大学计算机科学系

本文1995-09-12收到修改稿

快速收敛遗传算法的步骤如下：

(1)确定染色体 C 的编码形式，选定 C 的适应度函数 $F(C)$；

(2)对染色体 C 按其结构及长度 L 实施分割操作，生成 k 个染色体段. 基于每个染色体段各自随机地生成 k 个段群体，$PU_i=\{C_{ij}\}$，其中 $i(1,2,\ldots,K)$ 为段群体号，$j(1,2,\ldots,N)$ 为段群体内染色体号；

(3)定义染色体段的段适应度函数为：

$$f^{g+1}(C_{ij}^{g+1})=F(C_1^{g+1})=F(C_{11}^{g+1}UC_{21}^{g+1}\ldots UC_{i-1,1}^{g+1}UC_{ij}^{g}UC_{i+1,1}^{g}U\ldots UC_{k1}^{g}) \quad (1)$$

其中上标 $g(g\geqslant 1)$ 表示代(*Generation*)数，$C_{i1}^{g}(i=1,2,\cdots,k)$ 是依据段适应度函数 $f^{g}(C_{ij}^{g})(j=1,2,\ldots,N)$ 已求得的第 g 代的各段群体中的最优染色体段，而 $g+1$ 代的各段群体中的各染色体段的适应度值由 $f^{g+1}(C_{ij}^{g+1})$ 求得，对第 i 段群体求其 $g+1$ 代最优染色体段 C_{i1}^{g+1} 时要用到 $g+1$ 代最优染色体段(当段号小于 i 时)和 g 代最优染色体段(当段号大于 i 时)，依此式求得的 $f^{g+1}(C_{ij}^{g+1})$ 越大，表示段 C_{ij}^{g+1} 越优，最后在 N 个染色体段中选取一个最优的命名为 C_{i1}^{g+1}. 由(1)式定义的染色体的段适应度函数可见，f^{g+1} 是动态变化的，即 f^{g+1} 随 g_i 代和 $g+1$ 代各段群体中的最优染色体段的不同而不同；

(4)依据(1)式计算各 PU_i 中染色体的段适应度 $f^{g+1}(C_{ij}^{g+1})$，并将段群体内的染色体分别按其段适应度值由大到小排序；

(5)搜索过程由下面循环体实现.

```
g=0;
for( ; ; ){
        g=g+1;
        for (i=1;i<=k;i++)
                {对段群体 PUi 按传统的遗传算法实施遗传操作产生 2m 个子染色体，2m<<N；
                替换 PUi 中段适应度值小的 2m 个染色体，即 PUi 中的后 2m 个染色体；
                for (j=1;j<=N;j++)
                    {Cf^{g+1}=Cf1^{g+1}UCf2^{g+1}...UCfi-1,1^{g+1}UCfiUCfi+1,1U...UCfk1;
                    计算 F(Cf^{g+1})；
                    if (Cf^{g+1} 满足目标解)
                            解码 Cf^{g+1}，得到目标解，算法结束；
                    else
                            f^{g+1}(Cfj^{g+1})=F(Cf^{g+1})；
                    }
                按 PUi 中段适应度值由大到小将染色体重新排序；
                }
}
```

假定一次遗传操作产生 $2m$ 个子染色体，那么传统的 *GA* 需要计算适应度的次数为 $2m$ 次；改进的算法对于一个段群体的一次遗传操作若产生 $2m$ 个子染色体，则实际上对于 K 个段群体来说相当于产生了 $2m\cdot N^{k-1}\cdot k$ 个子染色体，而对于适应度的计算量为 $K\cdot N$ 次；对于传统的 *GA* 来说，若产生 $2m\cdot N^{k-1}\cdot k$ 个子染色体，则计算子染色体适应度的量亦为 $2m\cdot N^{k-1}\cdot k$ 次，显然下式成立：$2m\cdot N^{k-1}\cdot k>k\cdot N$. 因而改进的算法在产生同样个数的子染色体时能大大减少其适应度的计算量，所以能大幅度加快收敛过程.

2 实验结果

本文以八皇后问题为例对快速收敛的遗传算法进行了验证. 八皇后问题是一个 NP 问

题,是把八个皇后摆在8·8的棋盘上,使得没有一个皇后能俘获任何别的皇后,即每一行、列仅有一个皇后,每条对角线上至多有一个皇后.

(1)染色体的编码

八皇后问题的解可以用一个8·8矩阵来表示:

$$A=\begin{vmatrix} A11 & A12 & \cdots & A18 \\ \cdots & & & \\ A41 & A42 & \cdots & A48 \\ \cdots & & & \\ A81 & A82 & \cdots & A88 \end{vmatrix} \tag{2}$$

染色体的编码为{0,1}上长度为64的位串,1表示该位置有皇后,0表示该位置无皇后.

(2)染色体的分割

本实验将染色体分别分割为2段、4段和8段,下面给出当$K=4$时各段所包含的位串.

$$\begin{aligned} &(A_{11}\ldots A_{18}A_{21}\ldots A_{28})\in PU_1 \\ &(A_{31}\ldots A_{38}A_{41}\ldots A_{48})\in PU_2 \\ &(A_{51}\ldots A_{58}A_{61}\ldots A_{68})\in PU_3 \\ &(A_{71}\ldots A_{78}A_{81}\ldots A_{88})\in PU_4 \end{aligned} \tag{3}$$

(3)适应度函数的选取

定义函数:

$$E_1=\sum_{i=1}^{8}\left|\sum_{j=1}^{8}A_{ij}-1\right| \tag{4}$$

$$E_2=\sum_{j=1}^{8}\left|\sum_{i=1}^{8}A_{ij}-1\right| \tag{5}$$

$$E_3=\sum_{k=1}^{2*8-1}\delta\left(\sum_{i=1}^{8}\sum_{j=1}^{8}g_1(A_{ij})-1\right) \qquad g_1(A_{ij})=\begin{cases} A_{ij} & \text{若 } i+j-1=k \\ 0 & \text{否则} \end{cases} \tag{6}$$

$$E_4=\sum_{k=1}^{2*8-1}\delta\left(\sum_{i=1}^{8}\sum_{j=1}^{8}g_2(A_{ij})-1\right) \qquad g_2(A_{ij})=\begin{cases} A_{ij} & \text{若 } i-j+8=k \\ 0 & \text{否则} \end{cases} \tag{7}$$

$$E=E_1+E_2+E_3+E_4 \tag{8}$$

其中i,j分别表示行号和列号;

$$\delta(x)=\begin{cases} x & \text{若 } x>0 \\ 0 & \text{若 } x\leqslant 0 \end{cases} \tag{9}$$

上述诸函数的定义使得$E_1=0$和$E_2=0$保证每行和每列上有且仅有一个皇后,$E_3=0$和$E_4=0$保证每条对角线上至多有一个皇后,显然满足目标解时$E=0$.适应度函数定义为:

$$F=1/(E+1) \tag{10}$$

由定义可知$0<F\leqslant 1$.当找到目标解时$F=1$,算法结束.

图1为使用传统遗传算法解决八皇后问题的适应度变化曲线,图2为使用快速收敛的遗传算法时的适应度变化曲线,其中图2给出了$k=2$、4、8的3种情况.从图1和图2对比可以看出快速收敛的遗传算法极大地加快了收敛速度.

3 结论与讨论

本文所提出的一种基于将染色体分割重组的方法,在实施遗传操作时对各段群体独立

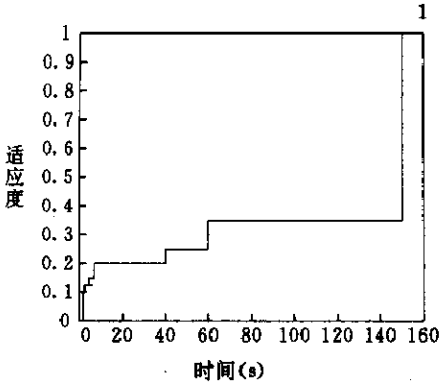

图1 采用传统GA的适应度变化曲线

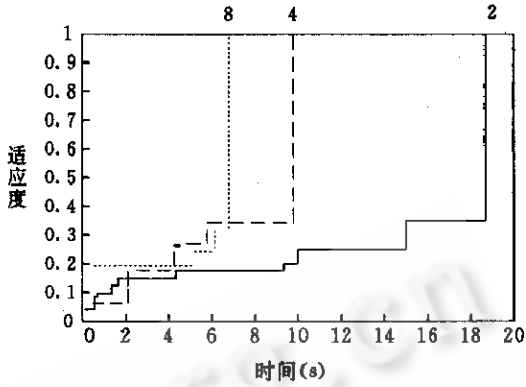

图2 采用快速GA的适应度变化曲线

进行,段重新组合后又不致于减少搜索空间的大小,因而能加快收敛过程并能找到目标解,通过八皇后问题验证了该方法的有效性.另外,使用该方法对函数如 $\lg(x)$和 $\mathrm{Sin}(x)$的求解也进行了模拟实验,收到了同样的效果.

需要指出的是,由于根据(1)式定义的染色体段的适应度函数是动态变化的,所以在该方法的运行过程中,存在排除掉一些更好的染色体的可能性,即不是或不完全是由上一代的最优的染色体段组合的染色体,其全局的适应度反而可能更好,但根据对于八皇后问题和某些函数的求解等问题的模拟实验表明,这种失去一些更好的组合的概率较小,因此该方法用于这类问题能够保证几倍、十几倍甚至几十倍地提高收敛速度.该方法对于其它类型问题的应用范围与效果尚有待进一步研究.

参考文献

1 Holland J H. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press, 1975.

2 Booker L B, Goldbery D E, Holland J H. Classifier systems and genetic algorithms. Artificial Intelligence, 1989, 40(1~3):235~282.

A FAST CONVERGENT GENETIC ALGORITHM

Zhou Chunguang Zhou Guoqin Li Bo Cheng Yanfeng

(*Department of Computer Science Jilin University Changchun* 130023)

Liang Yanchun

(*Department of Mathematics Jilin University Changchun* 130023)

Abstract This paper proposes a fast convergent genetic algorithm to overcome the slow convergent rate of conventional genetic algorithms, and takes 8—queen problem as an example to examine the algorithm. The simulated results demonstrates that the algorithm's convergent rate is faster than that of the conventional techniques.

Key words Genetic algorithm, fast convergent, chromosome segment, segment fitness function, segmented population.