

一种具有个体学习能力的演化算法^{*})

A New Evolutionary Algorithm with Individual Learning Capability

何 峰 康立山 陈毓屏

(武汉大学软件工程国家重点实验室 武汉430072)

Abstract A new concept, fitness function of gene, is proposed in this paper. Using the fitness of gene to control the individual mutation and direct the individual learning greatly improves the search efficiency and local search capability of evolutionary algorithm. The comparison experiments in solving N-queens problem demonstrate the advantage of this new evolutionary algorithm. The average search efficiency of this algorithm is 1000 times higher than ordinary evolutionary algorithm.

Keywords Evolutionary algorithm, N-queens problem, Individual learning

1 引言

70年代由美国学者 John H. Holland 提出的遗传算法是一种借鉴生物界自然选择思想和自然遗传机制的全局搜索算法^[1]。它把问题的每一个可能解看作一个个体,而个体的集合则形成种群。算法在整个种群空间内随机搜索,按一定的评价函数对当前种群中每一个个体进行评价,然后基于个体的评价按优胜劣汰原则选择出父体,再用变异、杂交等遗传算子作用于父体以产生新的个体形成下一代种群,如此反复,使种群不断进化,直至产生最终的解。因为演化算法在解决大空间、非线性、全局寻优等复杂问题时具有传统方法所不具备的独特的优越性,所以它得到了广泛的研究和应用。

然而演化算法源于对生物演化过程的直接模拟,它在应用中有两个严重的缺点,即收敛速度慢和存在早熟收敛^[2]。由于遗传算法决定了演化算法的搜索路径,在演化算法中有决定性的作用。因此本文认为要改善演化算法的收敛性,根本的着手点是改进遗传算子。本文在对已有的演化算法中的问题进行分析之后,提出了一种新的演化算法,并通过求解 N 皇后问题的对比实验,展示了新算法的优越性。

2 一种新的演化算法

用演化算法求解问题时,需要将问题的表现型解空间编码,映射到演化算法能操作的基因型解空间。基因型解空间中的每一个个体由一系列基因组成,代表了一个候选解。一个由 m 个基因组成的个体可以表示成一个 m 元组: $(a_1, a_2, \dots, a_m)$ 。这个元组中的每一个位置称为一个基因位,在每一个基因位上都有一个可选值的集合, $A_i \in \{g_{i1}, g_{i2}, \dots, g_{in_i}\}$, 称为个体中第 i 个基因位的等位基因。基于这些概念,演化算法中的变异操作是指:给定一个个体 $I = (g_{11}, g_{21}, \dots, g_{m1})$, 产生一个新的个体 $I' = (g_{11}, g_{21}, \dots, g_{xy}, \dots, g_{m1})$, 其中 $x = \text{random}(1, m)$, $y = \text{random}(1, n_x)$ 。杂交操作(如一点杂交)则是指:给定两个个体 $I_1 = (g_{11}, g_{21}, \dots, g_{m1})$ 和 $I_2 = (h_{11}, h_{21}, \dots, h_{m1})$, 产生两个新的个体 $I' = (g_{11}, g_{21}, \dots, h_{xy}, \dots, h_{m1})$ 和 $I'_2 = (h_{11}, h_{21},$

$\dots, g_{xy}, \dots, g_{m1})$, 其中 $x = \text{random}(1, m)$ 。变异和杂交操作产生新的个体都直接加入到新一代种群中,参与下一轮的竞争。

上面描述的演化算法的遗传操作,有两个方面可以改进:

1) 引入“基因适应函数”的概念,把选择策略引入到变异操作中。基因适应函数用于计算一个个体中各个基因的适应值。新的变异算子基于基因的适应值,运用某种选择策略优先选择较差的基因进行变异。这样可以避免传统变异操作的搜索行为过于盲目的问题。传统变异操作把个体中的所有基因位都同等看待,如果一个问题基因数目较多,每个基因的等位基因数目也很多,则它的搜索效率很低。

2) 通过增加个体学习算子来提高演化算法的搜索效率和局部搜索能力。如果把遗传操作看成是决定一个新个体适应能力的先天因素,那么一般的演化算法中缺少个体可通过后天学习来提高适应能力。虽然遗传操作的随机性有使种群保持多样性的优点,但它也使演化算法在搜过程中缺乏目的导向性,导致算法中存在大量的无效搜索。由遗传操作得到新的个体,再让它进行一次学习能弥补演化算法的这个缺陷。

新的演化算法的框架如下:

- (1) $t = 0$;
- (2) 初始种群 $P(t) = \{I_1, I_2, \dots, I_n\}$;
- (3) 计算 $P(t)$ 中个体及基因的适应值;
- (4) 对 $P(t)$ 中的每个个体应用学习算子;
- (5) while(不满足终止条件){
- (6) 由 $P(t)$ 通过遗传与选择操作形成新的种群 $P(t+1)$;
- (7) 计算 $P(t+1)$ 中个体及基因的适应值;
- (8) 对 $P(t+1)$ 中的每个个体应用学习算子;
- (9) $t = t + 1$;
- (10)}

3 实验

我们用 N 皇后问题来检验这个新的演化算法。N 皇后问题是 8 皇后问题的推广,要求在 $N \times N$ 的棋盘上摆放 N 个皇后,使这 N 个皇后不互相攻击,即没有两个皇后在同一行、同一列、同一对角线上。

3.1 一般演化算法的 N 皇后问题求解

如果规定棋盘上每一列只放一个皇后, N 皇后问题的每

^{*}) 国家自然科学基金资助项目(60133010, 60073043, 70071042)。何 峰 博士生, 主要研究领域为演化算法、面向对象软件工程方法; 康立山 教授, 博士导师, 主要研究领域为计算机科学理论、计算数学、并行计算、演化算法; 陈毓屏 副教授, 主要研究领域为并行计算、演化算法、计算机软件。

一种可能的摆放格局就可以用一个 N 元数组 $queen[N]$ 表示,第 i 列的皇后在棋盘上的坐标为 $(i, queen[i])$ 。一个格局中,一个皇后攻击到的所有皇后数目称为这个皇后的攻击数。由于皇后之间的攻击是对称的,一个皇后的攻击数也表明了有多少个皇后攻击了这个皇后。而所有皇后的攻击数之和称为这个格局的总攻击数。显然,总攻击数为零的格局为 N 皇后问题的一个解。

```
struct Individual
{
    int queen[N]; // 每列皇后所在行号
    int total_attacks; // 该个体对应格局下的总攻击数
};
```

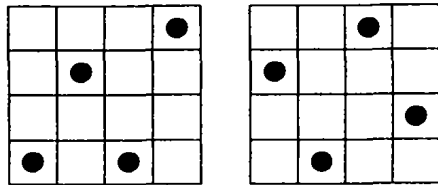


图1 4皇后问题的两个格局及其表示

由于每列上只有一个皇后,相互攻击的皇后必然出现在

表1 一般演化算法的 N 皇后问题求解,数据格式:演化时间(秒)/评价函数运行次数

		实验编号										平均值
		1	2	3	4	5	6	7	8	9	10	
皇后数目	20	0.326/ 13106	0.079/ 3783	0.092/ 3824	0.150/ 5891	0.288/ 11156	0.071/ 2771	0.144/ 6320	0.416/ 16499	0.361/ 15095	0.067/ 2927	0.199/ 8137.2
		3.045/ 55099	3.219/ 58312	2.437/ 44389	1.005/ 17614	3.548/ 64381	3.742/ 68427	1.469/ 25944	1.428/ 25468	3.744/ 67713	3.432/ 62834	2.707/ 49018.1
	60	13.976/ 163381	9.529/ 11044	2.782/ 45772	8.422/ 98308	5.705/ 65473	10.380/ 121591	8.240/ 95522	5.469/ 62687	4.251/ 51742	3.164/ 48558	7.192/ 86407.8
	100											

```
struct Individual2
{
    int queen[N]; // 每列皇后所在行号
    int attacks_of_queen[N]; // 每列皇后在该个体对应格局下的攻击数
    int total_attacks; // 该个体对应格局下的总攻击数
};
```

运用上面的启发式规则,新的变异算子如下:

- (1) 从 N 个列号中随机抽样 $\log_2 N$ 个列号;
- (2) 从这 $\log_2 N$ 个列上,选择攻击数最大的皇后实施变异操作,即为这个皇后随机选择一个新的行号。

为了便于描述 N 皇后问题的学习算子,我们定义两个记号:

$bad_queens = \{i | 0 \leq i < N \wedge attacks_of_queen[i] \neq 0\}$, 即一个格局中所有对其它皇后造成攻击的皇后的列号集合。个体的学习过程就是要尽可能减小 bad_queens 集合来提高自身的适应值。

$safe_positions(i) = \{r | 0 \leq r < N \wedge \forall j (0 < j < N \wedge j \neq i \wedge queen[j] \neq r \wedge |queen[j] - r| \neq |j - i|)\}$, 即一个格局中第 i 列上所有没有受到来自其它各列上皇后攻击的行号集合。

给定一个个体,学习算子定义如下,基本思路是尽可能将每个皇后调整到不受攻击的地方。

- (1) 临时变量 tmp 集合的初值为空;
- (2) while(bad_queens 集合不为空){
- (3) 从 bad_queens 集合中随机选取一个列号 i , 并把 i 从 bad_queens 集合中删除;
- (4) if($safe_positions(i)$ 集合不为空{
- (5) 从 $(safe_positions(i))$ 集合中随机选取一个行号 j ;

同一行、同一向左倾斜的对角线(简称左对角线)或同一向右倾斜的对角线(简称右对角线)上。为了计算每个皇后的攻击数,只需要统计这个皇后所在的行、左对角线和右对角线上的皇后数目即可。为此,引入三个数组, $r[N]$, $ld[2N-1]$ 和 $rd[2N-1]$ 分别为 N 行、 $2N-1$ 条左对角线和 $2N-1$ 条右对角线上的皇后数目计数。第 i 列上的皇后出现在第 $j = queen[i]$ 行、第 $i+j$ 条左对角线、第 $i-j+(N-1)$ 条右对角线上。这样,两趟遍历所有皇后就可以求出总攻击数,得到一个时间复杂度为 $O(N)$ 的评价函数^[3]。

我们用 C++ 实现了这个算法,在 Pentium III 500MHz CPU, 128M 内存的 PC 机上进行了实验,每个例子计算 10 次,结果如表 1 所示。由此可见,一般演化算法求解 N 皇后问题的效果是不能令人满意的。

3.2 新算法的 N 皇后问题求解

一个显而易见的启发式规则是:攻击数较大的皇后造成了较大的破坏,是一个个体中较差的基因,算法应优先考虑调整攻击数较大的皇后的位置。我们用皇后的攻击数作为基因的适应值,在个体的表示中增加了一个数组用来保存当前格局中每个皇后的攻击数。新的表示方式为:

- (6) 在 bad_queens 集合中找出攻击 i 皇后的那些皇后,把她们的攻击计数减 1, 如果某个皇后的攻击计数减为 0, 则把她从 bad_queens 集合中删除;
- (7) 在 tmp 集合中找出攻击 i 皇后的那些皇后,把她们的攻击计数减 1, 如果某个皇后的攻击计数减为 0, 则把她从 tmp 集合中删除;
- (8) 调整总攻击数, 把 i 皇后的攻击数设为 0, 并把她的行号设为 j ;
- (9) }
- (10) else{
- (11) 把 i 加入 tmp 集合中;
- (12) }
- (13)}

由于计算 $safe_positions(i)$ 集合的时间复杂度为 $O(N)$, 而 bad_queens 集合在最坏情况下大小为 N , 因此这个算子的时间复杂度为 $O(N^2)$ 。

我们再用同样的环境下用新的算法进行实验,结果如表 2 所示。效率有明显的提高。

3.3 比较分析

图 2 和图 3 分别展示了一般演化算法和新算法在求解 N 皇后问题时的收敛状况。一般演化算法运行代数较大,而且在演化的后期阶段,搜索效率明显下降,特别是当皇后数目较大的时候。比如在求解 100 皇后问题时,如图 2 所示,实际上后 500 多代的演化只是在找最后一对相互攻击皇后的正确位置,占整个运行代数的一半以上。而对于新算法,如图 3 所示,运行代数显著下降,并且没有出现停滞的现象。这充分说明新的算法比一般演化算法有更快的收敛速度和更强的局部搜索能力。

表2 新算法的N皇后问题求解,数据格式:演化时间(秒)/评价函数运行次数

		实验编号										平均值
		1	2	3	4	5	6	7	8	9	10	
皇后数目	100	0.011/ 95	0.024/ 195	0.009/ 55	0.010/ 45	0.014/ 115	0.013/ 70	0.009/ 45	0.006/ 50	0.009/ 65	0.008/ 200	0.011/ 93.5
	1000	0.122/ 75	0.235/ 290	0.151/ 135	0.390/ 590	0.197/ 220	0.198/ 235	0.181/ 165	0.210/ 190	0.152/ 175	0.265/ 335	0.210/ 241
	10000	9.822/ 195	9.322/ 105	9.342/ 125	9.767/ 85	10.277/ 220	10.619/ 155	10.527/ 155	10.596/ 290	12.077/ 210	9.300/ 145	10.165/ 168.5

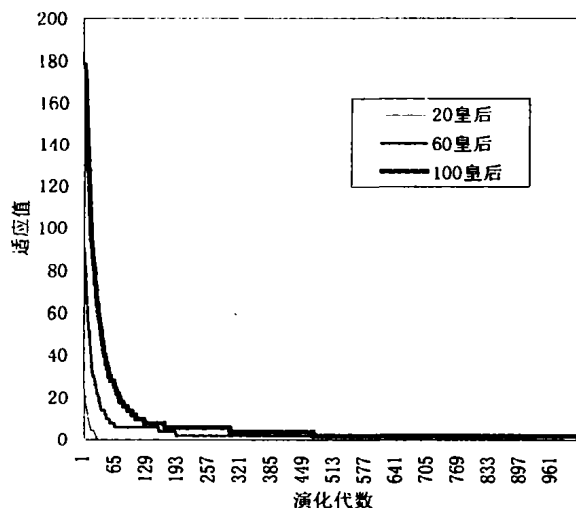


图2 一般演化算法求解N皇后问题的收敛曲线

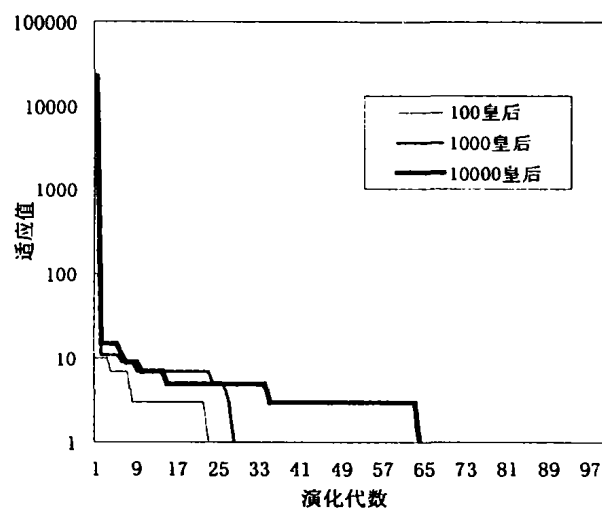


图3 新算法求解N皇后问题的收敛曲线

参考文献

- Holland J H. Adaptation in Natural Artificial System. The University of Michigan Press, 1975
- 潘正君, 康立山, 陈毓屏. 演化计算. 清华大学出版社. 广西科技

术出版社, 1998

- Sosic R, Gu J. Efficient Local Search with Conflict Minimization: A Case Study of the N-Queens Problem. IEEE Transactions on Knowledge and Data Engineering, 1994, 6(5): 661~668

(上接第58页)

发展是在电子商务上,这是因为WUM现在主要应用于电子商务领域,挖掘网上用户信息,而XML在电子商务上的应用也逐渐展开。例如在个性化网站方面,WUM根据日志、Cookie等信息,利用聚类分析,发现喜好近似的用户类和同时被访问的网页类,从而根据用户特征设计用户感兴趣的网页内容和结构,而使用XML通过中间层技术则可以方便地将设计好的网页内容实时地、动态地展现给用户^[13]。

结束语 当前数据挖掘和XML研究正方兴未艾,在今后的若干年将形成更大的高潮,由于Internet用户迅速增加,为了快速高效地找到网上的知识,研究在Web上的数据挖掘以及XML格式化,加强对非结构化数据如文本数据、图形图像数据、多媒体数据的挖掘,将是近期数据挖掘和XML相结合的重要课题。

参考文献

- Extensible Markup Language. <http://www.w3.org/XML/>
- Guillaume D, Murtagh F. Clustering of XML documents. Computer Physics Communications, 2000, 127: 215~227
- Agrawal R, Imielinski T, Swami A. Database Mining: A Performance Perspective. IEEE Transactions on Knowledge and Data Engineering, Special issue on learning and Discovery in Knowledge-Based Databases, 1993, 5(6): 914~925

- Han Jiawei, Kamber M. Data Mining: Concepts and Techniques. 机械工业出版社, 2001
- Grossman R, et al. The Management and Mining of Multiple Predictive Models Using the Predictive Modeling Markup Language. Information and Software Technology, 1999, 41: 589~895
- XML Miner. www.metadatamining.com
- Etainoni O. The World Wide Web: Quagmire or Gold Mine. Communications of the ACM, 1996, 39(11): 65~68
- Madria S K, et al. Research issues in Web data mining. In: Proc. of Data Warehousing and Knowledge Discovery, First Intl. Conf. DaWaK '99, 1999. 303~312
- Lang K, Burnett M. XML, Metadata and Efficient Knowledge Discovery. Knowledge-Based Systems, 2000, 13: 321~331
- Madria S K, et al. Research issues in web data mining. In: Proc. of Data Warehousing and Knowledge Discovery, First Intl. Conf. DaWaK'99, 1999. 301~312
- Cooley R, Mobasher B, Srivastava J. Web Mining: Information and Pattern Discovery on the World Wide Web (A Survey Paper) (1997). In: Proc. of the 9th IEEE Intl. Conf. on Tools with Artificial Intelligence (ICTAI'97), Nov. 1997
- Srivastava J, et al. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data (2000). SIGKDD Explorations, 2000, 1(2)
- Goldfarb C F, Prescod P. XML用户手册. 人民邮电出版社, 2000