

遗传算法的特点及应用领域研究

杜文丽^{1,2} 原 亮²

(1.河北交通职业技术学院 河北 石家庄 050051; 2.军械工程学院 河北 石家庄 050003)

【摘 要】遗传算法是一种基于概率意义的随机搜索算法,它的思想是构造一个问题的解的初代种群,经过选择,交叉和变异产生新的最优解种群。遗传算法的特点具有自组织、自适应和自学习性,遗传算法提供了一种求解复杂系统优化问题的通用框架,在工程设计、演化硬件电路设计以及人工智能等方面应用前景广阔。

【关键词】遗传算法;最优解;种群

1.遗传算法 (Genetic Algorithms,GA) 研究的历史比较短,20 世纪 60 年代末期到 70 年代初期,主要由美国国家 Michigan 大学的 John Holland 与其同事,学生们研究形成了一个较完整的理论和方法,遗传算法作为具有系统优化、适应和学习的高性能计算和建模方法的研究渐趋成熟。

遗传算法的基本思想,遗传算法是从代表问题可能潜在解集的一个种群 (population) 开始的,而一个种群则由经过基因 (gene) 编码 (coding) 的一定数目的个体 (individual) 组成。每个个体实际上是染色体 (chromosome) 带有特征的实体。染色体作为遗传物质的主要载体,即多个基因的集合,其内部表现是某种基因组合,它决定了个体的形状的外部表现。初代种群产生之后,按照适者生存和优胜劣汰的原理,逐代 (generation) 演化产生出越来越好的近似解。在每一代,根据问题域中个体的适应度 (fitness) 大小挑选 (selection) 个体,并借助于自然遗传学的遗传算子 (genetic operators) 进行组合交叉 (crossover) 和变异 (mutation),产生出代表新的解集的种群。这个过程将导致种群像自然进化一样的后生代种群比前代更加适应于环境,末代种群中的最优个体经过解码 (decoding),可以作为问题近似最优解。

2.遗传算法的特点

传统的优化方法主要有三种:枚举法、启发式算法和搜索算法:

(1)枚举法 可行解集合内的所有可行解,以求出精确最优解。对于连续函数,该方法要求先对其进行离散化处理,这样就可能因离散处理而永远达不到最优解。此外,当枚举空间比较大时,该方法的求解效率比较低,有时甚至在目前先进计算机工具上无法求解。

(2)启发式算法 寻求一种能产生可行解的启发式规则,以找到一个最优解或近似最优解。该方法的求解效率比较高,但对每一个需求解的问题必须找出其特有的启发式规则,这个启发式规则一般无通用性,不适合于其他问题。

(3)搜索算法 寻求一种搜索算法,该算法在可行解集合的一个子集内进行搜索操作,以找到问题的最优解或者近似最优解。该方法虽然保证不一定能够得以问题的最优解,但若适当地利用一些启发知识,就可在近似解的质量和效率上达到一种较好的平衡。

遗传算法不同于传统的搜索和优化方法。主要区别在于:

自组织、自适应和自学习性 (智能性)。应用遗传算法求解问题时,在编码方案、适应度函数及遗传算子确定后,算法将利用进化过程中获得的信息自行组织搜索。由于基于自然的选择策略“适者生存、不适者被淘汰”,因而适应度大的个体具有较高的生存概率。通常适应度大的个体具有更适应环境的基因结构,再通过基因重组和基因突变等遗传操作,就可能产生更适应环境的后代。进化算法的这种自组织、自适应特征,使它同时具有能根据环境变化来自动发现环境的特性和规律的能力。自然选择消除了算法设计过程中的一个最大障碍,即需要事先描述问题的全部特点,并要说明针对问题的不同特点算法应采取的措施。因此,利用遗传算法,我们可以解决那些复杂的非结构化问题。

遗传算法的本质并行性。遗传算法按并行方式搜索一个种群数目的点,而不是单点。它的并行性表现在两个方面,一是遗传算法是内在并行的 (inherent parallelism),即遗传算法本身非常适合大规模并行。最简单的并行方式是让几百甚至数千台计算机各自进行独立种群的演化计算,运行过程中甚至不进行任何通信 (独立的种群之间若有少量的通信一般会带来更好的结果),等到运算结束时才通信比较,选取最佳个体。这种并行处理方式对并行系统结构没有什么限制和要求,可以说,遗传算法适合在目前所有的并行机或分布式系统上进行

并行处理,而且对并行效率没有太大影响。二是遗传算法的隐含并行性 (implicit parallelism)。由于遗传算法采用种群的方式组织搜索,因而可同时搜索解空间内的多个区域,并相互交流信息。使用这种搜索方式,虽然每次只执行与种群规模 N 成比例的运算,但实质上已进行了大约 $O(N^3)$ 次有效搜索,这就使遗传算法能以较少的计算获得较大的收益。

遗传算法不需要求导或其他辅助知识,而只需要影响搜索方向的目标函数和相应的适应度函数。

遗传算法强调概率转换规则,而不是确定的转换规则。

遗传算法可以更加直接的应用

遗传算法对给定问题,可以产生许多的潜在解,最终选择可以由使用者确定。在某些特殊情况下,如多目标优化问题不止一个解存在,有一组 pareto 最优解。这种遗传算法对于确认可替代解集而言是特别合适的。

3.遗传算法的应用领域

遗传算法提供了一种求解复杂系统优化问题的通用框架,它不依赖于问题的具体领域,对问题的种类有很强的鲁棒性即健壮性,所以广泛应用于很多学科。下是遗传算法的一些主要应用领域:

(1)函数优化 函数优化是遗传算法的经典应用领域,也是对遗传算法进行性能评价的常用算例。很多人构造出了各种各样的复杂形式的测试函数,有连续函数好有离散函数,有凸函数也有凹函数,人们用这些几何特性各异的函数来评价遗传算法的性能。而对于一些非线性、多模型、多目标的函数优化问题,用其他优化方法较难求解,遗传算法却可以方便地得到较好的结果。

(2)组合优化 随着问题规模的扩大,组合优化问题的搜索空间急剧扩大,有时在目前的计算机上用枚举法很难或者甚至不可能得到其精确最优解。对于这类复杂问题,人们已意识到应把精力放在寻求其满意解上,而遗传算法则是寻求这种满意解的最佳工具之一。实践证明,遗传算法对于组合优化中的 NP 完全问题非常有效。例如,遗传算法已经在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面得到成功的应用。

(3)生产调度问题 生产调度问题在许多情况下所建立起来的数学模型难以精确求解,即使经过一些简化之后可以进行求解,也会因简化太多而使得求解结果与实际甚远。因此,目前在现实生产中主要靠一些经验进行调度。遗传算法已成为解决复杂高度问题的有效工具,在单件生产车间调度、流水线生产车间调度、生产规划、任务分配等方面遗传算法都得到了有效的应用。

(4)自动控制 在自动控制领域中许多与优化相关的问题需要求解,遗传算法的应用日益显示了良好的效果。例如用遗传算法进行航空控制系统的优化、基于遗传算法的模糊控制器优化设计、基于遗传算法的参数辨识、利用遗传算法进行人工神经网络的结构优化设计和权值学习都显示出了遗传算法在这些领域中应用的可能性。

(5)机器人智能控制 机器人是一类复杂的难以精确建模的人工系统,而遗传算法的起源就来自于对人工自适应系统的研究,所以机器人智能控制理所当然地成为遗传算法的一个重要应用领域。例如遗传算法已经在移动机器人路径规划、关节机器人运动轨迹规划、机器人逆运动学求解、细胞机器人的结构优化和行动协调等方面得到研究和应用。

(6)图像处理和模式识别 图像处理和模式识别是计算机视觉中的一个重要研究领域。在图像处理过程中,如扫描、特征提取、图像分割等不可避免地会产生一些误差这些误差会影响到图 (下转第 54 页)

G06:0950.39-
[Q]

YQ

+?

-

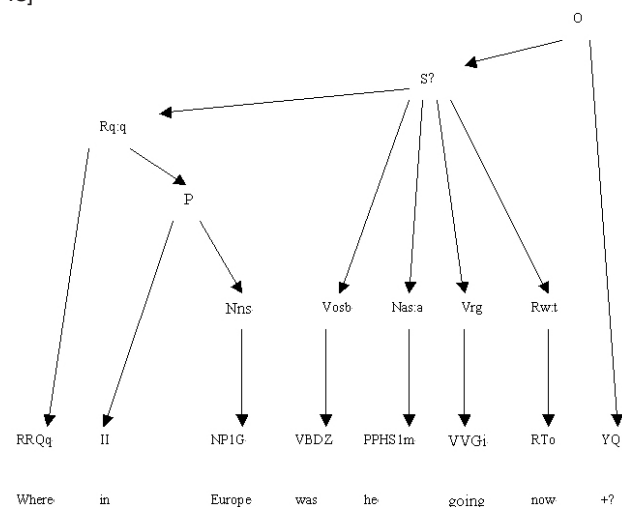


图 3-1

从 word 列还原出来的句子是 'Where in Europe was he going now?'. 从 parse 字段可以还原出这句话对应的分析 '[Q][S?][Rq:q RRQq [P II [Nns NP1g Nns][P]Rq:q [Vosb VBDZ Vosb] [Nas:s PPHS1m Nas:s] [Vrg VVGi Vrg] [Rwt RTo Rwt][S?] YQ Q]'. 这个分析对应的树如图 2.4 所示。根节点 O 和它的孩子结点 S?、YQ 组成规则 O → S? YQ。

用程序抽取规则的具体方法是，先从原始的语料库的 parse 列还原出句子的分析。然后在内存中建立如图 3-1 的树，最后用广度优先策略遍历树并把规则输出到另一个文件。生成树的算法如下：

```
while(还有符号没有读入)
{
    读入一个符号 c;
    switch (c)
    {
        case '[':
            // 为当前节点生成一个孩子结点;
            // 指针指向孩子结点;
            break;
        case ']':
            // 指针指向当前节点的父亲结点;
            break;
        else
            // 复制数据到当前结点;
            break;
    }
}
```

抽取规则的程序是一个广度优先遍历树的算法。树结点的数据结构采用孩子兄弟表示法。具体的数据结构和算法如下：

```
typedef struct node{
    char data[MAX]; //存储数据
    struct node *FirstChild; //指向第一个孩子结点
    struct node *Brother; //指向兄弟结点
}Tree;
ExtractRule(Tree *root)
{
    if(root != NULL)
    {
        // 输出当前结点及其孩子结点;
        ExtractRule(root->FirstChild);
    }
    else
        return;
}
```

初步整理出规则集后剔除重复的规则就得到项目用的规则集。规则集里总共有 34755 条不重复的规则。大多数规则出现的频率不高，经常出现的规则只占一小部分。见图 3-2

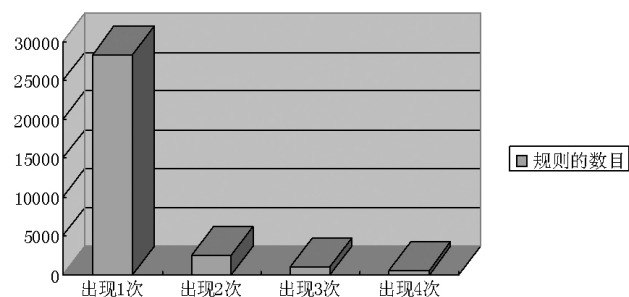


图 3-2

4. 结论

本文描述了在实际项目中，对 SUSANNE 语料库数据预处理的一个步骤。这种基于“还原重构”的处理方法确保了在抽取语法规则的时候不会出现信息的遗漏和丢失。规则抽取程序的算法复杂度和普通树遍历的算法复杂度一致^[4]，能在可接受的时间内完成规则抽取的任务。

参考文献

- [1] 俞士汶.《计算语言学概论》.北京:商务印书馆,2003.
- [2] Geoffrey Sampson. English for the Computer, Clarendon Press, 1995, 6.
- [3] Geoffrey Sampson. English for the Computer, Clarendon Press, 1995, 6.
- [4] 严蔚民,吴伟民.《数据结构》.北京:清华大学出版社,1997.

[责任编辑:韩铭]

(上接第 31 页)像处理和识别的效果。如何使这些误差最小是使视觉达到实用化的重要要求。遗传算法在图像处理中的优化计算方面是完全胜任的。目前已在图像恢复、图像边缘特征提取、几何形状识别等方面得到了应用。

(7) 人工生命 人工生命是用计算机等人工媒体模拟或构造出具有自然生物系统特有行为的人造系统。自组织能力和自学习能力是人工生命的两大主要特征。人工生命与遗传算法有着密切的关系，基于遗传算法的进化模型是研究人工生命现象的重要理论基础。虽然人工生命的研究尚处于启蒙阶段，但遗传算法已在其进化模型、学习模型、行为模型等方面显示了初步的应用能力。可以预见，遗传算法在人工生命及复杂自适应系统的模拟与设计、复杂系统实现性理论研究中，将得到更为深入的发展。

(8) 遗传程序设计 Koza 发展了遗传程序设计的概念，他使用了以 LISP 语言所表示的编码方法，基于对一种树型结构所进行的遗传

操作自动生成计算机程序。

(9) 机器学习 学习能力是高级自适应系统所应具备的能力之一。基于遗传算法的机器学习，特别是分类器系统，在许多领域得到了应用。例基于遗传算法的机器学习可用于调整人工神经网络的连接权，也可用于神经网络结构的优化设计。

参考文献

- [1] 徐清振,肖成林.遗传算法的研究与应用.现代计算机.2006.5.
- [2] 徐阳,王友仁.演化硬件理论与应用技术研究.航空电子技术 2003.3.
- [3] 周明,孙树栋.遗传算法原理及其应用.北京:国防出版社.1999.

作者简介:杜文丽(1967—),女,河北交通职业技术学院副教授,军械工程学院在读硕士,研究遗传算法改进及演化硬件应用。

[责任编辑:韩铭]