

# 双变异算子遗传算法的应用

鲁 群, 周爱武

(安徽大学 计算机学院, 安徽 合肥 230039;

安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

**摘 要:**针对简单遗传算法(SGA)所存在的缺点和不足,提出了一种新的改进遗传算法——双变异算子GA。该算想法通过将所有产生的子代个体与父代个体混合作为下一代种群,在种群选择前对适应度值较低的个体进行一次变异,然后通过选择、交叉,再一次变异产生新种群,再利用自适应算法改变交叉和变异率及最优保存策略保护历代最优个体,利用matlab软件编程计算,在TSP中得到了较好的优化结果。实例说明,双变异算子的遗传算法能够最大限度使种群多样性,这样最有可能得到最优解,也易突破局部收敛的局限而达到全局最优。

**关键词:**双变异算子;种群多样性;最优保存策略;遗传算法;自适应

**中图分类号:**TP18

**文献标识码:**A

**文章编号:**1673-629X(2008)07-0042-03

## Application of Genetic Algorithm Based on Dual Mutation

LU Qun, ZHOU Ai-wu

(School of Computer, Anhui University, Hefei 230039, China; Ministry of Education Key Lab. of Intelligent Computing and Signal Processing, Anhui University, Hefei 230039, China)

**Abstract:** Aims at SGA's weakness and shortage, putting forward a kind of new improvement GA, namely dual mutation GA. This viewpoint's passing to mix all output son generation individual and father generation individual as the next generation kinds, and lower individual to carry on a mutation before choosing in the kinds, then pass a choice, crossover, again the mutation to produce new kinds, making use of self-adaptation algorithm again, mutation and superior keep strategy protection in the past the superior individual. The case shows that this algorithm can rest rain it self quickly in TSP, and easily overcome the local restraint weakness and get the entire optimal result. Perfect outcome was obtained after programming with software Visual C++.

**Key words:** dual mutation; population diversity; superior keep strategy; genetic algorithm; self-adaptation

## 0 引言

生物在自然界中的生存繁衍,显示出了其对自然环境的自适应能力。受其启发,人们致力于对生物各种生存特性的机理研究和行为模拟,为人工自适应系统的设计和开发提供了广阔的前景。遗传算法(Genetic Algorithms,简称GAs)就是这种生物行为的计算机模拟中令人瞩目的重要成果。基于对生物遗传和进化过程的计算机模拟,遗传算法使得各种人工系统具有优良的自适应能力和优化能力。遗传算法所借鉴的生物学基础就是生物的遗传和进化。遗传算法(Genetic Algorithm, GA)<sup>[1-3]</sup>是近几年发展起来的一种崭

新的全局优化算法。1962年霍兰德(Holland)教授首次提出了GA算法的思想,它借用了仿真生物遗传学和自然选择机理,通过自然选择、遗传、变异等作用机制,实现各个个体的适应性的提高。从某种程度上说遗传算法是对生物进化过程进行的数学方式仿真。GA最为严重的就是“过早收敛”问题。由于种群规模是有限的,经复制、交叉、变异和按适应度进行选择,使得高于种群平均适应度的个体在下一代中得到更多的复制,这样不断进行,一旦某些个体在种群中占有绝对优势,GA则会强化这种优势,从而使搜索范围狭窄,产生“近亲繁殖”。针对此问题,文中提出双变异算子的遗传算法。

## 1 基本遗传算法(SGA)

遗传算法(GA)是模拟遗传选择和自然淘汰的生物进化过程的计算模型,它是一种全局优化算法,它借用了生物遗传学的观点,通过自然选择、遗传、变异等

收稿日期:2007-10-10

基金项目:安徽省信息产业发展专项基金(2006005)

作者简介:鲁 群(1973-),男,硕士研究生,研究方向为智能计算理论与应用、数据库设计;周爱武,副教授,硕士生导师,研究方向为智能计算与优化方法、数据库设计等。

作用机制<sup>[4]</sup>,实现各个个体的适应性的提高。

传统的简单遗传算法(SGA)概括起来包括以下几个步骤(如图1所示):

(1) 编码。GA 在进行搜索之前先将解空间的可行解数据表示成遗传空间的基因型串结构数据,这些串结构数据的不同组合便构成了不同的可行解。

(2) 初始群体的产生。产生初始种群的方法有两种:一种是用完全随机的方法产生的;另一种是根据某些先验知识再随机选取样本。遗传算法以该群体作为初始迭代点。

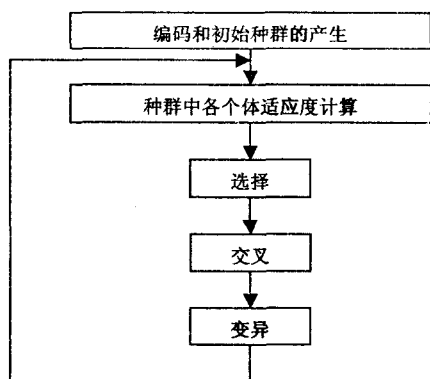


图1 遗传算法

(3) 适应度计算。根据实际标准计算个体的适应度,评判个体的优劣,即该个体代表的可行解的优劣。适应度函数的选择是实现算法关键的一步,因为它是GA寻求最优解的基础。

(4) 选择。从当前群体中选择  $M$  个优良(适应度高)的个体,使它们有机会被选中进入下一次迭代过程,舍弃适应度低的个体。

(5) 交叉。对选择产生的  $M$  个个体,按照事先设定的杂交概率任意选取两个进行杂交运算,或者称为交叉运算,产生新一代群体的两个新个体。杂交是两个个体交换一部分基因链的运算,是个体之间随机交换信息的一种手段。重复这一过程直到产生新一代的群体。

(6) 变异。在杂交运算产生的新的群体中,按照一定的概率从中选取若干个体,按一定的策略进行变异操作。变异运算增加了群体的多样性,避免了算法过早地陷入局部最优解。

(7) 检验停机条件是否满足,满足则停机;否则转到(3)继续进行进化过程。

## 2 双变异算子遗传算法

### 2.1 双变异遗传算法

在文献[5]中已经证明了,对于任意初始的无限种群,由选择算子按适应值重复进行选择,最终可以实现

使得全局寻优概率密度在评价函数的峰值处不断增强,并收敛到最大适应值个体点。在将传统的遗传算法应用于多峰值复杂函数的寻优时,发现存在着收敛速度慢,且存在陷入过早收敛而使结果与最优解相差较大的缺陷。其原因主要是传统遗传算法采用固定的交叉概率和变异概率,这样容易造成适应度高的个体过度早熟,使算法陷入局部最优。在文献[6]中也提到了双变异遗传算子。这里提出的主要改进为:

(1) 把上一代产生的新个体与父个体混合在一起,作为新一代的种群,即产生大种群。

(2) 在该算法中,设置了两个变异算子  $P_1, P_2$ ,以期望种群的多样性,其中  $P_1$  用于对种群选择之前,即在适应度计算之后就进行一次变异,此次变异主要是产生更多的个体,因此  $P_1$  应稍大( $P_1 > P_2$ )。

算法(如图2所示)具体操作:

① 编码和初始种群的产生。

② 判断是否第一代,若是,直接进行选择个体,作为下一代的种群;如果不是,则将变异产生的个体与父个体混合,构成等待选择的种群。

③ 计算种群中各个个体适应度。

④ 以一定概率对适应度低的进行大变异  $P_1$  变异。

⑤ 选择:从变异后的种群中选择  $N$  个个体,采用比例选择方法来进行选择。

⑥ 交叉和变异操作:采用自适应算法,对  $P_c$  和  $P_m$  进行调整。对于适应度高于群体平均适应度的个体,  $P_c$  与  $P_m$  取较小值,此解进入下一代;而低于平均适应度的个体,  $P_c$  与  $P_m$  取较大值,使该解被淘汰掉。故该方法可提供某个解的相对较佳的  $P_c$  与  $P_m$ ,在保持群体多样性的同时,保证了GA的收敛性。

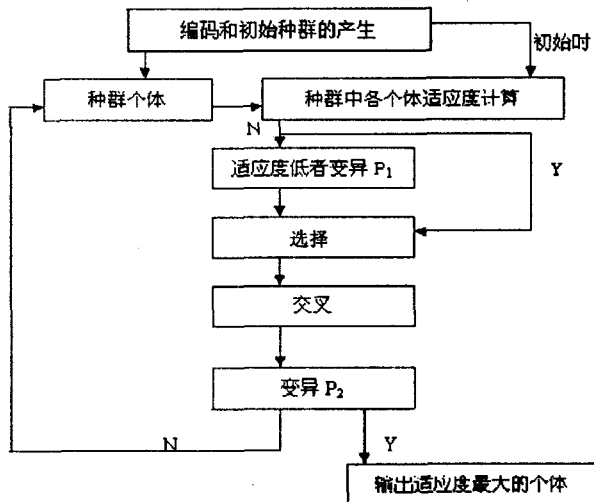


图2 流程图

⑦ 是否达到停机要求,是,则结束;否则进入②,

代数加 1。

## 2.2 可行解的转换

基本思想:在由个体基因型到个体表现型的变换中,增加使其满足约束条件的处理过程。即寻找出一种个体基因型和个体表现型之间的多对一的变换关系,使进化过程中所产生的个体总能够通过这个变换而转化成解空间中满足约束条件的一个可行解。这种处理方法虽然对个体的编码方法、交叉运算、变异运算等没有附加的要求,但它却是以扩大搜索空间为代价的,所以一般会使得遗传算法的运行效率有所下降。如图 3 所示。

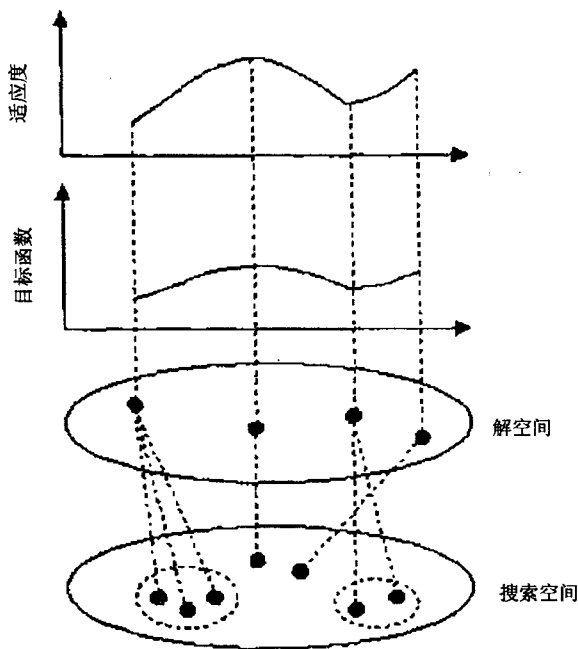


图 3 可行解的变化

## 3 最优保存策略

用历代最优个体替换掉当前群体中的最差个体,使迄今为止所得到的最优个体不会被选择、交叉、变异等遗传运算所破坏。其具体操作是<sup>[7]</sup>:

(1) 找出当前群体中适应度最高的个体和适应度最低及次低的个体。

(2) 若当前群体中最佳个体的适应度比历代最优个体的适应度高时,则复制当前群体中的最佳个体取代原先的最优个体而成为新的历代最优个体。

(3) 添加一个与 oldpop 种群的最优个体有较大的相异因子的个体,计算它的适应值,并将它与种群 newpop 的次劣个体进行比较,若比次劣个体的适应值小,则不替换种群 newpop 的次劣个体,反之则替换。

最优保存策略可视为选择操作的一部分。该策略的实施可保证迄今为止,所得到的最优个体不会被交叉、变异等遗传运算所破坏,它是遗传算法收敛性的一

个重要保证条件。

## 4 实验结果及分析

本算法通过 matlab 编程实现,设定当前种群大小为 50,迭代代数 200,变异概率  $P_1 = 0.5$ ,  $P_2 = 0.05$ , cities=30。由实验(如图 4、图 5 所示)可以看出:采用了双变异遗传算法的控制策略 SGA 在提高种群搜索范围上有明显提高,使解空间很好地脱离早熟,并且得到了已知的最优解。因此,基于双变异的遗传算法的寻优效率较简单遗传算法和自适应遗传算法性能有了较大提高,具有很强的鲁棒性、适应性、稳定性和种群的多样性。

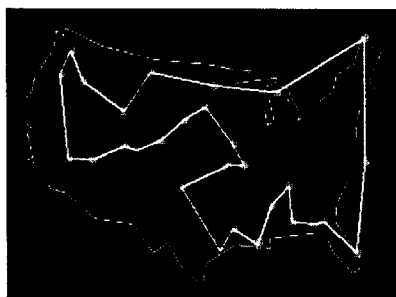


图 4 路线图 1

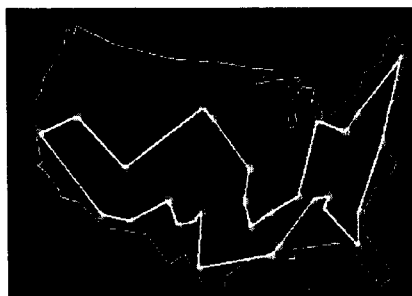


图 5 路线图 2

## 5 结束语

针对传统遗传算法的早熟收敛、得到的结果可能为非全局最优收敛解以及在进化后期搜索效率较低等缺点,提出了双变异算子的遗传算法。它利用种群选择前对适应度低的个体进行大概率变异,以期得到更多的个体,使种群多样性,并且将新的最优保存策略融入遗传算法,同时采用自适应算法改变交叉和变异概率。通过 matlab 的模拟实验说明,该算法收敛速度快,易突破局部收敛的局限而达到全局最优,得到了较好的优化结果。

### 参考文献:

- [1] Boseniuk T, Ebeling W. Boltzmann - Darwinn and Heackel - strategies in optimization problems[C]//Pro Int Conf on Par-

(下转第 52 页)

$6_y\} = F^+$ 。得证。

**算法 1 和算法 2 可终止性分析** 算法 1 重复运用提升规则, 算法 2 重复运用创建规则对给定的 DTDD 进行转换, 每转换一次,  $\sum$  中  $PFD_{XML}$  或  $TFD_{XML}$  的个数就减少一个, 由于  $\sum$  中  $PFD_{XML}$  或  $TFD_{XML}$  的数目是有限的, 所以算法 1 和算法 2 可以在有限的时间内终止。

**算法 1 和算法 2 的时间复杂度分析** 算法 1 和算法 2 的时间复杂度主要是由 FOR 循环体和合并消除冗余两步骤决定。FOR 循环体每执行一步, 消除  $\sum$  中一个  $PFD_{XML}$  或  $TFD_{XML}$ , 设最坏情况下  $\sum$  中所有  $FD_{XML}$  都是  $PFD_{XML}$  或  $TFD_{XML}$ , 令  $\sum$  中函数依赖数目  $n = |\sum|$ , 所以整个 FOR 循环体要执行  $n$  步。算法中的第 3 步合并和消除冗余元素类型是检查节点的冗余情况, 令  $D'$  中节点数目  $m = |D'|$ , 检查节点的冗余每次取 2 个元素, 要执行  $C_m^2$  步 ( $C_m^2 = m(m-1)/2 = m^2/2 - m/2$ ), 所以合并和消除冗余元素类型的时间复杂度最坏情形下为  $O(m^2)$ , 所以整个算法的时间复杂度为  $O(m^2 + n)$ 。

## 5 结束语

分析了 XML 文档中的部分函数依赖和传递函数依赖, 对规范相应函数依赖提出算法 1 和 2, 并对两个

算法为无损联接分解和依赖保持分解给出了证明, 对算法的可终止性和时间复杂度也进行了分析, 证明了这两个算法正确、可行。

## 参考文献:

- [1] Aitken P G. XML - the Microsoft Way[M]. [s. l.]: Addison Wesley, Inc. 2002.
- [2] Bray T, Paoli J, Sperberg - McQueen C M. Extensible Markup Language(XML)1.0[EB/OL]. 1998 - 02. <http://www.w3.org/TR/199811/REC-XML-19980210>.
- [3] Biron P V, Maihotra A. XML - Schema Part 2: Datatypes, W3C[EB/OL]. 2000 - 04. <http://www.w3.org/TR/xmlschema2>.
- [4] Arenas M, Libkin L. A Normal Form for XML Documents [C]//In: Proceedings of ACM Symposium on Principles of Database Systems (PODS). Madison, Wisconsin, USA: [s. n.], 2002: 85 - 96.
- [5] 施伯乐, 何继潮, 崔 靖. 关系数据库理论及应用[M]. 郑州: 河南科学技术出版社, 1989: 355 - 445.
- [6] 谈子敬, 施伯乐. DTD 的规范化[J]. 计算机研究与发展, 2004, 41(4): 594 - 600.
- [7] 张忠平, 王 超, 朱扬勇. 基于约束的 XML 文档规范化算法[J]. 计算机研究与发展, 2005, 42(5): 755 - 764.
- [8] 刘方鑫. 数据库原理与技术[M]. 北京: 电子工业出版社, 2002.

(上接第 44 页)

allel Solving from Nature. New York: Springer, 1990: 177 - 188.

- [2] 黄高义. 遗传算法实现平面连杆机构运动——轨迹综合 [D]. 上海: 上海交通大学机械与动力工程学院, 2003.
- [3] 潘 峰, 蔡维由. 改进遗传算法和神经网络在汽轮发电机组故障诊断中的应用[J]. 华中电力, 2004, 17(2): 1 - 4.
- [4] 金朝红, 吴汉松, 李腊梅, 等. 一种基于自适应遗传算法的神经网络学习算法[J]. 微计算机信息, 2005, 10(1): 49 - 51.

(上接第 47 页)

date generation[C]//In: Proc 2000 ACM - SIGMOD Int Conf Management of Data (SIGMOD'00). TX, Dallas: [s. n.], 2000.

- [2] Cheung D W, Han J, Ng V, et al. A fast distributed Algorithm for mining association rules[C]//In: Proc 1996 Int Conf Parallel and Distributed Information Systems. Miami Beach, FL: [s. n.], 1996: 31 - 44.
- [3] 铁治欣, 陈 奇, 俞瑞钊. 关联规则采掘综述[J]. 计算机应用研究, 2000, 17(1): 1 - 5.
- [4] Han Jiawei, Kamber M. 数据挖掘概念和技术[M]. 范 明,

- [5] Qi Xiaofeng. Theoretical analysis of evolution algorithms with an infinite population size in continuous space, Part 1, 2: basic properties of selection and mutation[J]. IEEE Tran. on Neural Networks, 1994, 5(1): 102 - 119, 120 - 129.
- [6] 方威云, 方千山, 王永初. 双变异率自适应遗传算法研究及其应用[J]. 南昌航空工业学院学报: 自然科学版, 2006, 16(2): 17 - 20.
- [7] 毕惟红, 任红民, 吴庆标. 一种新的遗传算法最优保存策略 [J]. 浙江大学学报: 理学版, 2006, 33(3): 32 - 35.

孟小峰, 等译. 北京: 机械工业出版社, 2001.

- [5] 陈文庆, 许 棠. 关联规则挖掘 Apriori 算法的改进与实现 [J]. 微机发展, 2005, 15(8): 155 - 157.
- [6] 胡吉明, 鲜学丰. 挖掘关联规则中 Apriori 算法的研究与改进[J]. 计算机技术与发展, 2006, 16(4): 99 - 102.
- [7] 朱孝宇, 王理冬, 汪光阳. 一种改进的 Apriori 挖掘关联规则算法[J]. 计算机技术与发展, 2006, 16(12): 89 - 91.
- [8] 何中胜, 刘宗田. 一种无候选集产生的并行关联规则挖掘算法[J]. 计算机工程与应用, 2004(24): 163 - 165.