

遗传算法

维基百科，自由的百科全书

遗传算法（英語：Genetic Algorithm，**GA**）是计算数学中用于解决最佳化的搜索算法，是进化算法的一种。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等等。

遗传算法通常实现方式为一种计算机模拟。对于一个最优化问题，一定数量的候选解（称为个体）可抽象表示为染色體，使种群向更好的解进化。传统上，解用二进制表示（即0和1的串），但也可以用其他表示方法。进化从完全随机个体的种群开始，之后一代一代发生。在每一代中评价整个种群的适应度，从当前种群中随机地选择多个个体（基于它们的适应度），通过自然选择和突变产生新的生命种群，该种群在算法的下一代迭代中成为当前种群。

目录

遗传算法的机理

算法

GA参数

特点

变量

适用的问题

发展历史

应用领域

相关技术

参见

参考文献

外部链接

遗传算法的机理

在遗传算法裡，优化问题的解被称为个体，它表示为一个变量序列，叫做染色体或者基因串。染色体一般被表达为简单的字符串或数字串，不过也有其他的依赖于特殊问题的表示方法适用，这一过程称为编码。首先，算法随机生成一定数量的个体，有时候操作者也可以干预这个随机产生过程，以提高初始种群的质量。在每一代中，都會评价每一个体，并通过计算适应度函数得到适应度数值。按照适应度排序种群个体，适应度高的在前面。这里的“高”是相对于初始的种群的低适应度而言。

下一步是产生下一代个体并组成种群。这个过程是通过选择和繁殖完成，其中繁殖包括交配（crossover，在算法研究领域我们称之为交叉操作）和突变（mutation）。选择则是根据新个体的适应度进行，但同时并不意味着完全以适应度高低为导向，因为单纯选择适应度高的个体将可能导致算法快速收敛到局部最优解而非全局最优解，我们称之为早熟。作为折中，遗传算法依据原则：适应度越高，被选择的机会越高，而适应度低的，被选择的机会就低。初始的数据可以通过这样的选择过程组成一个相对优化的群体。之后，被选择的个体进入交配过程。一般的遗传算法都有一个交配概率（又称为交叉概率），范围一般是0.6~1，这个交配概率反映两个被选中的个体进行交配的概率。例如，交配概率为0.8，则80%的“夫妻”会生育后代。每两个个体通过交配产生两个新个体，代替原来的“老”个体，而不交配的个体则保持不变。交配父母的染色体相互交换，从而产生两个新的染色体，第一个个体前半段是父亲的染色体，后半段是母亲的，第二个个体则正好相反。不过这里的半段并不是真正的一半，这个位置叫做交配点，也是随机产生的，可以是染色体的任意位置。再下一步是突变，通过突变产生新的“子”个体。一般遗传算法都有一个固定的突变常数（又称为变异概率），通常是0.1或者更小，这代表变异发生的概率。根据这个概率，新个体的染色体随机的突变，通常就是改变染色体的一个字节（0变到1，或者1变到0）。

经过这一系列的过程（选择、交配和突变），产生的新一代个体不同于初始的一代，并一代一代向增加整体适应度的方向发展，因为总是更常选择最好的个体产生下一代，而适应度低的个体逐渐被淘汰掉。这样的过程不断的重复：评价每个个体，计算适应度，两两交配，然后突变，产生第三代。周而复始，直到终止条件满足为止。一般终止条件有以下几种：

- 进化次数限制；
- 计算耗费的资源限制（例如计算时间、计算占用的内存等）；
- 一个个体已经满足最优值的条件，即最优值已经找到；
- 适应度已经达到饱和，继续进化不会产生适应度更好的个体；
- 人为干预；
- 以及以上两种或更多种的组合。

算法

- 选择初始生命种群
- 循环
 - 评价种群中的个体适应度
 - 以比例原则（分数高的挑中机率也较高）选择产生下一个种群（轮盘法（roulette wheel selection）、竞争法（tournament selection）及等级轮盘法（Rank Based Wheel Selection））。不仅仅挑分数最高的的原因是这么做可能收敛到局部的最佳点，而非整体的。
 - 改变该种群（交叉和变异）
- 直到停止循环的条件满足。

GA参数

- 种群规模（P, population size）：即种群中染色体个体的数目。
- 字串长度（l, string length）：个体中染色体的长度。

- 交配概率 (pc, probability of performing crossover) : 控制着交配算子的使用频率。交配操作可以加快收敛, 使解达到最有希望的最佳解区域, 因此一般取较大的交配概率, 但交配概率太高也可能导致过早收敛, 則稱為早熟。
- 突變概率 (pm, probability of mutation) : 控制着突變算子的使用频率。
- 中止条件 (termination criteria)

特点

遗传算法在解决优化问题过程中有如下特点:

- 遗传算法在适应度函数选择不当的情况下有可能收敛于局部最优, 而不能达到全局最优。
- 初始种群的数量很重要, 如果初始种群数量过多, 算法会占用大量系统资源; 如果初始种群数量过少, 算法很可能忽略掉最优解。
- 对于每个解, 一般根据实际情况进行编码, 这样有利于编写变异函数和适应度函数(Fitness Function)。
- 在编码过的遗传算法中, 每次变异的编码长度也影响到遗传算法的效率。如果变异代码长度过短, 变异的多样性会受到限制; 如果变异代码过长, 变异的效率会非常低下, 选择适当的变异长度是提高效率的关键。
- 变异率也是一个重要的参数。
- 对于动态数据, 用遗传算法求最优解比较困难, 因为染色体种群很可能过早地收敛, 而对以后变化了的数据不再产生变化。对于这个问题, 研究者提出了一些方法增加基因的多样性, 从而防止过早的收敛。其中一种是所谓**觸發式超级变异**, 就是当染色体群体的质量下降(彼此的区别减少)时增加变异概率; 另一种叫**随机外来染色体**, 是偶尔加入一些全新的随机生成的染色体个体, 从而增加染色体多样性。
- 选择过程很重要, 但交叉和变异的重要性存在争议。一种观点认为交叉比变异更重要, 因为变异仅仅是保证不丢失某些可能的解; 而另一种观点则认为交叉过程的作用只不过是种群中推广变异过程所造成的更新, 对于初期的种群来说, 交叉几乎等效于一个非常大的变异率, 而这么大的变异很可能影响进化过程。
- 遗传算法很快就能找到**良好**的解, 即使是在很复杂的解空间中。
- 遗传算法并不一定总是最好的优化策略, 优化问题要具体情况具体分析。所以在使用遗传算法的同时, 也可以尝试其他算法, 互相补充, 甚至根本不用遗传算法。
- 遗传算法不能解决那些“大海捞针”的问题, 所谓“大海捞针”问题就是没有一个确切的适应度函数表征个体好坏的问题, 使得算法的进化失去导向。
- 对于任何一个具体的优化问题, 调节遗传算法的参数可能会有利于更好更快收敛, 这些参数包括个体数目、交叉率和变异率。例如太大的变异率会导致丢失最优解, 而过小的变异率会导致算法过早的收敛于局部最优点。对于这些参数的选择, 现在还没有实用的上下限。
- 适应度函数对于算法的速度和效果也很重要。

变量

最简单的遗传算法将染色体表示为一个数位串，数值变量也可以表示成整数，或者实数（浮点数）。算法中的杂交和突变都是在字节串上进行的，所以所谓的整数或者实数表示也一定要转化为数位形式。例如一个变量的形式是实数，其范围是0~1，而要求的精度是0.001，那么可以用10个数位表示：0000000000表示0，1111111111表示1。那么0110001110就代表0.398。

在遗传算法里，精英选择是一种非常成功的产生新个体的策略，它是把最好的若干个个体作为精英直接带入下一代个体中，而不经任何改变。

通过并行计算实现遗传算法一般有两种，一种是所谓粗糙并行遗传算法，即一个计算单元包含一个种群；而另一种是所谓精细并行遗传算法，每一个计算单元处理一个染色体个体。

遗传算法有时候还引入其他变量，例如在实时优化问题中，可以在适应度函数中引入时间相关性和干扰。

适用的问题

遗传算法擅长解决的问题是全局最优化问题，例如，解决时间表安排问题就是它的一个特长，很多安排时间表的软件都使用遗传算法，遗传算法还经常被用于解决实际工程问题。

跟传统的爬山算法相比，遗传算法能够跳出局部最优而找到全局最优点。而且遗传算法允许使用非常复杂的适应度函数（或者叫做目标函数），并对变量的变化范围可以加以限制。而如果是传统的爬山算法，对变量范围进行限制意味着复杂的多的解决过程，这方面的介绍可以参看受限优化问题和非受限优化问题。

发展历史

遗传算法由密歇根大学的约翰·霍兰德和他的同事于二十世纪六十年代在对细胞自动机（英文：cellular automata）进行研究时率先提出。在二十世纪八十年代中期之前，对于遗传算法的研究还仅仅限于理论方面，直到在匹兹堡召开了第一届世界遗传算法大会。随着计算机计算能力的发展和实际应用需求的增多，遗传算法逐渐进入实际应用阶段。1989年，纽约时报作者约翰·马科夫写了一篇文章描述第一个商业用途的遗传算法--进化者（英文：Evolver）。之后，越来越多种类的遗传算法出现并被用于许多领域中，财富杂志500强企业中大多数都用它进行时间表安排、数据分析、未来趋势预测、预算、以及解决很多其他组合优化问题。

应用领域

- 计算机自动设计（CAD, Computer-Automated Design）
- 工业工程与运作管理
- 物流系统设计
- 生产调度
- 制造系统控制
- 系统优化设计
- 汽车设计，包括材料选择、多目标汽车组件设计、减轻重量等。
- 机电系统设计。



日本新幹線N700系列車“气动双翼”的獨特空氣動力造型車鼻；是遗传算法運算結果

- [分布计算机网络的拓扑结构](#)。
- [电路设计](#)，此类用途的遗传算法叫做[进化电路](#)。
- [电子游戏设计](#)，例如计算平衡解决方案。
- [机器智能设计和机器人学习](#)。
- [模糊控制系统的训练](#)。
- [移动通讯优化结构](#)。
- [时间表安排](#)，例如为一个大学安排不冲突的课程时间表。
- [旅行推销员问题](#)。
- [神经网络的训练](#)，也叫做[神经进化](#)。

相关技术

[遗传程序](#)是John Koza与遗传算法相关的一个技术，在遗传程序中，并不是参数优化，而是计算机程序优化。遗传程序一般采用树型结构表示计算机程序用于进化，而不是遗传算法中的列表或者数组。一般来说，遗传程序比遗传算法慢，但同时也可以解决一些遗传算法解决不了的问题。

[交互式遗传算法](#)是利用人工评价进行操作的遗传算法，一般用于适应度函数无法得到的情况，例如，对于图像、音乐、艺术的设计和“优化”，或者对运动员的训练等。

[模拟退火](#)是解决全局优化问题的另一个可能选择。它是通过一个解在搜索空间的随机变动寻找最优点的方法：如果某一阶段的随机变动增加适应度，则总是被接受，而降低适应度的随机变动根据一定的概率被有选择的接受。这个概率由当时的退火温度和适应度恶化的程度决定，而退火温度按一定速度降低。从模拟退火算法看，最优化问题的解是通过寻找最小能量点找到的，而不是寻找最佳适应点找到的。模拟退火也可以用于标准遗传算法里，只要把突变率随时间逐渐降低就可以了。

参见

- [演化策略](#)
- [遗传编程](#)
- [演化规划](#)
- [模拟退火](#)
- [禁忌搜索](#)
- [旅行推销员问题](#)
- [最优化](#)
- [交叉 \(遗传算法\)](#)
- [突變 \(遺傳演算法\)](#)

参考文献

- Goldberg, David E (1989), [遗传算法：搜索、优化和机器学习](#), Kluwer Academic Publishers, Boston, MA.

- Goldberg, David E (2002), 创新的设计：竞争遗传算法课程，Addison-Wesley, Reading, MA.
- Harvey, Inman (1992), 物种适应和遗传算法持续进行的基础 in 'Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life', F.J. Varela and P. Bourguine (eds.), MIT Press/Bradford Books, Cambridge, MA, pp. 346-354.
- Koza, John (1992), 遗传算法：通过自然选择编写计算机程序
- Michalewicz, Zbigniew (1999), 遗传算法+数据结构=进化程序，Springer-Verlag.
- Mitchell, Melanie, (1996), 遗传算法概论，MIT Press, Cambridge, MA.
- Poli, R., Langdon, W. B., McPhee, N. F. A Field Guide to Genetic Programming. Lulu.com, freely available from the internet. 2008. ISBN 978-1-4092-0073-4.
- Schmitt, Lothar M (2001), 遗传算法理论，Theoretical Computer Science (259), pp. 1-61
- Schmitt, Lothar M (2004), 遗传算法理论（二），Theoretical Computer Science (310), pp. 181-231
- Vose, Michael D (1999), 简单遗传算法：基础和理论，MIT Press, Cambridge, MA.

外部链接

- https://web.archive.org/web/20160507233728/http://userweb.eng.gla.ac.uk/yun.li/ga_demo/ - 格拉斯哥大学的在线交互式演示与学习GA_demo
 - <https://web.archive.org/web/20050903002901/http://cs.felk.cvut.cz/~xobitko/ga/> - 用Java语言编写的遗传算法在线介绍程序。
 - 伊利诺斯遗传算法实验室 (<http://www-illigal.ge.uiuc.edu/illiGAL>) - 可以下载技术报告和程序源代码。
 - [Global Optimization Algorithms - Theory and Application \(http://www.it-weise.de/projects/book.pdf\)](http://www.it-weise.de/projects/book.pdf) (页面存档备份 (<https://web.archive.org/web/20080911075107/http://www.it-weise.de/projects/book.pdf>), 存于互联网档案馆)
-

取自“<https://zh.wikipedia.org/w/index.php?title=遗传算法&oldid=70370586>”

本页面最后修订于2022年2月27日 (星期日) 08:50。

本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅使用条款）
Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。
维基媒体基金会是按美国国内稅收法501(c)(3)登记的非营利慈善机构。