

遗传算法理论与实现技术

(三)

遗传算法改进简介

- 尽管遗传算法在理论上具有概率1的收敛特性，但实际应用时往往出现早熟收敛或收敛缓慢等缺点。
- 因此，设计遗传算法的框架、操作和参数，主要应针对如何设法高效产生或有助于产生优良的个体成员，而这些成员应能够充分表征整个解空间的特性，从而提高算法的搜索效率，避免早熟收敛现象。
- 归纳而言，就是要综合考虑GA的探索(exploration)和开发(exploitation)能力，综合考虑种群的全局收敛(convergence)和分散多样性(diversity)。
- 现今的改进方法，除了采用自适应算法参数外，大都是针对基因操作、种群的宏观操作、基于知识的操作和并行化GA进行的。

复制操作

- De Jong(1975)研究了回放式随机采样复制，其缺点是选择误差较大；另外他又提出了无回放式随机采样复制以降低选择误差。
- Brindle(1981)提出了一种选择误差更小、操作简单确定式采样以及无回放式余数随机采样方法。
- Back(1992)提出了与适配值大小和正负无关的均匀排序策略；另外又提出全局收敛的最优串复制策略，提高了搜索效率，但不适于非线性较强的问题。
- 对于多目标优化，分享(sharing)操作，即对种群内个体的适配值或选择概率进行变换以减小相似个体的选择强度，从而减少相似个体的复制量，以尽可能地保持种群的多样性来达到同时搜索多个区域的目的。

交叉操作

- Goldberg(1989)提出了部分匹配交叉算子(PMX)。
- Starkweather等(1991)提出了增强边缘重组算子(Enhanced Edge Recombination)。
- Davis(1991)提出了序号交叉算子(Order Crossover)和均匀排序交叉算子(Uniform Order-based Crossover)。
- Smith(1985)提出了循环交叉算子(Cycle Crossover)。
- De Jong(1975)提出了单点交叉算子(One-point Crossover)和多点交叉算子。
- Syswerda(1989)提出了双点交叉算子(Two-point Crossover)。
- 另外，还有置换交叉、算术交叉、启发式交叉等。

变异操作

- 除前节介绍的常用变异算子外，变异操作主要发展了自适应变异、多级变异等操作。

高级基因操作

- 双倍体和显性遗传(Diploid and Dominance)用以延长曾经适配值高而目前较差的基因块的寿命，并且在变异概率低的情况下也能保持一定的多样性。
- 倒位操作(Inversion)用以助长有用基因块的紧密形式。
- 优先策略(Elitism)用以把目前解群中最好的解直接放入下一代种群中，如此保证各代种群中总会有目前为止最好的解。
- 静态繁殖(Steady-state Reproduction)用以在迭代过程中用部分优质子串来更新部分父串作为下一代种群以使优质的父串在下一代中得以保留；没有重串的静态繁殖(Steady-state Without Duplication)用以在形成下一代种群时不含重复的串。
- 多倍体结构(Multiple Chromosome Structure)、分离(Segregation)、异位(Translocation)等操作。
- 王凌等提出了多交叉和多变异操作的混合优化框架，以及基于模拟退火机制的变异操作。
- 在种群宏观操作方面，主要是引入小生存环境和物种形成(Niche and Spectiation)的思想，这在分类问题中得到了应用。
- 在基于知识的操作方面，主要是将问题的特殊信息与GA相结合，包括混合算法的构造以及在遗传算子中增加知识的操作等。

算法结构

- Krishnakumar(1989)为克服群体数目大造成计算时间长的缺点，提出了所谓 μ GA的小群体方法，仿真结果显示了较高的计算效率和适于动态系统优化的潜力，但尚无严格的理论分析。
- Schraudolph等(1992)针对二进制编码优化精度差的缺点，提出了一种类似于对解空间进行尺度变换的参数动态编码策略，较好地提高了GA的精度，但不适于非线性较强的多模型优化问题。
- Androulakis等(1991)采用实数编码提出了一种扩展遗传搜索算法，把搜索方向作为独立的变量来处理以解决有约束的优化问题。
- Poths等(1994)为克服算法早熟收敛的缺点，提出了类似于并行实现思想的基于变迁和人工选择的遗传算法。
- Grefenstette(1981)全面研究了GA并行化实现的结构问题，并给出了多种结构形式，主要有同步主-仆方法(Synchronous Master-slave)、亚同步主-仆方法(Semi-synchronous Master-slave)、分布式异步并发方法(Distributed Asynchronous Concurrent)、网络方法(Network)。
- Goldberg(1989)提出了基于对象设计GA并行结构的思想。
- Muhlenbein等(1991)用并行遗传算法求解高维多模型函数的全局最小解，从而提供了GA求解高度复杂优化问题的有力实例。
- 王凌等将GA和SA有机结合提出了GASA优化框架，有效解决了一系列调度问题和光学仪器的设计问题。相应地，还有GA与禁忌搜索、神经网络、混沌、拉氏松弛法、梯度法等的结合。

针对函数优化的改进

- Goldberg(1989)引入分享(Sharing)思想将解空间分成若干子空间，然后在子空间中产生子群体成员分别进行优化，以求得整个问题的解，避免算法只收敛到某个局部解。
- De Jong(1975)提出聚集(Crowding)的思想，根据群成员中的相似性来部分替换群体中的个体成员，从而将一些个体成员分别聚集于各个群集中，然后在各个群集中分别求解问题的局部解以实现与分享思想相同的目标。然而，这些方法的应用还有一定的限度，对于解是随机分布的情况就不易奏效。
- 孟庆春等(1995)提出门限变换思想，在选择个体成员进入下一代时，引入门限变换函数将某些优良成员周围的成员传到下一代以达到除劣增优的目的，从而避免搜索的盲目性。
- 对带约束优化问题，处理方法主要有：
 - 把问题的约束在个体的表示形式中体现出来，并设计专门的遗传算子，使个体所表示的解在GA运行过程中始终保持可行性。这种方法最直接，但适用领域有限，算子的设计也较困难。
 - 在编码过程中不考虑约束，而在GA运行过程中通过检验解的可行性来决定解的弃用。这种方法一般只适用于简单的约束问题
 - 采用惩罚的方法出来约束越界问题。

并行遗传算法

- GA的内在并行性在Holland提出遗传算法时就得到了认识，因此，在并行计算机上实现GA是提高算法性能和效率的有效途径。
- 并行GA
 - 同步主仆式
 - 异步并发式
 - 网络式
 - GAMAS

同步主仆式

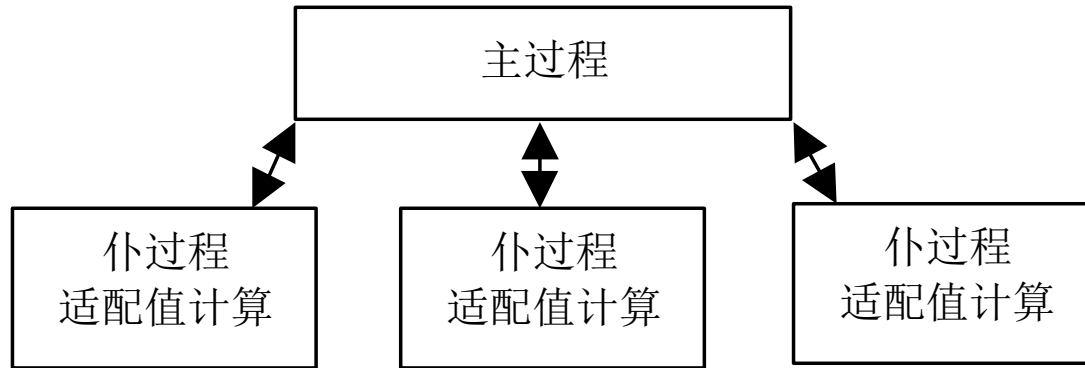


图 2.8.1 同步主仆式并行遗传算法

- 在这种并行方式中，一个主过程协调若干个仆过程。其中，主过程控制选择、交叉和变异的执行，仆过程仅执行适配值的计算。
- 这种并行化方式很直观，且易于实现。
- 其存在的两个主要缺点是：若各仆过程计算适配值的时间存在明显差异时，将会造成整个系统长时间的等待；整个系统可靠性较差，对主过程状况的依赖性较大。

异步并发式

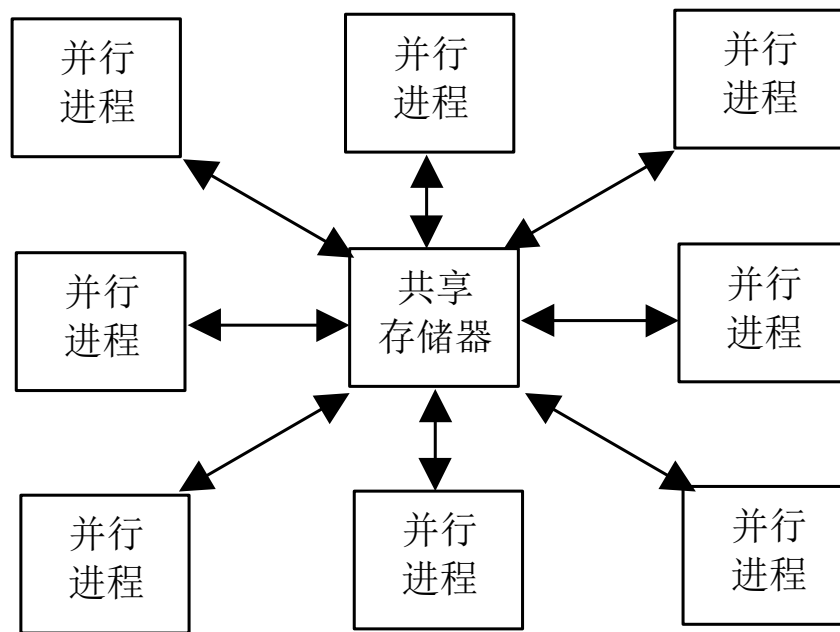


图 2.8.2 异步并发式并行遗传算法

- 在这种并行方式中，通过存取一个共享存储器，若干个同样的处理机彼此无关地执行各个遗传算子和适配值的计算。只要存在一个并行进程，同时共享存储器可继续运行，则整个系统就可进行有效的处理。
- 显然，这种方式不易实现，但可大大提高系统的可靠性。

网络式

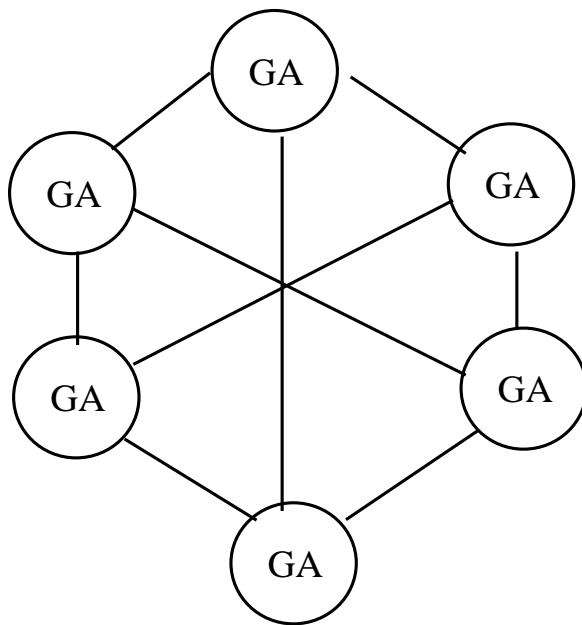


图 2.8.3 网络式并行遗传算法

- 在这种并行方式中，若干个无关的遗传算法分别在独立的存储器上进行独立的遗传操作和适配值计算，同时各个子群体在每一代中发现的最佳个体通过相应的通讯网络传送给其它子群体。
- 与前两种方式相比，由于存在通讯时的间断，此方式的连接带宽缩小了。但是，由于各独立过程的自治性，系统的可靠性提高了。

GAMAS模型

- 设计GA的目标则是快速得到高质量的解，但在GA运行过程中其全局收敛与种群多样性存在一定的矛盾，因此均衡整体探索(exploration)和局部开发(exploitation)能力对提高算法性能很关键。若增大选择压力并减小变异概率，则种群将有可能早熟收敛，反之则可能收敛速度缓慢。

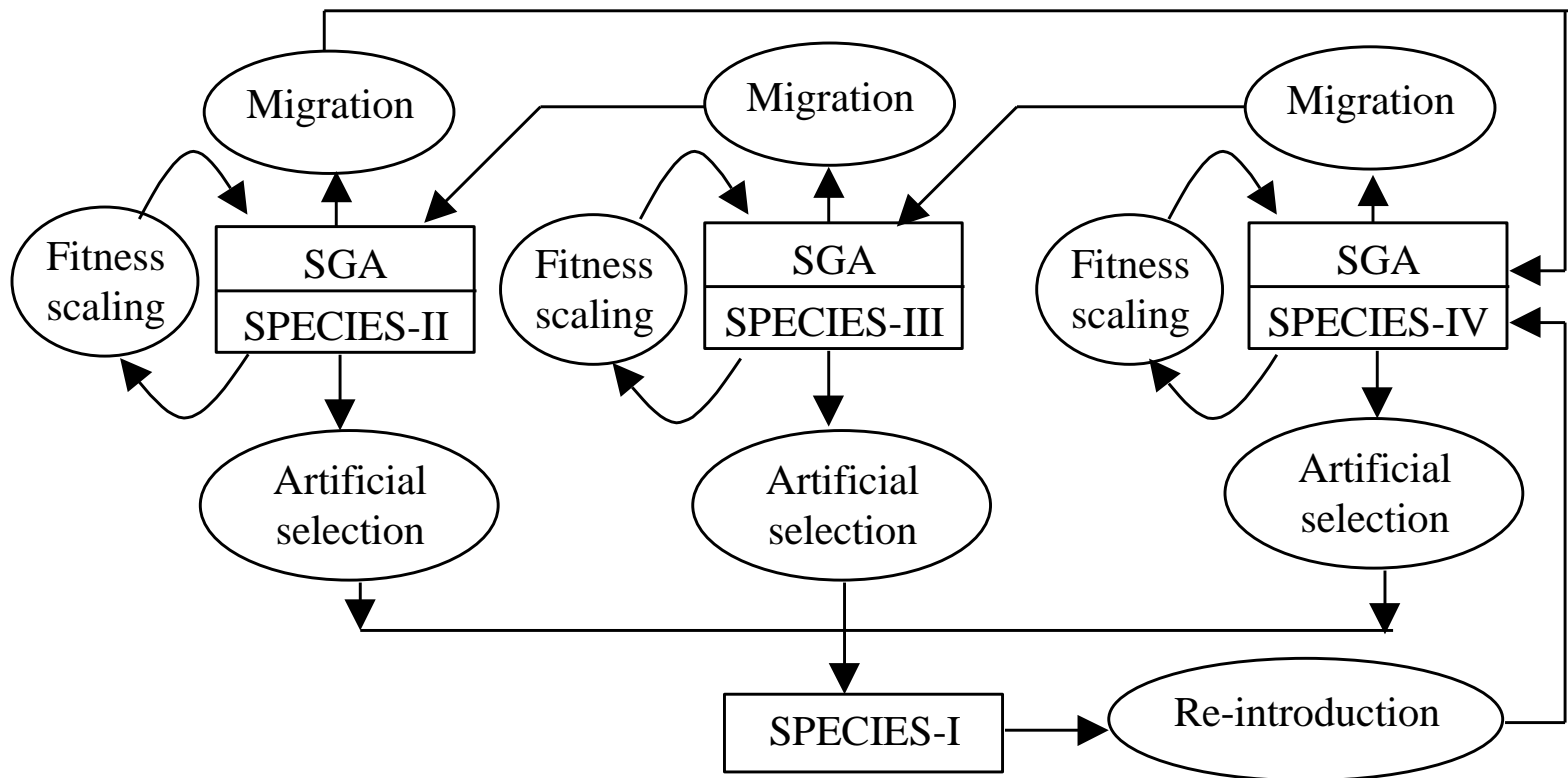


图 2.8.4 GAMAS 模型结构

GAMAS(续)

- GAMAS将种群分解成四个种群数目相同的子种群，分别记为SPECIES-I, SPECIES-II, SPECIES-III和SPECIES-IV，各子种群均有其搜索的倾向性。其中：
 - SPECIES-I作为一个子种群本身不进行遗传搜索过程，而是只收集所得到的较好个体再进行人工选择并分配给其他子种群。
 - SPECIES-II采用较高的变异概率(譬如0.05)，以使得搜索侧重于整体探索。
 - SPECIES-IV采用较低的变异概率(譬如0.003)，以使得搜索侧重于局部开发。
 - SPECIES-III则采用SPECIES-II和SPECIES-IV之间的变异概率(譬如0.005)。
- SPECIE-II、SPECIE-III和SPECIE-IV这三个子种群采用SGA结构进行微进化(micro evolution)，利用适配值尺度变换操作来解决适配值差异问题，并采用相同的交叉概率(譬如0.75)，当然也可不同。如果它们获得了比SPECIES-I中个体更好的个体，则用人工选择操作替换SPECIE-I中较差的个体，然后再用SPECIE-I中的个体替换SPECIE-IV中的所有个体，并且在SPECIE-II、SPECIE-III和SPECIE-IV之间实行个体迁移，整个过程称为宏进化(macro evolution)。
- 另外，GAMAS还采用重初始化策略来增强个体的多样性以防止早熟收敛。基于模式分类和函数优化的数值仿真验证了GAMAS的有效性，而且很显然GAMAS是一种并行遗传算法框架。

结语

- GA不只是一种单纯的优化算法，而是一种以生物进化思想为基础的一般方法论，是解决复杂问题的有力工具。
- GA不是传统的确定性计算工具，动态的复杂问题的求解也不可能是确定的，应建立新的评价标准，这在后文也将给予简单介绍。
- GA的理论正在深入，应用日趋广泛，但它仅是生物进化系统的简单近似模拟，其本身的发展也是不断进化的过程，理论研究需要引入新的数学工具并吸收生物学的最新成果，应用研究的成败依赖于对GA及其所解决问题的深刻理解。
- 随着GA理论的愈来愈完善，应用领域愈来愈拓宽，在人工生命、其它优化计算技术等领域与GA相结合后，GA将会发挥其更大的潜力。