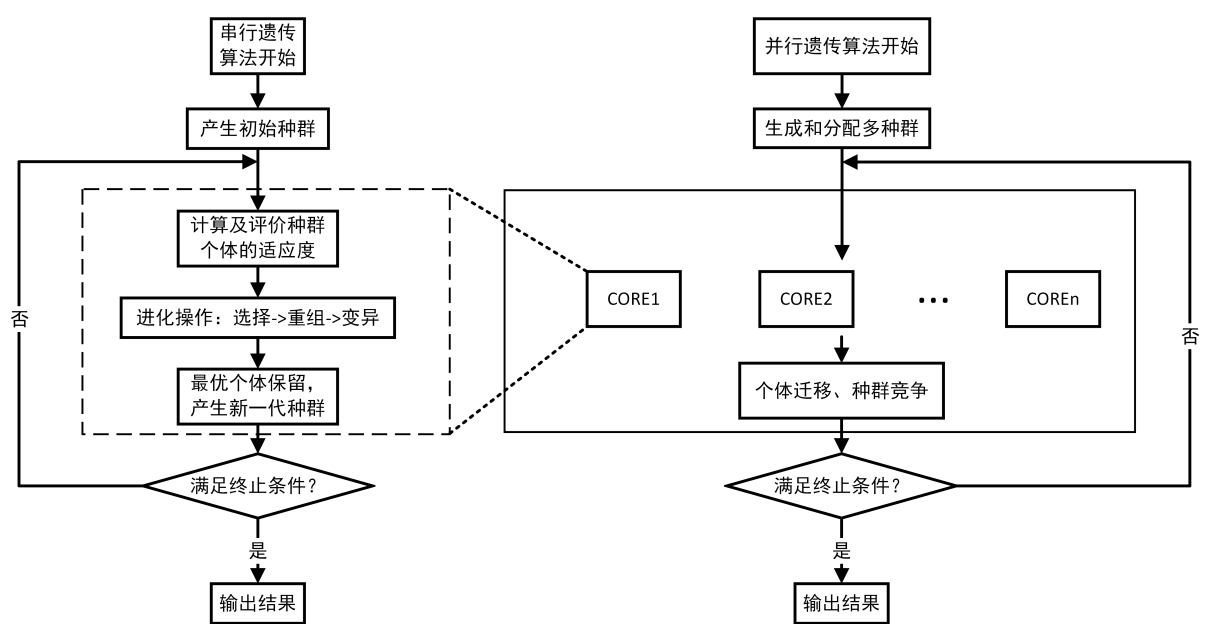


1 进化算法概述

自然界生物在周而复始的繁衍中，基因的重组、变异等，使其不断具有新的性状，以适应复杂多变的环境，从而实现进化。进化算法精简了这种复杂的过程而抽象出一套数学模型，用较为简单的编码方式来表现复杂的现象，并通过简化的遗传过程来实现对复杂搜索空间的启发式搜索，最终能够在较大的概率下找到全局最优解，同时天然地支持并行计算。

下图展示了常规遗传算法 (左侧) 和在并行计算下的遗传算法 (右侧) 的基本流程。



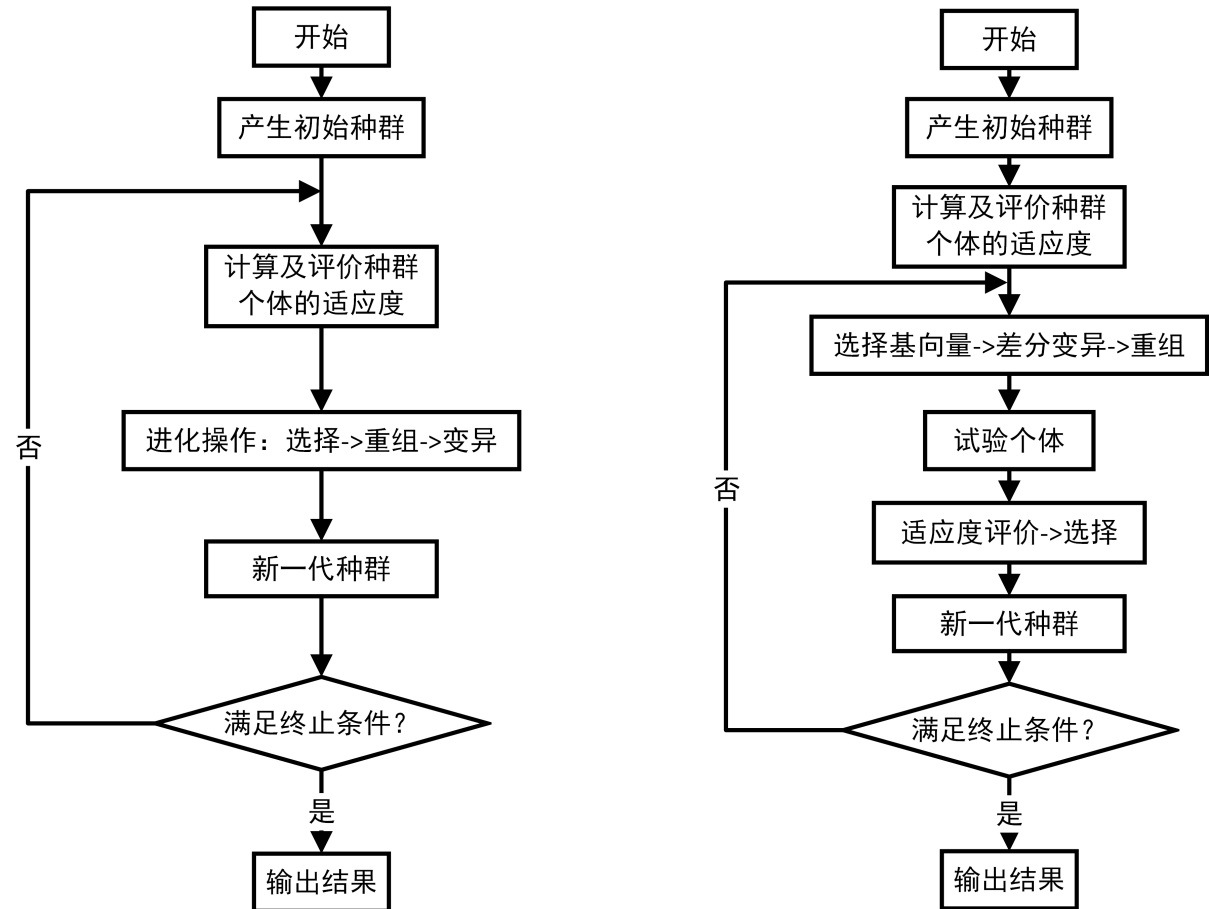
图中 $CORE_n$  表示计算核心。不同的计算核心处理不同的单个子种群 (当然也可以处理多个子种群)，种群间互相独立进行进化 (区域模型)，种群间进行个体迁移和种群竞争。这只是其中一种并行计算遗传算法，另外还有全局模型和本地模型。

除了种群间可以并行计算外，种群内的若干个个体之间也可以实现并行计算。因此遗传算法具有很强的并行性。

值得注意的是：在进化算法中，重组和交叉并不是同一个概念，交叉是重组的一种。

对于常规遗传算法，在计算开始时，根据设计的编码规则随机初始化许多个体 (形成一个或多个种群)，然后评估种群中个体的适应度，并根据适应度来选择一些个体到交配池，然后对交配池中的个体进行一定概率的重组和变异产生育种后代。至此，环境中同时存在父代和育种种群，此时需要从中选择出一些个体最终得到新一代种群。

在这个过程里面出现了两次选择：第一次是从当前种群中选择出一些个体参与重组和变异等进化操作；第二次是在父代和重组变异得到的育种后代中选择一些个体保留到下一代，这个阶段的选择也有时候被称为“重插入”或者“环境选择”。但很多时候这两次选择之中的某一次经常会在算法描述中被省略掉，这并不意味着只有一次选择，本质上依旧是存在两次选择，下图展示了经典遗传算法和经典差分进化算法的流程对比：



(a)经典遗传算法

(b)差分进化算法

左图是经典的遗传算法流程图，可以看到它里面只出现了一次选择。实际上，它在重组和变异得到育种个体之后，无条件地代替了父代个体而形成新一代种群，这本质上也是一次选择，只是在选择中把父代所有个体都淘汰掉了。这种做法也有一定的弊端：收敛速度较慢。因此有不少加入了“精英保留”的改进型遗传算法，比如把育种种群的绝大部分个体（小于全部）代替父代中等数量的非最优个体而得到新一代种群；另外还有把父代个体和育种个体合并，在统一的、相同的环境下进行择优选择一半个体得到新一代种群；差分进化里面的一种经典做法是按照个体的索引顺序，每个育种个体只与其相同位置上的父代个体进行优胜劣汰保留其中一个，从而得到新一代种群。

由图也可以看出，进化算法和传统的搜索和优化算法有着显著不同，最明显的差异是：

- 进化算法具有天然的并行性，它可以并行地搜索一组点，而不是一个点。
- 进化算法使用的是概率转换规则，并非确定性转换规则。
- 进化算法不需要额外的信息，只有目标函数和相应的适应度影响搜索方向。
- 进化算法鲁棒性强，可以与各种算法轻松地结合在一起。
- 进化算法可以整合其他优化算法的优点，比如利用其他优化算法的优化结果来生成初始种群，这种二次搜索方式在很多场合下可以大幅度提高搜索效率。
- 进化算法可以给特定的问题提供多样化的搜索结果，让用户自己选择。比如在多目标优化的进化算法里，算法给出的是一组帕累托最优解。这些最优解可以作为多组备选方案。

选择、重组和变异是进化算法提供的经典操作算子。很多改进的进化算法都是围绕他们展开的。或许在命名上可能十分多样化，但万变不离其宗，本质上也能够被归入到这几类操作算子当中。