**任务分配编码方式介绍**

在已知一段时间内有n个任务的情况下，生成一串包含（1~n）的列表作为一段任务分配的编码。这段编码可以决定任务分配的顺序和最终完成所有任务的总时间。以下三个算法本质上都是根据计算在有限时间内得出一个使得总时间相对最小的编码。

**模拟退火算法**

根据一段初始编码和任务的优先级以及先后次序计算出完成所有任务的总时间，作为系统当前总能量。

以一定的概率交换两个相邻任务的编码，并计算新的总能量。

根据当前温度（会随迭代次数而随之降低）、当前总能量和新的总能量来计算是否接收新的总能量来代替当前总能量。（当前温度越高、新的总能量越低（时间越短）越有可能接收）

重复上面的步骤，知道当前温度降低到某一阈值，最后的编码即认为是最终的任务编码。

**遗传算法**

与模拟退火只维护一个编码不同，遗传算法会维护一个种群，种群中的每个个体都是一条编码。

首先会初始化一个种群，内部的个体均为随机的编码，然后对这个种群依次按照下面三个步骤进行迭代：

1. 选择

根据种群中个体的运行时间，选择最优的个体复制多次加入新的种群，在剩余个体中采用轮盘赌的方式进行选择加入新种群中以满足新种群和旧种群的个体数目一致。

1. 交叉

从新种群中选择两个个体进行交叉操作，生成两个新的个体用以代替原先的个体。重复上述多次。

1. 变异

从新种群中选择一个个体进行编码位置上的交换，重复上述多次。

经过一定的迭代次数后，在最终的种群中选择最好的个体（花费时间最短的个体）作为最终的任务编码。

**蚁群算法**

与遗传算法中的种群类似，蚁群算法需要维护一个蚁群。蚁群中的每只蚂蚁都有一张列表，代表这只蚂蚁依次走过的任务编号（一张完整的列表就是一个任务编码）。而每个任务编号之间都有一个值，我们称为信息素（两个任务之间的信息素越多蚂蚁就越可能走这条路径）。蚂蚁每从一个任务走到另一个任务，这两个任务之间的信息素就会增加（总时间越短，增加的越多）。

初始条件下每个蚂蚁的列表都为空，任务编号之间的信息素也都相同。每只蚂蚁依据信息素的大小采用轮盘赌的方式补完自己的列表。在所有蚂蚁都补完列表后，更新所有的信息素，同时蒸发相应比例的信息素。

循环上述操作多次以后，选择蚁群中最优的蚂蚁（即它的列表所代表的任务编码花费时间最短）的列表作为最终的任务编码。