对于问题一，算法时间复杂度为

任务是包含了洒水作业时间，路程时间

路程时间= 前去洒水点的时间+回补给点时间

为的是任务时间最小。

决策变量



若它是解，则为1，否则为0(我考虑了一下，不应该写成，因为这是决策变量，具体来说，是这个意思

如果以变量集合 1代表一种可行方案的话，那么就记



所以真正关心的不是变量X的值，而是这个变量X为1时候的具体方案，即ijk)。

因此我们即是要求解

对于满足约束条件，且使得最远时间最小（目标函数）的，所有的可行方案。

强调这句话的意思是，关注的是方案，而不是X本身。

为了方便，洒水车的出发点跟编号大小有关系，1-10在D1

为了记号方便，记表示决策变量中出现的洒水车，也就是这个决策中待执行洒水任务的洒水车。

那么初始时，停靠在两个站点的洒水车如下



使用以下获取车型



也就是说表示第

用



获取每辆洒水车的出发点

用



这个获取每辆洒水车的洒水作业时间 。

为了方便，也常记为

待优化的目标函数是 ，为使得最长洒水任务的那个洒水车的任务时间最小化，即



应该满足以下约束：

每辆洒水车只能去喷洒点执行喷洒作业一次。



每个喷洒点只能被喷洒一次。



每个补给点不能补给超过8次



额外的约束，最长任务时间肯定超过每辆车本身大于等于自身执行任务的时间。当它是最长的执行任务时间时取等。



写成数学表达式就是：





其中表示，在部分决策i确定的（即此时i为已知），而其他决策j,k的所有情况下的洒水时间。所以这是一个集合。

后面问题可能的分步决策



有理由说明。（降低问题规模），聚类的思想



问题二

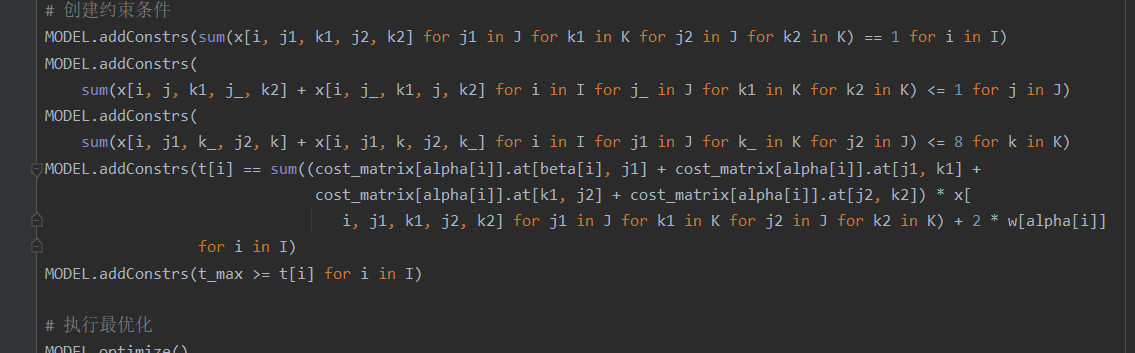
这是未改进的

如果以表示的最优解 ，以表示 ，这好比优化



那么原优化目标为





，从for的层数可以看出，算法的时间复杂度是



尽管线性规划有成熟的理论求解，但这显然不是一个好的优化目标。而且从题中可以看出，可选的喷洒作业点是很多的，是非常大的。那么以上算法的时间复杂度很高，第二次指派洒水任务时，未充分利用第一次已经执行了洒水任务的信息，而造成了对一些不必要的情况迭代，这也就是暴力求解（穷举）。显然，随着问题规模的增大，任务越来越复杂，暴力求解并不是个好的方法，以下可以避免如此高的复杂度。

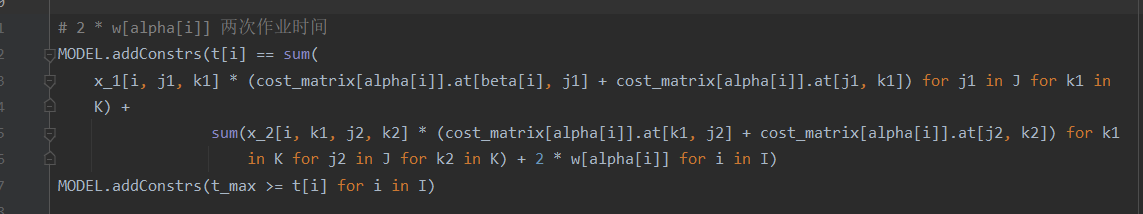
因此，考虑优化



其中表示在第一步求解的局部最优解的情况下，再去求解两部的全局最优解。而我们知道，每辆洒水车执行完作业后，都会回到补给点。那么第二次执行洒水任务，只依赖于第一次洒水任务执行完，每辆洒水车所在的补给点位置。那么可写出算法时间复杂度为



这就将算法复杂度下降一阶。



那篇博客的聚类，大概意思是把“现有的6个补给点强行作为6个聚类中心，然后再选2个“这样写进规划的约束里。然后对每个点进行分类就行了。

引入决策变量，表示从（候选点集合，作业点集合）

（为了与前文保持一致，这里记号是k,i而不是k,j，为的就是保持（作业点）这个符号表示含义不变。

如果以表示8个聚类中心的话，那么，也就是



对于从J几几那些点选的新补给点，将他们与Z一起并入集合里的元素，那么它也得满足补给不超过8次

限制了必须有8个补给点，其中每个候选点（包含6个补给点）都有两种选择，要么可以成为新补给点，要么不是补给点



但是这并没有约束选出了多少个新补给点，那么需增加





对于再高规模的算法，显然只能用启发式算法。比如遗传算法。