遗传算法在组合优化问题中的应用实例

摘要：本文先从遗传算法和组合优化问题介绍出发，给出个人理解和利用遗传算法解决组合优化问题的基本思路。接着依次2011年建模国赛B题、2020年mathercup挑战赛优化题和2020年建模国赛B题三个实例进行分析，根据不同的情况对遗传算法的框架进行修改，重新解决其中的组合优化问题。最后总结出利用遗传算法解决组合优化问题的基本思路和个人感悟。

一、研究背景

**1.1遗传算法**

遗传算法（Genetic Algorithm，GA）最早是由美国的 John holland于20世纪70年代提出,该算法是根据大自然中生物体进化规律而设计提出的。是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。该算法通过数学的方式,利用计算机仿真运算,将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时,相对一些常规的优化算法,通常能够较快地获得较好的优化结果。遗传算法已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。**【1】**

**1.2组合优化问题**

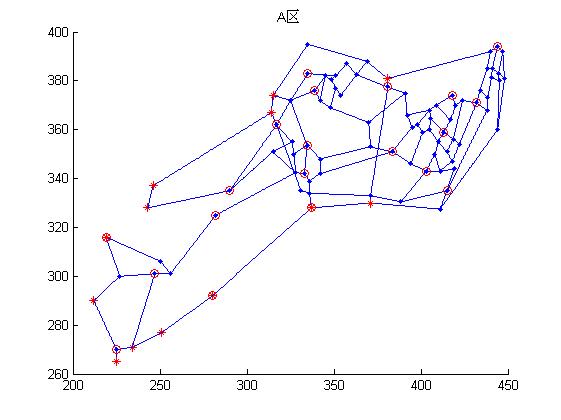
组合优化（Combinatorial Optimization）问题的目标是从组合问题的可行解集中求出最优解，通常可描述为：令Ω={s1，s2，…，sn}为所有状态构成的解空间，C(si)为状态si对应的目标函数值，要求寻找最优解s\*，使得对于所有的si∈Ω，有C(s\*)=minC(si)。组合优化往往涉及排序、分类、筛选等问题，它是运筹学的一个重要分支。

组合（最）优化问题是最优化问题的一类。最优化问题似乎自然地分成两类：一类是连续变量的问题，另一类是离散变量的问题。具有离散变量的问题，我们称它为组合的。在连续变量的问题里，一般地是求一组实数，或者一个函数；在组合问题里，是从一个无限集或者可数无限集里寻找一个对象——典型地是一个整数，一个集合，一个排列，或者一个图。一般地，这两类问题有相当不同的特色，并且求解它们的方法也是很不同的。**【2】**

**二、实例1——交警平台设置**

**2.1问题简述**

该问题节选自2011年数学建模国赛B题：

****

**说明：**

（1）图中实线表示市区道路；红色线表示连接两个区之间的道路；

（2）实圆点“·”表示交叉路口的节点，没有实圆点的交叉线为道路立体相交；

（3）星号“\*”表示出入城区的路口节点；

（4）圆圈“○”表示现有交巡警服务平台的设置点；

（5）圆圈加星号“○\* ”表示在出入城区的路口处设置了交巡警服务平台；

对于重大突发事件，需要调度全区20个交巡警服务平台的警力资源，对进出该区的13条交通要道实现快速全封锁。实际中一个平台的警力最多封锁一个路口，请给出该区交巡警服务平台警力合理的调度方案。（警车的时速为60km/h）

（点坐标和具体题目在附录的文件夹中）

**2.2 问题分析**

对于这个问题，首先我们要明确问题的本质是：在事件发生的时候如何选择20个平台中的13个平台出动警力前往13个目标点的距离的最大值的最小。即在解空间中找到最优解，使得该解的方案能在最短时间内完成封锁任务。

平台合理分配的基本思想是找寻每种可能方案中的平台到达交通要道的最长时间T中的最小值，其最小值所对应的方案便是合理分配方案。规划模型的数学规划式如下：





其中，P为20个交巡警服务平台标号集，G为13条交通要道标号集。

显然这是个经典的组合求最优化问题，首先要从20个交警平台中选择13个平台，再把13个平台分配给13个目标点，这个组合的计算量是非常大的。那我们如果应用遗传算法去求解这个问题，应该选择第二个步骤进行应用，即用编码的方式去表示13个平台的分配结果，而“20个平台去选择13个平台”这个步骤可以在编码生成，也就是种群的个体生成上进行操作。

**2.3 遗传算法求解**

1）产生初始种群

初始种群在产生方法上有两种：一是完全随机产生，它适用于对问题的解无任何先验知识的情况。随机性较强，因而也较公正。二是某些先验知识可转变为必须满足的一组要求，然后在满足这些要求的解中再随机地选取样本。

虽然在第二种方法中选择初始种群可使遗传算法更快的达到最优解，并且种群有一定的目标性和代表性，但取例不如完全随机的广泛，而且先验知识是否可靠也是一个问题。尤其是针对本题中平台在A区的分布算是广泛，与各个路口的距离并没有出现特别大的偏差，所以综合来说，我们选择完全随机产生的方法。并且在分析问题后，我们种群的内容并不是常规的二进制值，而是一段目标序列，这样的数据结构可以方便我们计算适应度和交叉。

2）适应度函数

根据模型和种群内容，我们可以直接根据种群的个体序列来计算每段序列对应的路口与平台之间的距离集合中的最大值max作为适应度大小。并且为了方便后面的筛选，在返回值上应该同时设置最大值和最小值。对于某个个体，适应度越小说明该个体越优秀，适应度越大说明个体越差。



3）选择

针对本题的特点来设置，选择将使适应度较小(优良)个体有较大的存在机会，而适应度较大（低劣）的个体继续存在的机会也较小。我们采用赌轮选择机制，令表示群体的适应度值之总和，表示种群中第个染色体的适应度值，它产生后代的能力正好为其适应度值所占份额。

4）交叉

基于类路径表示的编码方法，要求一个个体的染色体编码中不允许有重复的基因码，也就是说要满足任意一个路口必须而且只能匹配一个平台的约束。基本遗传算法的交叉操作生成的个体一般不能满足这一约束条件。这里我们选择部分匹配交叉(PMX)方法。

部分匹配交叉操作要求随机选取两个交叉点，以便确定一个匹配段，根据两个父个体中两个交叉点之间的中间段给出的映射关系生成两个子个体。

5）变异

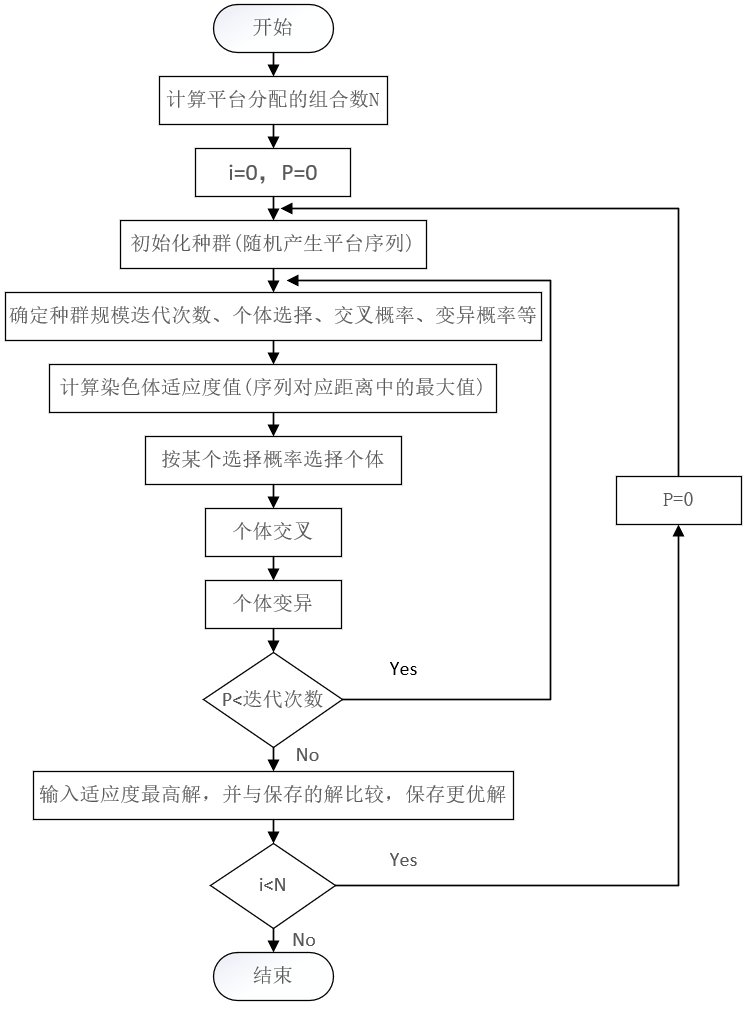
基于二进值编码的变异操作不能适用，不能够由简单的变量的翻转来实现。在序列选择问题中个体的编码是目标的序列，随机的在这个序列抽取两个路口，然后交换他们的位置。这样就实现了个体编码的变异，算法如下：

1、产生两个0到1之间的随机实数；

2、将这两个随机实数转化为0到n（目标路口数）-1之间的随机整数；

3、将这两个随机整数指代的路口进行交换。

**2.4流程图**



**2.5求解结果**

在种群数量100、迭代次数300的情况下得出合理分配方案，该方案下平台到达交通要道的最长时间T为8.015457min，其为每种可能方案的最小值。具体分配方案如下：

表6.3.3：站台分配方案

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 服务平台节点 | 对应交通要道 |  | 服务平台节点 | 对应交通要道 |
| A1 | 62 |  | A11 | 24 |
| A2 | 38 |  | A12 | 23 |
| A3 | 16 |  | A13 | 21 |
| A4 | 48 |  | A14 | 22 |
| A6 | 30 |  | A15 | 28 |
| A7 | 29 |  | A16 | 14 |
| A10 | 12 |  |  |  |

三、实例2——货仓拣货问题

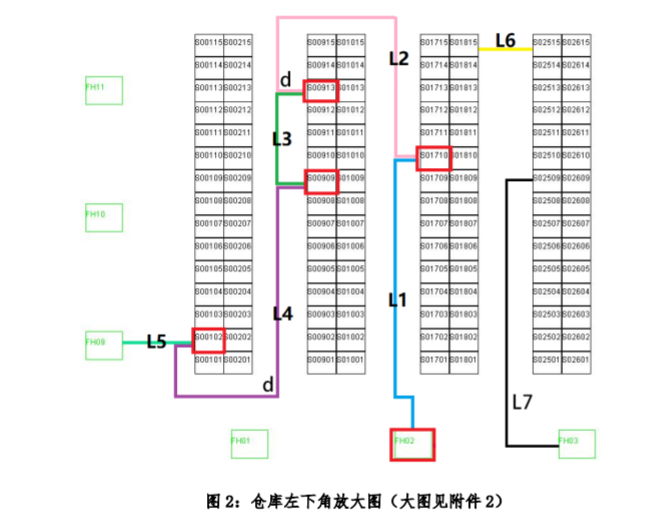
**3.1 实例2前言**

实例2是我参加比赛但因为队友跑路，最后做好了答案文章没写所以没有提交。该实例的代码借鉴了遗传算法的框架进行修改**【3】**，可以实现最佳路线示意图的实时变化和收敛曲线，用法在压缩包里有说明。

**3.2 问题简述**

该实例节选自2020年mathercup高校数学建模挑战赛C题：

在一个仓库中，拣货员根据任务单的内容从某个复核台出发依次拣货，然后完成任务单的拣货任务后选择一个复核台提交。



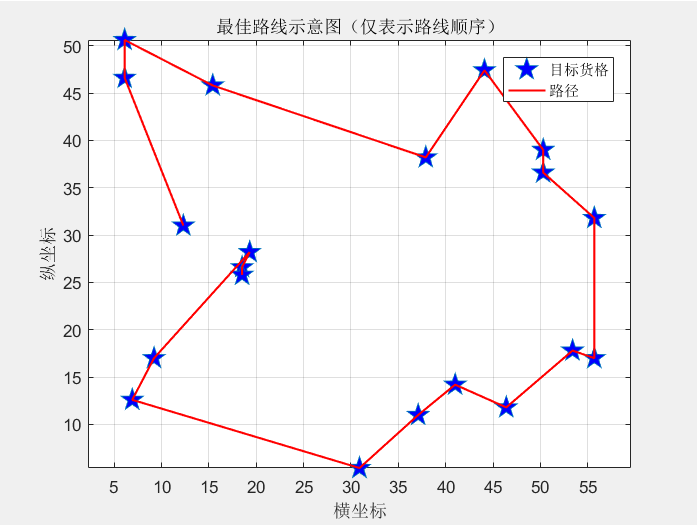
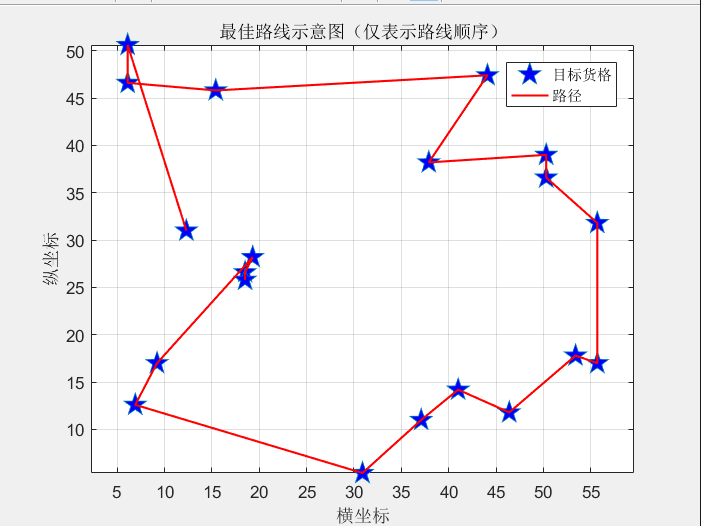
假设所有复核台正常工作，任务单 T0001 等待拣货，拣货员 P 在复核台 FH10 领取了任务单 T0001。请给 P 规划理想的拣货路线，包括货 格访问顺序、返回的复核台，计算完成出库花费的时间（拣货员拣货开始 到所有任务复核打包完成花费的时间）。

**3.3 问题分析**

这个问题的本质是求得最短路线，但与普通的TSP问题不同是出发点和提交点、以及在仓库的拣货路线其实分为两个部分。因此在计算适应度的时候，选取的应该是个体编码转换的欧拉回路去掉一个边之后，分别连接两个复核台的最短路程。也就是说，外层是遗传算法，内层还有一个去掉某个边、选取复核台重新计算路线的过程。幸好在前面已经求得每个货格和每个复核台之间的距离矩阵，因此直接查表选最短组合就行，内层并不麻烦，顶多是o（n2）的复杂度。

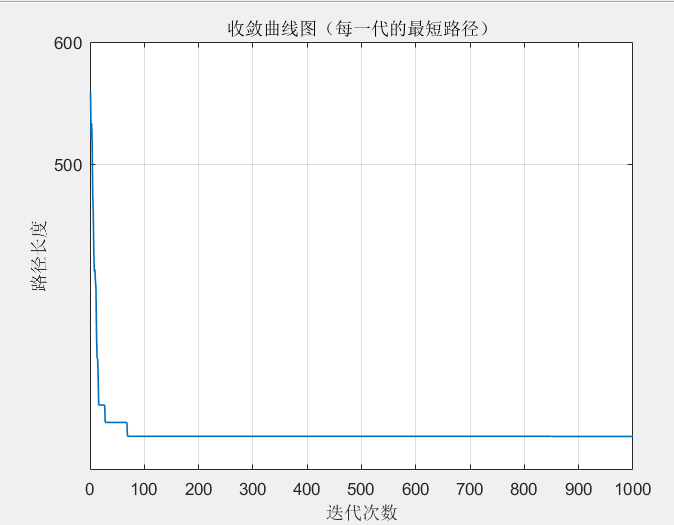
**3.4 遗传算法求解**

该实例的matlab运行的实时变化变化类似如下图：



这是在单次运行中的一次变化，每次运行该图会变化六七次。

收敛曲线：



四、实例3——沙漠探险游戏

**4.1 实例3前言**

该实例是我刚刚九月份参加的建模国赛B题，但真的太难了，最后可惜只拿到省二，但好像是咱们学校选B题做的最好的了（就我所知），所以不确定答案做的是不是对。有六张地图，其中的第一问还是确定做法的，也比较有特点，用了一些小技巧，在计算适应度的时候**动态地改变编码意义**。

**4.2 问题简述**

考虑如下的小游戏：玩家凭借一张地图，利用初始资金购买一定数量的水和食物（包括食品和其他日常用品），从起点出发，在沙漠中行走。途中会遇到不同的天气，也可在矿山、村庄补充资金或资源，目标是在规定时间内到达终点，并保留尽可能多的资金。

游戏的基本规则如下：

（1）以天为基本时间单位，游戏的开始时间为第0天，玩家位于起点。玩家必须在截止日期或之前到达终点，到达终点后该玩家的游戏结束。

（2）穿越沙漠需水和食物两种资源，它们的最小计量单位均为箱。每天玩家拥有的水和食物质量之和不能超过负重上限。若未到达终点而水或食物已耗尽，视为游戏失败。

（3）每天的天气为“晴朗”、“高温”、“沙暴”三种状况之一，沙漠中所有区域的天气相同。

（4）每天玩家可从地图中的某个区域到达与之相邻的另一个区域，也可在原地停留。沙暴日必须在原地停留。

（5）玩家在原地停留一天消耗的资源数量称为基础消耗量，行走一天消耗的资源数量为基础消耗量的 倍。

（6）玩家第0天可在起点处用初始资金以基准价格购买水和食物。玩家可在起点停留或回到起点，但不能多次在起点购买资源。玩家到达终点后可退回剩余的水和食物，每箱退回价格为基准价格的一半。

（7）玩家在矿山停留时，可通过挖矿获得资金，挖矿一天获得的资金量称为基础收益。如果挖矿，消耗的资源数量为基础消耗量的 倍；如果不挖矿，消耗的资源数量为基础消耗量。到达矿山当天不能挖矿。沙暴日也可挖矿。

（8）玩家经过或在村庄停留时可用剩余的初始资金或挖矿获得的资金随时购买水和食物，每箱价格为基准价格的2倍。

第一关

参数设定：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 负重上限 | | 1200千克 | 初始资金 | 10000元 | |
| 截止日期 | | 第30天 | 基础收益 | 1000元 | |
| 资源 | 每箱质量（千克） | 基准价格  （元/箱） | 基础消耗量（箱） | | |
| 晴朗 | 高温 | 沙暴 |
| 水 | 3 | 5 | 5 | 8 | 10 |
| 食物 | 2 | 10 | 7 | 6 | 10 |

天气状况：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 日期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 天气 | 高温 | 高温 | 晴朗 | 沙暴 | 晴朗 | 高温 | 沙暴 | 晴朗 | 高温 | 高温 |
| 日期 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 天气 | 沙暴 | 高温 | 晴朗 | 高温 | 高温 | 高温 | 沙暴 | 沙暴 | 高温 | 高温 |
| 日期 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 天气 | 晴朗 | 晴朗 | 高温 | 晴朗 | 沙暴 | 高温 | 晴朗 | 晴朗 | 高温 | 高温 |

地图：

起点1

2

6

3

4

5

8

9

7

10

11

矿山12

13

14

村庄15

16

17

19

18

20

21

22

23

24

25

26

终点27

**4.3 问题分析**

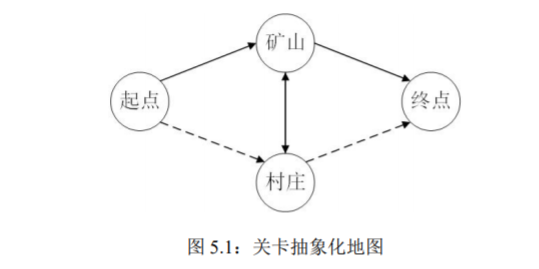
沙漠探险游戏是个动态又自由的过程，在规则天数下，可以选择去矿场挖矿，也可以选择直接去终点。在沙漠中无论是停留还是移动，在矿场中无论是挖矿还是休息，都要消耗食物和水，在预计没有食物和水的时候要计算是前往终点还是去村庄补给。但即使是去村庄补给，路上也会消耗，补给完后也要考虑是接着挖矿还是离开前往终点。

因此其中的变数太多，不能用简单的方法去求解。

在玩法上可以分为两种情况：前往矿山/村庄挖矿赚钱，然后回到终点。或是直接前往终点。

如果是前者，可以大致分为三个过程：前往村庄/矿场、在村庄-矿场之间作业、返回终点三个过程。

观察地图和规则可以知道，在沙漠中无意义的走动是优化结果中绝不允许的，所以第一步是把地图抽象化，用Floyd算法求出抽象化地图的距离矩阵。



为了能求解，我采用了二进制编码的方式去表示30天的决策向量。因为在前往村庄/矿山的过程中，操作只有前进和休息；在矿山中只有挖矿和休息；在补给和返回过程中和前进过程一样。因此一段和日期相同长度的0和1组成的代码就能代表一次探险的操作。

也就是说，外层是二进制编码的遗传算法，内层是根据个体编码转换成动态模拟过程，如果能存活就保留最后返回的金额数，以此作为适应度的指标。

而难点也在于内层的设计，在已知天气的情况下，需要时刻监测是否能返回终点（优先级最高），预测是不是需要补给（优先级次之），以及补给之后还值不值得回到矿山继续挖矿（优先级次次之）。

所以整体是个十分复杂的模拟过程，但仍然可以缩小解空间的范围，比如在编码设计上很显然“前进/挖矿”比起“停留”肯定是更加积极的行为，所以可以在随机种群的时候让积极的行为更多一些。

**4.4 改进遗传算法的细节**

遗传算法包括三个基本操作：选择、交叉和变异。我们分开从这些细节进行

修改。

1）产生初始种群

种群的每个个体的编码代表的是 30 天内玩家可能会使用的决策向量，根据

目标模型可知虽然整个过程分为三段，但决策变量一直是 0-1 变量。因此在产生

种群上我们直接采用长度为 30 的二进制编码即可。并且因为以赚钱和不在路上

浪费过多时间为目标，我们将 1 的生成概率设为 0.8,而 0 的生成概率为 0.2。

2）适应度函数

根据模型和种群内容，最优目标应该是成功完成游戏并且剩余金钱最高的个

体。因此我们修改为以输出的剩余金钱变量为主要的衡量对象。并且保存并输出

最好情况的个体编码。

8

3）选择

针对本题的特点来设置，选择将使适应度较大(即结束游戏后金钱最多)个体

有较大的存在机会，反之个体继续存在的机会较小，在选择上采用赌轮选择机制。

4）交叉

根据个体编码的“1 多 0 少”的特点，使用一次交叉，随机选取两个点，将两

个父个体中两个交叉点之间的中间段直接交叉，并将其余部分组合。这样的好处

是避免有规律的交叉可能会造成少数地位的“0”规律出现某几个位置，进而导致

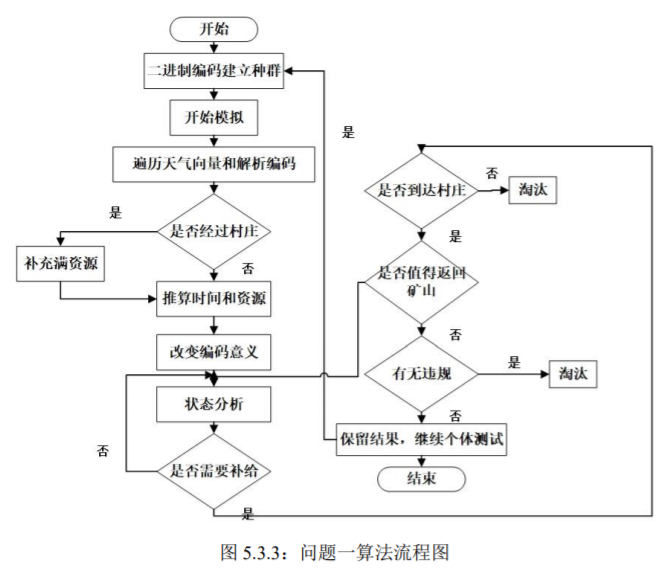
容易收敛到局部解。

5）变异

由于基于二进值编码的变异操作不能够由简单的变量的翻转来实现，可以利

用“1 多 0 少”，随机的在这个编码序列选取两个位置，然后交换他们的位置。这

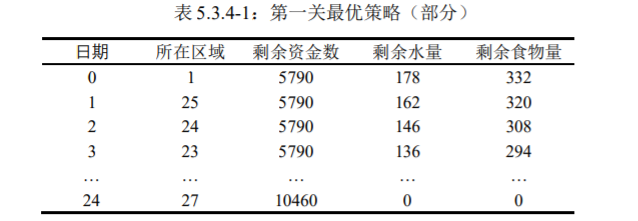
样就实现了个体编码的变异。**4.6 流程图**



**4.5 求解结果**

通过算法去求解模型，以迭代次数 20，种群数量 50，在外层通过不同的初始

水量和食物量的组合，求得的第一关最优策略部分如下表：



五、应用方法总结及个人感悟

**5.1 应用方法总结**

从三个实例可以看出，遗传算法解决组合优化问题的关键是如何将问题的本质转化成编码的方式，并且根据题意对遗传算法的框架进行细致的修改。

比如说实例1中遗传算法处理的部分是13个平台和13个路口的组合，实例2是拣货的顺序问题，这两个的交叉都是基于交换的方式。而实例3则采用了二进制编码，交叉方式和变异方式可以用更多的方法。

5.2 个人感悟

通过整理过去有关遗传算法的应用实例，让我对遗传算法有了更深的理解，比如说这次国赛我就想到了动态过程中改变编码意义的技巧。在沙漠探险后面的题目中我还用了格雷码+决策向量码结合的方式，用格雷码来选择道路。因为格雷码的特点是只改变一位变化不大，形式比较紧密，比较契合变异和交叉操作，尤其是这种1多0少的编码方式。

除了我标注的研究背景和框架借鉴有明确标注尾标外，其余全部都是自己原创写的。希望未来能接触更多的算法知识，代码和使用方法在压缩包里，比较多就不贴了，老师如果要看的话麻烦了，辛苦了！

六、参考资料

[1] 郑树泉．工业智能技术与应用[M]．上海：上海科学技术出版社，2019：250-251

[2] 《组合最优化算法和复杂性》，高等教育出版社，1988，C.H. Papadimitriou, K. Steiglitz

[3]遗传算法解决TSP问题MATLAB实现 [2019-01-21](https://www.omegaxyz.com/2019/01/21/matlab-tsp-all/) <xyjisaw> https://www.omegaxyz.com/2019/01/21/matlab-tsp-all/