2020 年高教社杯全国大学生数学建模竞赛

承　诺　书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称"竞赛章程和参赛规则"，可从 http://www.mcm.edu.cn 下载）。

我们完全清楚，在竞赛开始后参赛队员不能以任何方式，包括电话、电子邮件、"贴吧"、QQ 群、微信群等，与队外的任何人（包括指导教师）交流、讨论与赛题有关的问题；无论主动参与讨论还是被动接收讨论信息都是严重违反竞赛纪律的行为。

我们完全清楚，在竞赛中必须合法合规地使用文献资料和软件工具，不能有任何侵犯知识产权的行为。否则我们将失去评奖资格，并可能受到严肃处理。

我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D/E 中选择一项填写）：　B

我们的报名参赛队号（12 位数字全国统一编号）：　202008026022

参赛学校（完整的学校全称，不含院系名）：　黑龙江科技大学

参赛队员 (打印并签名) ：1.　王炬皓

2.　姚征兵

3.　孙博

指导教师或指导教师组负责人 (打印并签名)：

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期：   2020 年 09 月 13   日

（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

2020 年高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅记录（可供赛区评阅时使用）：

评阅人						
备注						

送全国评阅统一编号：
（赛区组委会填写）

（请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。）

基于组合排序及马尔可夫过程的策略优化问题求解

摘要

在日常生活中,连续决策的行为无时无刻不在发生,并且上一个决策往往影响下一个决策的行为发生的先决条件。研究这种行为能够指导企业及个人的决策从而为企业和个人的长期发展带来收益。

对于有限步态的行为,行为的种类有限且信息完全的模型中,可以采用分支限界法来裁剪不满足限制条件的解空间树下的任意子树。以问题一为例,模型以用户每天所在的地点以及当天所消耗的资金—资源当量(当天所消耗的资源兑换成资金的数量,下同)作为连接下一个解空间子树的边权重,分支限界法可利用玩家的负重小于等于 1200 千克,初始购买的资金等于 10000 元以及截止日期为 30 天等限制条件来裁剪解空间树的子树以减少运算量。

对于第二问,由于下一天的天气信息缺失,因此利用玩家在 10 天内经历天气的先验概率作为马尔可夫过程的状态转移概率,并将当天经历的天气作为后验概率对马尔可夫状态转移概率进行更新,来进一步计算下一步状态转移对剩余资金数量的影响。由于要达成通过关卡的目标,但是天气情况未知,在起点时必须假设天气最相对劣的情况。因此与第一问不同,第二问不可避免地存在资源折旧的情况。因此在任何情况下,模型都只能给出当前状态的最佳策略,并要求身边的资源大于等于在该区域到达终点期间天气相对恶劣的情况下所需要的资源。

对于第三问,有 $n>1$ 的玩家个数,比赛规则对玩家达成目标前路径规划的时间域和空间域的交接域做出了惩罚,并且若存在交接点收益会是原来的 $\frac{1}{n}$ 倍。利用蚁群算法可以通过限定迭代次数来使得系统多次模拟 n 个玩家的闯关行为,但每一次模拟都利用了上一次模拟遗留的信息,因此模拟得到的玩家闯关资金数会逐渐收敛到一个较优解。

最后,对模型的优缺点进行评价,并提出改进的方向。

关键词:分支限界法　路径规划　马尔可夫过程　蚁群算法

一、问题重述

1.1 问题背景

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。

1.2 目标任务

问题一**:**一位玩家在已知整个游戏时段内的天气状况事先全部已知,并在满足游戏初始参数设定和规则的情况下,给出一般情况下玩家的最优游戏策略。求解附件中的”第一关”和”第二关”,并将相应结果写入表格。

问题二**:**假设只有一名玩家，玩家仅知道当天的天气状况，可据此决定当天的行动方案，试给出一般情况下玩家的最佳策略，并对附件中的“第三关”和“第四关”进行具体讨论。

问题三**:**现有n名玩家，他们有相同的初始资金，且同时从起点出发。若某天其中的任意$k\left(2\le k\le n\right)$名玩家均从区域A行走到区域B($\mathrm{B}\ne\mathrm{A}$)，则他们中的任一位消耗的资源数量均为基础消耗量的$2k$倍；若某天其中的任意$k\left(2\le k\le n\right)$名玩家在同一矿山挖矿，则他们中的任一位消耗的资源数量均为基础消耗量的3倍，且每名玩家一天可通过挖矿获得的资金是基础收益的$\frac{1}{k}$；若某天其中的任意$k\left(2\le k\le n\right)$名玩家在同一村庄购买资源，每箱价格均为基准价格的4倍。其他情况下消耗资源数量与资源价格与单人游戏相同。并在给定限制条件下分别对附件中的”第五关”和”第六关”做讨论。

二、问题分析

问题一的分析:

考虑到游戏的目标是在规定时间内到达终点,并保留尽可能多的资金,而持续的挖矿可以带来更高的资金收益,因此通过编程先求出不经过矿山直接通过最短路径到达终点的最大剩余资金数,再求出到矿山挖矿所能得到的可能最大资金数。在挖矿过程中可能要多次在村庄处进行资源补给,而在村庄购买资源的价格是起点的两倍,因此算法的任务是确定在矿山采矿的天数以及在村庄补给次数来满足在起点、村庄、终点、村庄、矿山选择最短路径的情况下,同时满足负载条件,达到游戏结束时资金的最优值。

问题二的分析:

由于下一天天气条件未知,但历史天气数据已知,因此玩家可以依据历史各种天气发生的概率及方差来推算接下来各种决策的花费或收益以及资源的负重等等因素。对于该题,通过引入正态分布的3σ准则来对各种天气发生概率的边界条件做出限制,并且优先考虑更大风险的情况。因为受达成目标条件的限制,玩家需要实时考虑当前条件下的资源情况,因此不可避免地存在天气状况比预期策略更优带来的资金折旧,故资金折旧的情况时有发生。玩家的任何动作只能达到局部最优,而无法考虑全局最优。

问题三的分析:

第三问分为两小问,梁晓雯的区别和问题一与问题二类似,但玩家的数目大于1,并且对玩家行走路线的时空所消耗的资源和挖矿收益做了限制。为了表达出玩

家之间的交互作用,扩大问题空间的搜索范围,问题三采用基于蚁群算法的多元路径规划,通过合理设置蚁群算法的信息素来表达玩家规划路径的过程。多元搜索算法保证了一个时间段会有多个点模拟玩家的行为并在搜索过程中逐渐收敛到一个较优解。

三、模型假设

3. 1 假设玩家为了利益最大化不会刻意去绕远路

3. 2 假设起点、村庄、终点、村庄、矿山四点之间的最短路径确定过一次,则玩家为求安稳不会选择其它路径长度与其相等的最短路径。

3.3 假设为了可以在村庄随时兑换资源而灵活调整路径,玩家在路径等长的情况下偏向于选择经过村庄的路径。

3.4 假设玩家无法直接通过只有一个点相接的两个区域,区域之间联通性存在的必要条件是相邻区域至少有一条边相接。

3.5 假设玩家为完成条件为风险厌恶型理性人,玩家应保证身边的资源能够满足以较恶劣的天气下到达终点。

3.6 假设玩家在每个历史时间段,晴朗、高温、沙暴天气出现的概率相对稳定

四、符号说明

$Cost\{i,j,k,\cdots,m\|t\}$	在第 t 天开始经过路径 $i,j,k,\cdots,m$ 的资金—资源当量值
$Cost(T_{dig}\|t)$	初始时刻为 t 时在 T 天内矿山挖矿消耗的资金一资源当量值
$P(i->k)$	从区域 i 到区域 k 的最短路径长度
$T(i->k)$	从区域 i 到区域 k 的最短路径所经历的天数
$S^i(t)$	在 t 时刻开始停留矿山的天数,假设这是玩家第 i 次到达矿山
n	玩家到达矿山次数计数
$W(\arg s)$	计算相应资源消耗需要的重量
$\Pr(wh=\{1,2,3\}\|t)$	第 t 天下一天天气为给定编号时的概率, wh 只能为 $\{1,2,3\}$ 中的一个数,代表当天的天气为晴朗、高温或是沙暴
$Acc(\{1,2,3\}\|t)$	直到第 t 天为止,编号为 i 的天气累计出现的天数
$wh(t)$	第 t 天的真实天气
$\Pr(i\|t)$	从第 0 天到第 t 天沙尘暴发生的概率

$\sigma(t)$	从第0天到第t天天气历史数据计算得出的沙尘暴标准差数值
$E(t)$	第t天对下一天资金—资源消耗当量值的期望
$L(t)$	第t天对下一天所需要的资源重量的期望
$\tau_{ij}(\mathrm{t})$	从区域i到区域j路径上的信息素浓度在第t天的大小
$P_{ij}^{k}(t)$	t时刻玩家从区域i转移到区域j的概率
$\eta_{ij}(t)$	从区域i到区域j路径上的启发式函数在第t天的大小
ρ	是一个常数,表示信息素的挥发度,并且$0<\rho<1$
Q	是一个常数,表示玩家在循环一次释放的信息素总量
L_k	第k个玩家通关的天数
γ	惩罚函数因子,以保证

五、模型的建立与求解

5.1 问题一:所有天气状况已知条件下的模型

5.1.1 路径分析与求解空间搜索

根据**3. 2**,**3. 3**的假设,可以把第一关的图简化成如图1所示的带权有向图,其中a和c所需要的花费和负重分别为:

$$a=Cost\{1,25,24,22,9\mid t=0\}=1080, c=Cost\{1,25,26\mid t=0\}=590$$

$$W(a)=180kg, W(c)=115kg$$

是两个固定的值,因为它们的起始时间都为第0天,并且最短路径所到达的区域固定。e和d相互制约,它们的值取决于在矿山的逗留天数和村庄内补给的次数以及补给量,是多次路径的资金—资源当量值求和的结果。为了避免资源过剩带来的资金折旧,终点处兑换带来的折旧,玩家要在村庄购买资源则只够买恰好到下一次再到村庄或者到终点恰好需要的资源总量。上述条件描述成为等式形式:

$$Cost(b+d)+\sum_{n=1}^{\max n}Cost(T_{dig}\mid t_n)=Cost(e) \tag{5-1}$$

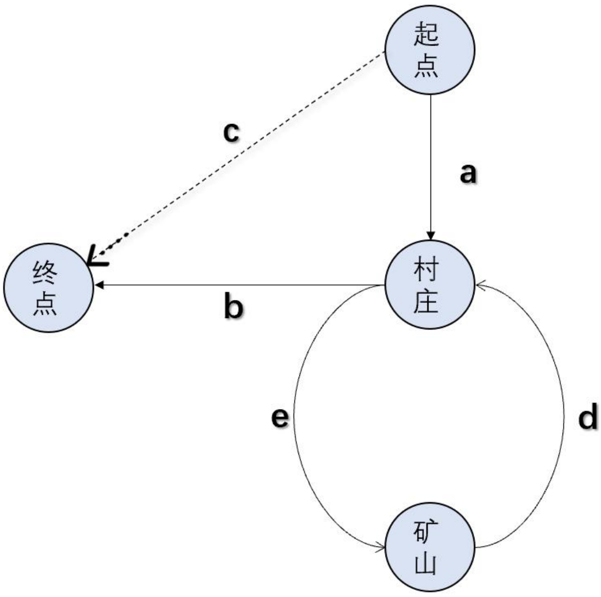


图 1.四节点之间关系简化后的有向带权图

因为$T(i->k)$被定义为从节点i到节点k的最短路径所经历的天数,从图中可以直观看出$T(1->12)=T(1->15)+T(15->12)$,这个结论对$i$和$k$没有普适性,但对于这三个节点这个结论没有错误。为了简化公式的表达,下列表达式利用了这个事实。

对于整个图的资金流通而言,则完成资金的线性规划函数为:

$$Max \quad z=1000t-Cost(d)-(a-c+Cost(b)+Cost(e))+10000 \qquad \textbf{(5-2)}$$

$s.t.$

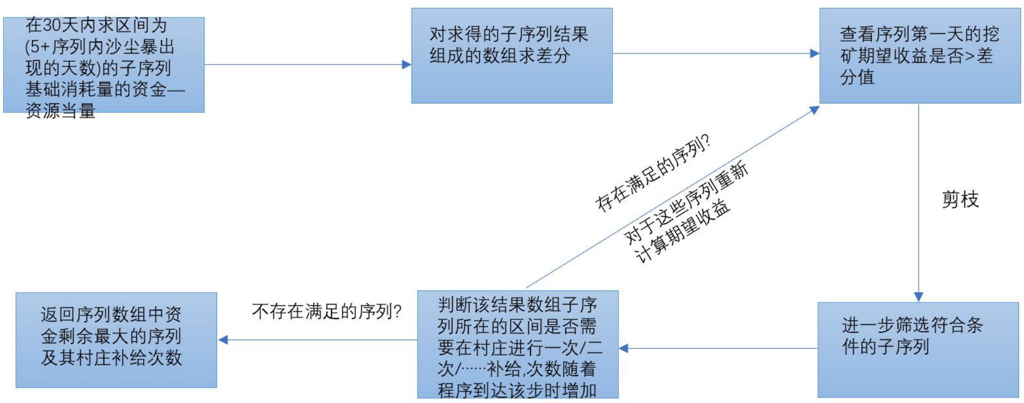
$$W(a+Cost(P(15->12)\,|\,t=T(1->12))+Cost(T_{dig}\,|\,t=T(1->12))+d+b)<=1200$$

$$T(1->12)+T_{dig}\,|\,t=T(1->12)+T(12->27)<=30$$

$$Cost(a+T(15->12)+T_{dig}\,|\,t=T(1->12)+d+b)<=10000$$

满足解搜索空间内所有的e、d的解的数值大小决定了$e->d->e$环被循环了几次,数值较大的e、d的解说明$e->d->e$被循环了多次(也即玩家在村庄购买了多次补给)。而目标函数z则是找到使玩家到达终点后资金数量达到最大的那条路径。

因为逗留在矿山挖矿的资金成本体现在从矿山返回终点的最短路径上的资源消耗,而$T(12->27)=5$,如果遇到沙尘暴天气则必须逗留,则计算的子序列长度+1。根据这条路径成本规则,给出了搜索解空间的算法流程:



C++编程得到的路径为第 0 天到第 8 天到村庄进行补给,第 10 天到达矿山,挖矿挖到第 13 天再会村庄补给,第 16 天开始挖矿挖到第 24 天,第 25 天原地等待,剩余的天数回到终点。
最大的到达资金数是 12935。

5.1.2 新的带权有向图与新的解空间树

第二问求解思路与第一问近似,新的带权有向图表示如图 2 所示。为了节约资金,玩家应选择先到达矿山 1 再到达矿山 2,期间路过村庄购买补给品。因此问题变成一个时间分配问题,即在满足时间限制的情况下如何分配工时到玩家在第一个矿山的挖矿时间和第二次挖矿的时间来使得步行成本最低,挖矿收益最大的问题。

对于第二关,a,k 两条固定路径的消耗花费及负重是:

$$\begin{gathered} a = Cost\{1,9,18,26,27,28,29 \mid t=0\} = 1580 \\ k = Cost\{1,9,18,26,35,43,52,60,61,62,63 \mid t=0\} = 2470, \\ W(a) = 284kg, W(k) = 417kg \end{gathered} \tag{5-3}$$

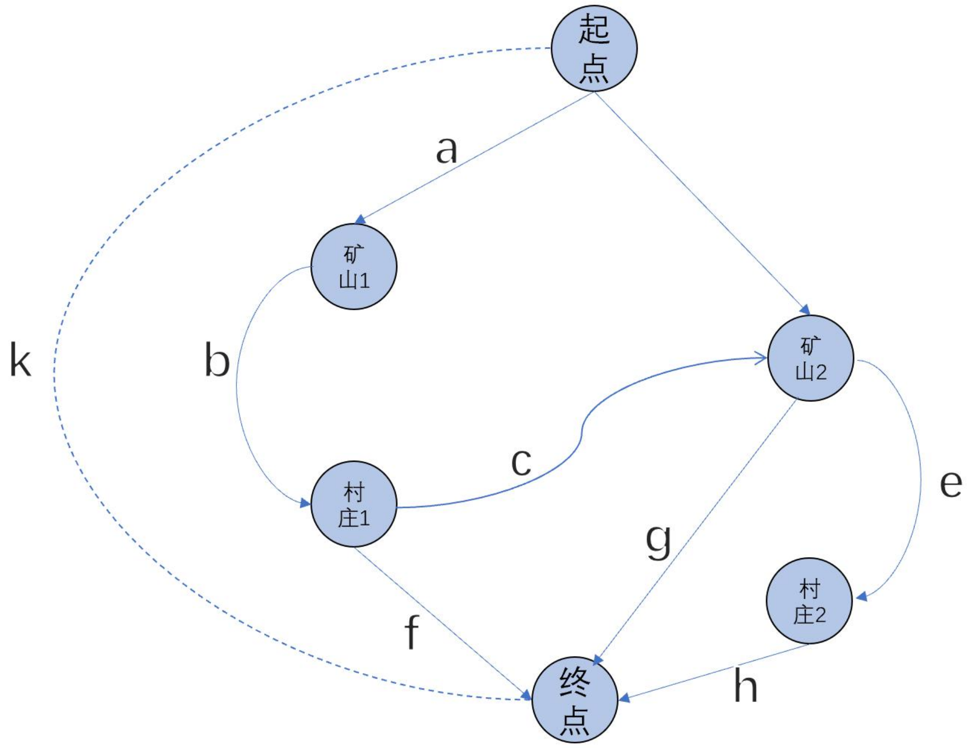


图 2.关卡 2 之间关系简化后的有向带权图,矿山 1 对应的区域为 30,村庄 1、矿山 2、村庄 2 对应地图区域依次为为 39,55,62,图中没有矿山 1 到矿山 2 的边是因为矿山 1 到矿山 2 的最短路径本来就包含节点村庄 1。

C++编程得到的路径为第 1 天到第 9 天到达矿山 1,第 10 天到第 14 天挖矿,第 15 天到村庄补给,第 16 到第 19 天到达第二个矿山,第 20 天到第 24 天在矿山 2 挖矿,第 25 天等待一天,第 29 天到第 30 天返回终点。
最大的到达资金数是 16465。

5.2 问题二:未知天气状况条件下的模型

5.2.1 天气预测模型的建立及路径分析

马尔可夫过程描述状态空间中从一个状态到另一个状态转换的随机过程,在未知下一天天气状况的条件下,马尔可夫过程的状态迁移概率只能从玩家已经经历的天气作为先验概率得出,但在初始条件下需要加上拉普拉斯平滑处理:

$$\Pr(wh=1|t=0)=\Pr(wh=2|t=0)=\frac{1}{2} \quad \textbf{(5-4)}$$

因此预测的第$t+1$天天气为编号i的概率为:

$$\Pr(wh=i|t+1)=\frac{Acc(1|t)+1}{t+2}, i=\{1,2,3\} \quad \textbf{(5-5)}$$

下一天天气已知后需要更新概率:

$$Acc(i|t+1)=Acc(i|t)+1 \qquad if \quad wh(t)=i \quad \textbf{(5-6)}$$

算法在$Acc(i|t+1)$更新后也会依据更新的数值计算$\Pr(wh=i|t+2)$的概率数值将数值迭代到第 10 天作为玩家策略的制定依据。

针对问题 2 第三、四关的路径分析图如图 3、4 所示,玩家同样可选择直接到达终点或者选择在矿山挖矿,与问题一不同的是,此时路径a,b都不能确定需要消耗的资源数及负重量。

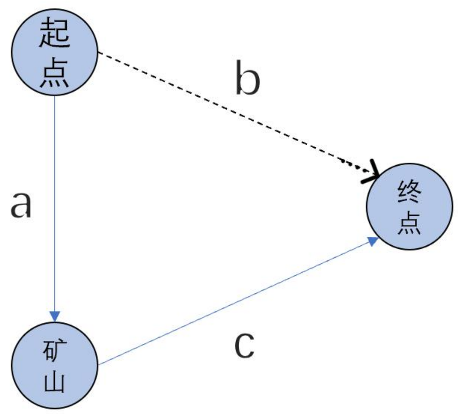


图 3.第二问第三关的路径描述

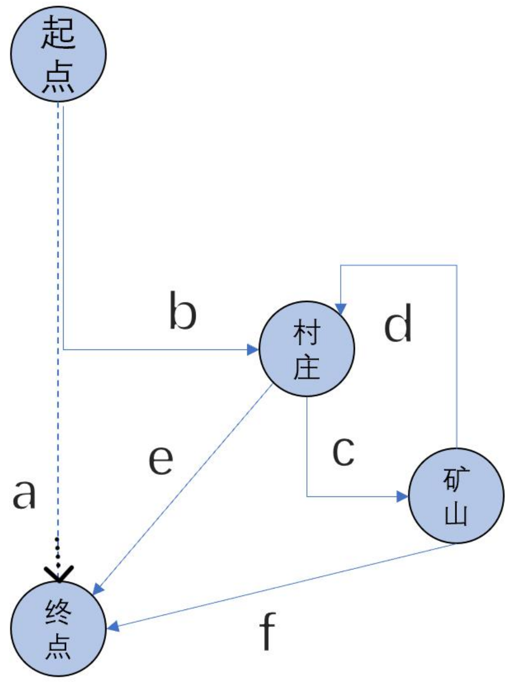


图 4.第二问第四关的路径描述

5.2.2 资金变动分析及最优策略制定

对于第三关,因为要满足至少通关的条件,假设边消耗资金的计算基于所有天的天气都是高温,因此a、b的消耗资金如下:

$$Cost(a) = 810$$
$$Cost(b) = 1080$$

而如果玩家选择去挖矿则要付出天数之差为$T(1->9)+T(9->13)-T(1->13)=2$天的基础消耗量,对应最恶劣情况下的资金为$540$,则若要覆盖成本,挖矿带来的效益需要覆盖$Cost(a)-Cost(b)+540=270$的资金,而该题中玩家最多在矿山挖矿的时间为

$$10-T(1->9)-T(9->13)+1=5 \qquad \textbf{(5-7)}$$

因为已知在 10 天内无沙暴天气若在晴朗天挖矿带来的收益是$200-55\times 3=35$,在高温天带来的收益为$200-135\times 3=-205$,故 5 天带来最大收益的情况为$35\times 5=175<270$,因此身处第三关的玩家应该选择直接到达终点。

对于第四关,根据假设 3.3,玩家为了随时补充资源以便灵活调整路径玩家会选择先到达村庄的路径,并且在图中这条路径同样包含在从起点直接到达终点的最短路径中。对应地路径为 1->2->3->4->9->14。

因为不管是村庄还是矿山到达终点的最短路径长度都为 3,根据正太分布的3σ准则,接下来 3 天发生沙尘暴的天数以99.7%的概率小于$3(\Pr(3|t)+\sigma(3|t))$天,其中t是到达村庄或矿山的天数。根据假设 3.5 可以认为接下来三天高温的天数期望为$3(\Pr(2|t)+3\sigma(2|t))$天,由于沙暴天必须逗留,对应晴朗天数的期望为$3-3(\Pr(2|t)+\sigma(2|t))$,因此所需要的期望资金为:

$$\begin{aligned} E(t)_c &= 900(\Pr(3|t)+\sigma(3|t))+810(\Pr(2|t)+\sigma(2|t))+330-330(\Pr(2|t)+\sigma(2|t)) \\ &= 330+990(\Pr(3|t)+\sigma(3|t))+480(\Pr(2|t)+\sigma(2|t)) \end{aligned} \qquad \textbf{(5-8)}$$

期望的负重为:

$$L(t)_c = 102+300(\Pr(3|t)+\sigma(3|t))+178(\Pr(2|t)+\sigma(2|t)) \qquad \textbf{(5-9)}$$

并且根据归一化条件有:

$$\Pr(1|t)+\Pr(2|t)+\Pr(3|t)=1 \qquad \textbf{(5-10)}$$

$\Pr(1|t),\Pr(2|t),\Pr(3|t)$的运算规则由公式(5-5)及(5-6)给出。

根据题设条件,在三十天内较少出现沙暴天气,可认为沙暴天气出现的概率小于$\frac{1}{3}$,故晴朗天气和高温天气发生的概率和大于$\frac{2}{3}$,反映到某个时间段,它之前出现的事件也应该如此:

$$\Pr(2|t)+\Pr(3|t)>\frac{2}{3} \qquad \textbf{(5-11)}$$

将三天作为一个步态是合适的,玩家在任何时间段都应该根据(5-8)、(5-9)及(5-11)作为行动依据,保证自己能顺利到达终点的同时使得资金最大化。

现在考虑在矿山挖矿时取得的期望收益分别为三组天气的发生概率乘以三种天气下挖矿取得的收益值,而晴朗,高温,沙暴天气下挖矿取得的收益值分别为$835,595,550$对应资源所需要的负重是$51kg,145kg,150kg$,则三种天气情况所对应的单位负重收益由大到小排序对应$r_{晴朗}>r_{高温}>r_{沙暴}$,引入概率模型后,挖矿一天的所带来的期望收益$E(r)$可以确定:

$$\begin{aligned} E(t)_r &= 550(\Pr(3|t)+\sigma(3|t))+595(\Pr(2|t)+\sigma(2|t))+835(1-\Pr(2|t)) \\ &= 835+550(\Pr(3|t)+\sigma(3|t))-240(\Pr(2|t)+\sigma(2|t)) \end{aligned} \qquad \textbf{(5-12)}$$

对应的负重期望为:

$$L(t)_r = 51+150(\Pr(3|t)+\sigma(3|t))+94(\Pr(2|t)+\sigma(2|t)) \qquad \textbf{(5-13)}$$

从关卡 4 的地图来看,玩家如果选择挖矿,则多走的两段路程发生在第六、第七天,对应的资金成本为 $E(6)_c + E(7)_c$,玩家可在途径的村庄将资源补给到最佳状态。对比式(5-12)和(5-13)可以看出,玩家在第六天时预计挖矿几天可以覆盖资金成本很大程度上取决于当时历史记录的沙暴和高温天气发生概率及分布上的标准差,当然,我们也可以从历史数据中得出晴朗天气发生概率及分布上的标准差。

当玩家经过一次村庄后晴朗,高温,沙暴天气下挖矿取得的收益值变为 670,290,100 对应的期望收益应该修正为:

$$\overline{E(t)_r} = 670+100(\Pr(3|t)+\sigma(3|t))-380(\Pr(2|t)+\sigma(2|t)) \qquad \textbf{(5-14)}$$

根据式子(5-9),只要 $\sigma(2|t)$ 不是很大,我们可以认为不管任何天气下挖矿都能取得收益。

综合上面的论述,从图 4 可以看出玩家只在村庄(即区域 14)时面临去矿山或是直接去终点的选择,因此能够最大化收益的策略如下:

设 t_k 是玩家第 k 次到达村庄的时间点,而玩家在村庄可以将身边的资源补充到理想状态。

1. 判断剩余时间持续挖矿的收益 $\sum_{i=t_k}^{30-T(18->25)} E(i)_r$ 是否大于成本 $E(t_k)_c + E(t_k+1)_c$,若果是,接着下一条判断;如果否,则玩家选择直接到达终点。
2. 判断如果身上携带 $1200kg$ 的资源,减去村庄-矿山的来回所需要的资源重量 $2(L(t_k)_c + L(t_k+1)_{c+1})$ 的情况下剩余在矿山挖矿的天数挖矿收益能否覆盖成本 $E(t_k)_c + E(t_k+1)_c$,如果是,接着下一条判断;如果否,则玩家选择直接到达终点。
3. 计算得出挖矿收益覆盖村庄-矿山的来回成本 $2(E(t_k)_c + E(t_k+1)_c)$ 的最短天数 $t_{\min}$,跳转到步骤[1],求和初始天 $i = t_k$ 变为 $i = t_k + t_{\min}$。

前面提到,若玩家按照策略步骤确定下一天的行为,则有 99.7% 的概率能够通关并达成目标条件。

5.3 问题三:多名玩家同时穿越沙漠的模型

自然界中的蚂蚁觅食是一种群体性行为,并非单只蚂蚁自行寻找事物源。蚂蚁在寻找食物源时,会在其经过的路径上释放一种信息素,信息素的浓度大小表征路径的远近,信息素浓度越高,,表示对应的路径浓度越短,蚂蚁会以较大的概率优先选择信息素浓度较高的路径,并释放一定量的信息素,以加强该条路径上的信息素浓度,这样就形成一个正反馈。

5.3.1 第五关问题求解

对第五关,将地图抽象为如图 5 所示的图结构,蚁群算法的蚂蚁数量 $m=2$,区域数 $n=7$。第五关天气状况已知,区域 i, j 之间边的权重为其当天的花费的资金数对应 $Cost(i,j|t)$,启发式函数 $\eta_{ij}(t)$ 应该为资金话费的倒数,也即:

$$\eta_{ij}(t) = \frac{1}{Cost(i,j|t)} \qquad \textbf{(5-15)}$$

第 t 天区域 i 与区域 j 连接路径上的信息素浓度为 $\tau_{ij}(t)$。初始时刻,各个区域之间的信息素浓度相同 $\tau_{ij}(0)=1$。

玩家k根据各个区域间连接路径的信息素浓度决定其下一个访问的城市。由于玩家可以自由选择下一天的行走方向,其行走路径受启发式函数以及信息素的影响,因此可以计算出玩家k经过路径$i->j$的概率:

$$P_{ij}^k=\{\frac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s\in allow_k}[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}(t)]^{\beta}} \qquad if \quad s\in allow_k \qquad \textbf{(5-16)}$$

$$0 \qquad\qquad if \quad s\notin allow_k\}$$

如上文所述,玩家在释放信息素的同时,各个区域间路径上信息素也在逐渐消失,当所有玩家完成一次迭代循环后,各个区域之间连接路径上的信息浓度可以进行实时更新,即

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}-\frac{1}{(10-t)^{\gamma}},j\neq 终点区域$$

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij},j=终点区域$$

$$\Delta\tau_{ij}=\sum_{k=1}^{n}\Delta\tau_{ij}^k \qquad \textbf{(5-17)}$$

其中$\frac{1}{(10-t)^{\gamma}}$是对于剩余天数的惩罚函数,减去该函数是要求避免玩家超过时间限制而不选择遣返终点的路径,其中γ是大小可变的惩罚因子。

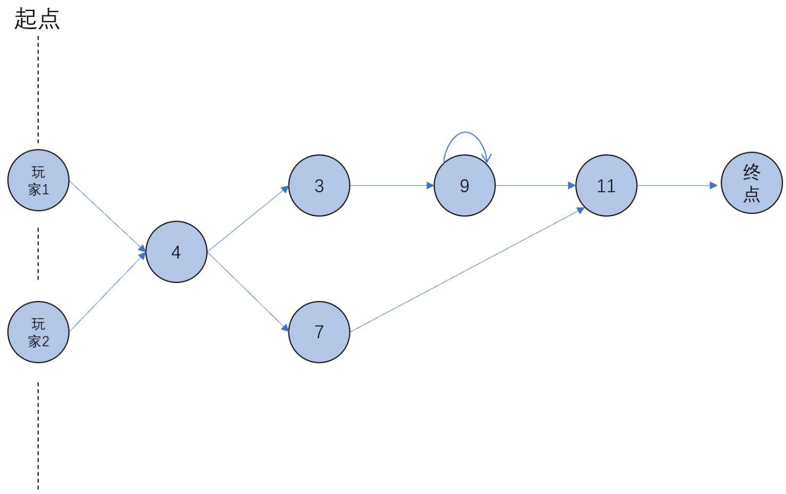


图 5.根据最短路径裁剪到了一些不相关的区域,区域 9 为矿山所在地,自相连的边说明下一天玩家仍旧在矿山挖矿。

在该题中,选择使用$ant\ cycle\quad system$模型,

$$\Delta\tau_{ij}^k=\{Q/L_k, \qquad \text{第}k\text{个玩家从区域}i\text{到达区域}j$$

$$0, \qquad \text{其它}\}$$

5.3.2 第六关问题求解

对于第六关,将地图抽象为如图6所示的图结构,与第五关不同的是启发式函数$\eta_{ij}(t)$,应引入概率模型使得模型可以描述天气随机的状况。第六关对应蚁群算法的蚂蚁数量$m=3$,区域数$n=11$。

对于第一次到达村庄补给以前的区域迁移花费的期望资金数为:

1.存在行走目的区域重复的玩家

$$Cost(i,j\,|\,t-)=kn(330+900(\Pr(3\,|\,t)+\sigma(3\,|\,t))+480(\Pr(2\,|\,t)+\sigma(2\,|\,\mathrm{t})))$$

其中n是存在目的区域重复的玩家个数,k为该关卡对行走目的区域资源消耗的惩罚因子。

2. 不存在目的区域重复的玩家

$$Cost(i,j\,|\,t-)=k(330+900(\Pr(3\,|\,t)+\sigma(3\,|\,t))+480(\Pr(2\,|\,t)+\sigma(2\,|\,\mathrm{t})))$$

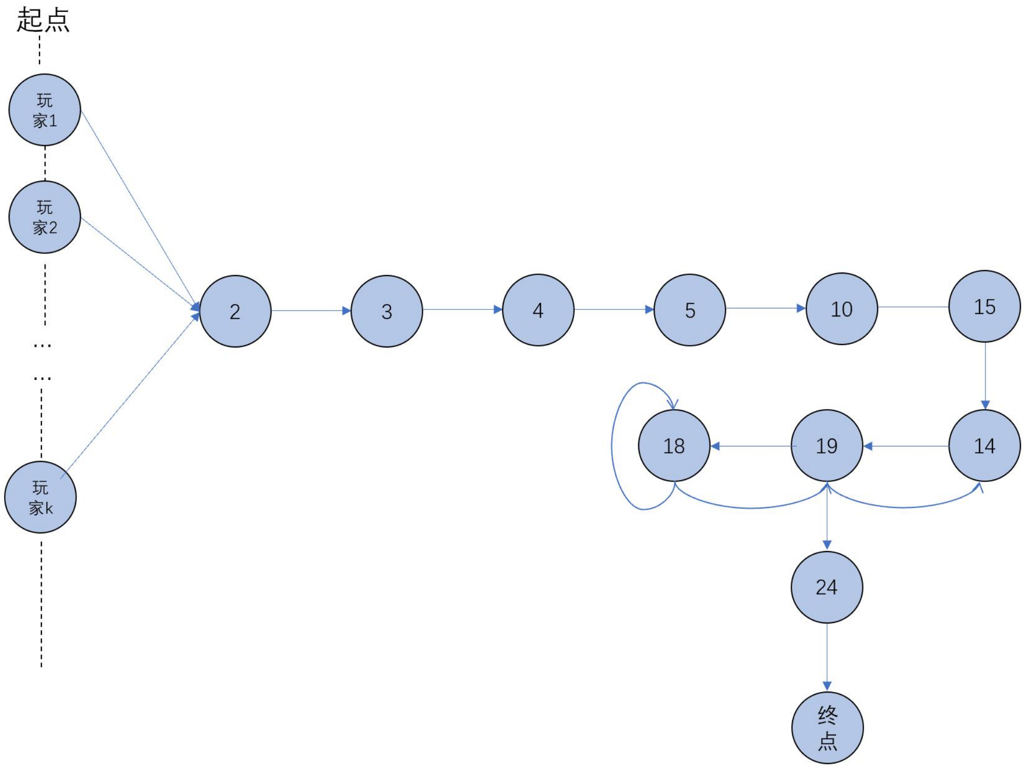


图 5.根据最短路径裁剪到了一些不相关的区域,区域 14 为矿山所在地,区域 18 为村庄所在地,自相连的边说明下一天玩家仍旧在矿山挖矿。

而经过一次村庄补给以后,补给品的价格翻了一倍,同样分为两种情况:

1.存在行走目的区域重复的玩家

$$Cost(i,j\,|\,t+)=2kn(330+900(\Pr(3\,|\,t)+\sigma(3\,|\,t))-480(\Pr(2\,|\,t)+\sigma(2\,|\,t))$$

其中n是存在目的区域重复的玩家个数,k为该关卡对行走目的区域资源消耗的惩罚因子。

2 不存在行走目的区域重复的玩家

$$Cost(i,j\,|\,t+)=2k(330+900(\Pr(3\,|\,t)+\sigma(3\,|\,t))-480(\Pr(2\,|\,t)+\sigma(2\,|\,t))$$

其余表达式跟第五关状态相同。

六、模型的评价与推广

在第一题中简单论述了基于分支限界法的组合排序算法,据此算法模拟玩家达到终点以前的路径选择问题。

马尔科夫网络的有关改进:可将马尔科夫网络转换成因子图，然后用$sum-product$算法高效求各个变量的边缘分布。

蚁群算法的有关改进:状态转移函数的作用是利用距离信息引导蚂蚁寻找最短路径。在条件较多的情况下蚂蚁搜索容易忽略掉现阶段状态的其他因素，存在盲目选择的现象，从而导致死锁，因此可以通过改进启发函数来解决或避免此类问题的发生。

参考文献:
[1] 郁磊,史峰,王辉,胡斐.MATLAB 智能算法 30 个案例分析[M].北京:北京航空航天大学出版社,2015:205-207
[2] 李宝磊,吕丹桔,张钦虎,等.基于多元优化算法的路径规划[J].电子学报,2016(9):2242-2246
[3] Puterman, Martin L. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, 2014.

附录

附录 1

最短路径算法

```
#include<iostream>
#include<queue>
#include<algorithm>
#include<vector>
using namespace std;

const int MAX=100;
int parent[MAX];
int flag[MAX];
vector<int> Map[MAX];

void setMap(){
    memset(flag,0,sizeof(flag));
    memset(parent,0,sizeof(flag));

    int x,y;
    while(cin>>x>>y&&(x!=0&&x!=0)){
        Map[x].push_back(y);
        Map[y].push_back(x);
    }
}

void print(int x){
    if(x==-1)return;
    cout<<x<<' ';
    return print(parent[x]);
}

void bfs(int x,int y){
    queue<int> que;
    que.push(x);
    flag[x]=-1;
```

```
    parent[x]=-1;
    while(!que.empty()) {
        int temp=que.front();
        que.pop();

        int size = Map[temp].size();
        for(int i=0;i<size;i++){
            int va = Map[temp][i];
            if(flag[va]==-1) continue;
            else{
                flag[va]=-1;
                parent[va]=temp;
                if (va==y)return print(va);
                que.push(va);
            }
        }
    }
    cout<<"no found"<<endl;
}

int main(){
    cout<<"输入每条边: ";
    setMap();
    cout<<"输入终点和起始点；";
    int x,y;
    cin>>x>>y;
    bfs(x,y);
}
```

附录 2

计算第一问

```
#include<iostream>
using namespace std;

const int  MAX=100;
// 定义基本量
pair<int,int>pa[10];

//构建行走和挖矿花费的两个序列
int len;
int temperature[MAX];
int c;

void setArray(){
    cout<<"请输入天数和每日气温，用 1 2 3 代表天晴 高温 沙尘暴"<<endl;
    cin>>len;
    for(int i=0;i<len;i++){
        cin>>temperature[i];
    }
}

void diff(){
    int sum=0xffffff;
    int p=0;
```

```
    for(int i=0;i<len;i++){
        int sub = c,add=0;
        int temp=0;
        while(sub--){//计算行走的花费
            int w=0,m=0;
            int flag=temperature[i+add];
            if(flag==3){//几天是沙尘暴多算一天
                w=pa[flag].first*pa[0].first+pa[flag].second*pa[0].second;
                sub++; //多计算一天
            }
            else{
                w=2*(pa[flag].first*pa[0].first+pa[flag].second*pa[0].second);
            }
            m=3*(pa[flag].first*pa[0].first+pa[flag].second*pa[0].second);
            temp+=m-w;
            add++;
        }

        if(temp<sum){
            sum=temp;
            p=i;
        }
        cout<<temp<<' ';
    }
    cout<<endl;
    cout<<"最优结果："<<sum<<endl;
    cout<<"最优组序列：";
    for(int i=p;i<p+c;i++){
        cout<<i<<' ';
    }
    cout<<endl;
}

//计算挖矿收益和需要物资
int income(){
    int rns=0, cost=0,weight=0;
    int water=0;
    int food=0;
    for(int i=0;i<len;i++){
        int flag = temperature[i];
        water+=3*pa[flag].first;
        food+=3*pa[flag].second;

        weight = water*pa[4].first+food*pa[4].second;
        if(weight>=1200-288){
            cout<<i<<' '<<weight<<endl;
            return 0;
        }
    }

    rns = len*1000-water*pa[0].first+food*pa[0].second;
    cost = water*pa[0].first+food*pa[0].second;

    //weight = water*pa[4].first+food*pa[4].second;
```

```
    cout<<"花费的水和食物数量是 "<<water<<' '<<food<<endl;
    cout<<"总共获利和花费物资重量"<<rns<<' '<<cost<<' '<<endl;
    return rns;
}

//计算行程需要物资
int route(){
    int water=0;
    int food =0;
    int ans = 0;
    int weight = 0;
    cout<<"每天的花费"<<endl;
    for(int i=0;i<len;i++){
        int flag = temperature[i];
        if(flag==3){
            water += pa[flag].first;
            food += pa[flag].second;
        }
        else {
            water+= 2*pa[flag].first;
            food += 2*pa[flag].second;
        }
        cout<<water<<' '<<food<<endl;
    }
    ans = water*pa[0].first+food*pa[0].second;
    weight = water*pa[4].first+food*pa[4].second;
    cout<<"花费的水和食物数量是 "<<water<<' '<<food<<endl;
    return ans;
}

//具体每个步骤消耗
void ans(){
    int water=0,food=0;
    int ccc;
    cout<<"请输入 1 或 2"<<endl;
    cin>>ccc;
    switch(ccc){
    case 1:for(int i=0;i<len;i++){
        int flag = temperature[i];
        if(flag==3){
            water += pa[flag].first;
            food += pa[flag].second;
        }
        else {
            water+= 2*pa[flag].first;
            food += 2*pa[flag].second;
        }
        cout<<32-water<<' '<<24-food<<endl;
    };break;
    case 2:for(int i=0;i<len;i++){
        int flag = temperature[i];
        water+=3*pa[flag].first;
        food+=3*pa[flag].second;
        cout<<86-water<<' '<<84-food<<endl;
```

```
    };break;

    }

}

int halt(){
    int water=0;
    int food =0;
    int ans = 0;
    int weight = 0;
    cout<<"每天的花费"<<endl;
    for(int i=0;i<len;i++){
        int flag = temperature[i];
        water += pa[flag].first;
        food += pa[flag].second;
        cout<<water<<' '<<food<<endl;
    }
    ans = water*pa[0].first+food*pa[0].second;
    weight = water*pa[4].first+food*pa[4].second;
    cout<<"花费的水和食物数量是 "<<water<<' '<<food<<endl;
    return ans;
}
int main(){
    cout<<"请根据需要选择对应模块"<<endl;
}
//第一组基础价格参数
//5 10
//5 7
//8 6
//10 10
//关卡 1 天气数列：1 2 2 3 2 1 2 2 2 3 3 2 2 1 1 2 1 3
```