“穿越沙漠”游戏下玩家的最优策略模型

摘要

“穿越沙漠”游戏要求玩家在规定时间内，利用手中的地图和初始资金，在起点和村庄购买物资，同时可以在矿山获得收益，每天需要根据不同的行为和天气消耗物资的规则下，合理安排行走路线、物质购买和挖矿安排以便能在终点获得最大的收益。本文根据每个问题不同的条件，建立对应的最优策略模型，并针对了具体的关卡进行了具体的分析。

针对问题一：经过分析，影响玩家决策的几个因素有持有资金、剩余水和食物、离目标地的距离和天气。本文将玩家游戏的天数、当前位置、剩余水、剩余食物设为玩家的状态，根据 *Floyd* 算法化简地图后，得出几条可能的最优路线，再利用动态规划算法，求解在不同状态下玩家在终点时持有的最大资金。最终求得第一关的最优解为 10470 元，第二关为 12730 元。

针对问题二：本文利用马尔可夫链，建立马尔可夫链天气预测模型，根据第一关的天气转换情况预测其余关卡的天气情况。再根据问题一的模型得出在不同天气状况下玩家的最佳策略，利用对应的天气情况概率和最佳策略下的最大资金，可以得出最终收益的数学期望，选择数学期望最大的一条策略即可。最后模拟了关卡三和关卡四的天气情况进行求解，经过验证，该模型是合理的。

针对问题三：对于问题三的第一小问，我们根据不同策略的收益，确定玩家选择策略的倾向。我们以玩家到达每一个点的平均收益来代替问题 1 中的实际收益S，定义点与点间转移的最大收益为势函数，通过势函数得到点与点之间的转移概率，接着通过排列组合求解点与点间的数学期望，最后可以再次利用问题 1 的方法求解到终点时数学期望最大的策略即为最佳策略。

对于问题三的第二小问，本文根据完全信息的静态博弈论建立了占优策略模型。本文定义了玩家的收益函数A，该函数与玩家的决策，剩余时间，剩余资源，以及所处的位置距离终点、村庄的距离有关。当于资源、时间充足的情况下，玩家无论选择何种策略，都不能保证是占优的，这种情况的解决根据问题三（1），利用期望收益求解。而当资源或时间不足的情况下，在参与人数超过两人时，则先考虑可以占优的一方或多方，再利用期望收益求解无法占优的参与者们。

综上所述，本文依据各题所给的条件较全面地分析了相关因素对玩家决策的影响，并给出了不同条件下玩家的最佳策略，实现了在终点的最大收益。经过分析验证，本文的模型具有合理性和一定的现实意义。

关键词：动态规划 马尔可夫链 人工势能场 博弈论

一、 问题重述

1. 问题背景

穿越沙漠的游戏规则是玩家凭借一张地图，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。

我们要在不同的游戏设定下给出玩家的最佳决策。

2. 需要解决的问题

问题一：假设只有一名玩家，在整个游戏时段内每天天气状况事先全部已知，试给出一般情况下玩家的最优策略。求解附件中的“第一关”和“第二关”，并将相应结果分别填入 Result.xlsx。

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$k(2 \le k \le n)$名玩家均从区域A行走到区域$B(B \ne A)$，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$k(2 \le k \le n)$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。

(1)假设在整个游戏时段内每天天气状况事先全部已知，每名玩家的行动方案需在第 0 天确定且此后不能更改。试给出一般情况下玩家应采取的策略，并对附件中的“第五关”进行具体讨论。

(2)假设所有玩家仅知道当天的天气状况，从第 1 天起，每名玩家在当天行动结束后均知道其余玩家当天的行动方案和剩余的资源数量，随后确定各自第二天的行动方案。试给出一般情况下玩家应采取的策略，并对附件中的“第六关”进行具体讨论。

二、 问题分析

1. 问题 1

为了研究一般情况下单人玩家的最佳策略，我们需要分析影响玩家决策的因素。如图所示：

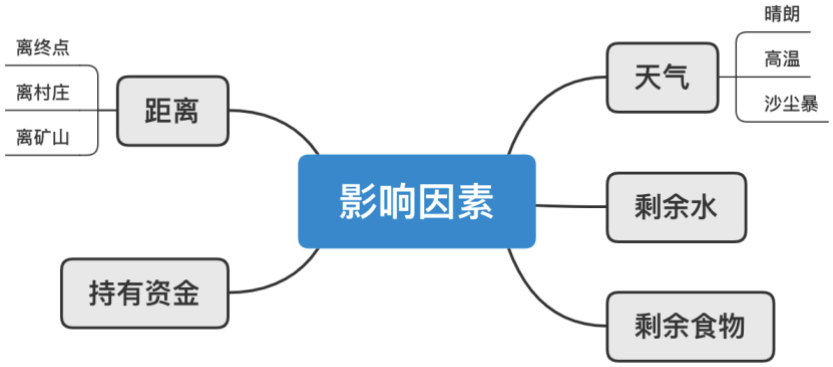

图 1 影响因素

问题一中玩家的已知条件是规定时间内的天气和地图。关于距离的因素，玩家可以先化简地图，通过 *Floyd* 算法得出起点、村庄、矿山和终点之间的最短路径从而减少多余的区域。接着可以利用动态规划算法，根据所在点、天数、水和食物的四个因素的影响递推得出在符合规则下到终点时的最大的持有资金。

2. 问题 2

第二题相比于第一题，玩家仅知道当天的天气状况，需要根据当天天气状况来决定当天的行动方案。因此我们需要首先判断能否将天气因素用其他影响玩家决策的因素来代替或者去除，即化简、整合合并决策因素。但经过初步分析，天气因素在影响玩家决策的因素中是不可或缺的。玩家需要根据未来天气情况，来做一个全局规划。

因此，我们需要考虑其他方法来解决这个问题。对于现有的相关因素——当天天气，我们可以假设当天天气可影响明天天气，我们可以使用马尔可夫链来对未来天气进行预测。且根据马尔可夫链的概率收敛特性，在给定各天气跳转的概率下，我们可以最终得到一个收敛的天气概率，并基于此实现对未来天气的预测。

获得未来天气可能情况以及其对应概率后，我们根据第一问得到对应的最佳策略情况下的收益，并得到该情况下收益的数学期望。最终收益的数学

期望最大的方案作为当前情况的方案。且随着日期的推进，玩家会逐步确定当天天气，为了使得最终决策最优，我们需要根据每天的实际天气情况动态的调整玩家策略。

3. 问题 3

（1）在有多个玩家的情况下，若按照单人模式选取最优策略，有较大可能与其他玩家走相同的路线，使得亏损更大。因此，在无法获取其他玩家选择方案的情况下，需要根据不同路线的收益，确定玩家选择策略的倾向。我们以玩家到达每一个点的平均收益来代替问题 1 中的实际收益S，将点和点转移的最大收益定为势函数，通过势函数得到点和点之间的转移概率，然后得到点和点转移的数学期望，最后就可以再次利用问题 1 的方法求解出最佳策略，该最佳策略指代的是具有最大数学期望的策略。

（2）在本题中，玩家的设定是只知道当天的天气状况，以及在当天行动结束后其他玩家的行动情况，对于玩家来说，第二天的行走路线不仅取决于路线的收益，也取决于第二天不同天气的概率以及其他玩家的决策影响。由于事先无法预知其他玩家的决策，但天气情况和地图信息是公开的，因此是完全信息的静态博弈。对于玩家选择的策略，需要考虑的因素有游戏的剩余时间，到达终点需要的资源消耗是否足够，以及村庄资源的补给，挖矿的期望收益。因此，需要为玩家寻找一个收益函数A，该函数与玩家的决策，剩余时间，剩余资源，以及所处的位置距离终点、村庄的距离有关。由游戏的规则可知，在没有与其他玩家出现挖矿冲突、路线冲突、购资源冲突时，玩家下一步的收益与其他玩家的决策无关。而在发生冲突时，对于资源、时间充足的情况下，玩家无论选择何种策略，都不能保证是占优的，因此，这种情况的解决根据 3（1），利用期望收益求解。而在问题（2）中只考虑存在占优情况下的策略选择。在参与人数超过两人时，则先考虑可以占优的一方或多方，再利用期望收益求解无法占优的参与者们。

三、 模型假设与约定

1) 问题二假设除特殊情况外，游戏中天气的转换的概率和第一关相近；
2) 问题三假设多名玩家同样聪明；
3) 问题三假设存在某一个临界资源和时间状况，使得多方参与的某些玩家具有严格占优的决策策略。

四、 符号说明及名词定义

符号	定义
$L_i^{t'\to t}$	第t'天到第t天且到达点i的消耗的水和食物的价值
$M_i^{t'\to t}$	第t'天到第t天且在点i的挖矿的收益
$S_i^{t'\to t}$	第t'天到第t天且到达点i的净收益
$G_{max}^{w,f,t,i}$	第t天在第i个点，有w箱水和f箱食物的最优解
P_{ij}	天气i转移到天气j的概率
W_m	表示标志为m的一个有序天气集合
$H_{max}^{w,f,t,i}$	第t天在点i且携带w箱水和f箱食物时的势函数
$P_{i,j}^t$	第t天，玩家从点i到其邻点j的转移概率
$E_{i,j}^t$	第t天，玩家从点i到其邻点j的净收益的数学期望
$A_i(s)$	玩家i下一步行动的总收益函数

五、 模型的建立与求解

1 问题一

1.1 问题一模型的建立

1.1.1 最简地图模型的建立

为了简化问题和减少后面利用动态规划求解问题的复杂度，我们可以先将地图转换为点和线的关系，即一个点代表一个区域，而相邻的区域用线连接表示可以到达。

之后我们可以继续简化地图，找出起点、村庄、矿山和终点两两之间的最短路径。求最短路径的问题我们可以先用 *Floyd* 算法求解。

假设从节点i到节点j的最短距离为$d_{i,j}$，而$d_{i,j}$不外乎两种可能：

1）从节点i直接到节点j；

2）从节点i经过若干个节点k到节点j。

则我们需要判断：

$$d_{i,k}+d_{k,j}<d_{i,j}$$

其中k为图内所有节点。若上式中成立，则证明是上述的第二种可能，那么$d_{i,j}$重置为$d_{i,k}+d_{k,j}$，遍历完所有节点k后，$d_{i,j}$即为所求最短距离。

求出最短距离后，只将起点、村庄、矿山和终点等目标地留下，线的权重的意义为从地点i到地点j需要t天。

最后将多余的线可以删去，例如若矿山->村庄->终点的距离和矿山->终点的距离相等，则我们可以将矿山->终点的这条线去掉，简化图。

1.1.2 基于动态规划的最佳策略模型的建立

在本题中，我们的最终目标是在物资充足的前提下在规定时间内到达终点且资金最大。而这个目标的复杂度太高，可以采用分治的思想，将大问题化解为小问题进求解，即用动态规划求解。

1）水和食物的消耗

依据题意我们可以得到在第t天玩家水的基础损耗D_1^t为:

$$D_1^t=\begin{cases}5 & 第t天为晴天\\ 8 & 第t天为高温\\ 10 & 第t天为沙尘暴\end{cases} \tag{1}$$

同理在第t天玩家食物的基础损耗D_2^t为:

$$D_2^t=\begin{cases}7 & 第t天为晴天\\ 6 & 第t天为高温\\ 10 & 第t天为沙尘暴\end{cases} \tag{2}$$

而考虑到玩家的行为因素，除了需要考虑在矿山的停留时间外，在前往目标地点途中无需停留，因此玩家其实只有三种行为对应三个基础消耗的倍数：在沙尘暴时停留、行走、挖矿。即：

$$\sigma^k=\begin{cases}1 & k=在沙尘暴时停留\\ 2 & k=行走\\ 3 & k=挖矿\end{cases} \tag{3}$$

接着我们可以推出经过t天在第i个点，有w箱水和f箱食物时水的损耗为

$$w_{cost}{}^{w,f,t,i}=w_{cost}{}^{w',f',t',i}+\Delta w^{t'\to t} \tag{4}$$

其中$\Delta w^{t'\to t}$为从第t'天到第t天水的损耗，满足

$$\Delta w^{t'\to t}=\sum_{k=t'}^{t}D_1^t\times\sigma^k \tag{5}$$

同理，可以推出第t天在第i个点，有w箱水和f箱食物时食物的损耗为

$$f_{cost}{}^{w,f,t,i} = f_{cost}{}^{w',f',t',i} + \Delta f^{t'\to t} \tag{6}$$

其中$\Delta f^{t'\to t}$为从第t'天到第t天食物的损耗，满足

$$\Delta f^{t'\to t} = \sum_{k=t'}^{t} D_2^t \times \sigma^k \tag{7}$$

2）水和食物的损耗价值$L_i^{t'\to t}$

根据题目所给的信息，在村庄所购的水和食物的价格为在起点的两倍，则我们可以列出在第t'天到第t天且到达点i的消耗的水的价值为：

$$W_i^{t'\to t} = \begin{cases} \Delta w^{t'\to t} \times w_{price}, & w_{cost}{}^{w',f',t',i} - w_0 \le \Delta w^{t'\to t} \\ \left(2\Delta w^{t'\to t} + w_0 - w_{cost}{}^{w',f',t',i}\right) \times w_{price}, & \Delta w^{t'\to t} < w_{cost}{}^{w',f',t',i} - w_0 < 0 \\ \Delta w^{t'\to t} \times 2 \times w_{price}, & w_{cost}{}^{w',f',t',i} - w_0 \ge 0 \end{cases} \tag{8}$$

同理可得第t'天到第t天且到达点i的消耗的食物的价值：

$$F_i^{t'\to t} = \begin{cases} \Delta f^{t'\to t} \times f_{price}, & f_{cost}{}^{w',f',t',i} - f_0 \le \Delta f^{t'\to t} \\ \left(2\Delta f^{t'\to t} + f_0 - f_{cost}{}^{w',f',t',i}\right) \times f_{price}, & \Delta f^{t'\to t} < f_{cost}{}^{w',f',t',i} - f_0 < 0 \\ \Delta f^{t'\to t} \times 2 \times w_{price}, & f_{cost}{}^{w',f',t',i} - f_0 \ge 0 \end{cases} \tag{9}$$

则总消耗价值为：

$$L_i^{t'\to t} = W_i^{t'\to t} + F_i^{t'\to t} \tag{10}$$

3）挖矿收益$M_i^{t'\to t}$

根据题目所得，每天的挖矿收益为$m = 3 \times m_basic$，则第t'天到第t天且在点i的挖矿收益为：

$$M_i^{t'\to t} = \begin{cases} 0 & \textit{不挖矿时} \\ m \times (t - t' + 1) & \textit{挖矿时} \end{cases} \tag{11}$$

4）净收益$S_i^{t'\to t}$

根据题意和上述式子可得，$S_i^{t'\to t}$由两部分构成，一是水和食物消耗的价值，二是由挖矿所得的资金，即：

$$S_i^{t'\to t} = -L_i^{t'\to t} + M_i^{t'\to t} \tag{12}$$

5）资金最优解

我们可以假设第t天在第i个点，有w箱水和f箱食物的最优解为$G_{max}^{w,f,t,i}$，而且满足：

$$G_{max}^{w,f,t,i} = max\left\{G_{max}^{w',f',t',ik} + S_{ik}^{t'\to t}\right\} \tag{13}$$

其中点ik包括点i的所有邻点和其本身，$S_{ik}^{t'\to t}$为从第t'天到第t天且到达点ik的净收益。

6）约束条件

$$s.t.\begin{cases} \Delta w > 0 \\ \Delta f > 0 \\ w\_weight \times w\_cost + f\_weight \times f\_cost \le max\ \_load \\ t \le set\_time \end{cases}$$

1.2 问题一的求解

1.2.1 最简地图模型的求解

以下我们以第一关的地图为例化简。

1）将地图变形为点和线的图。

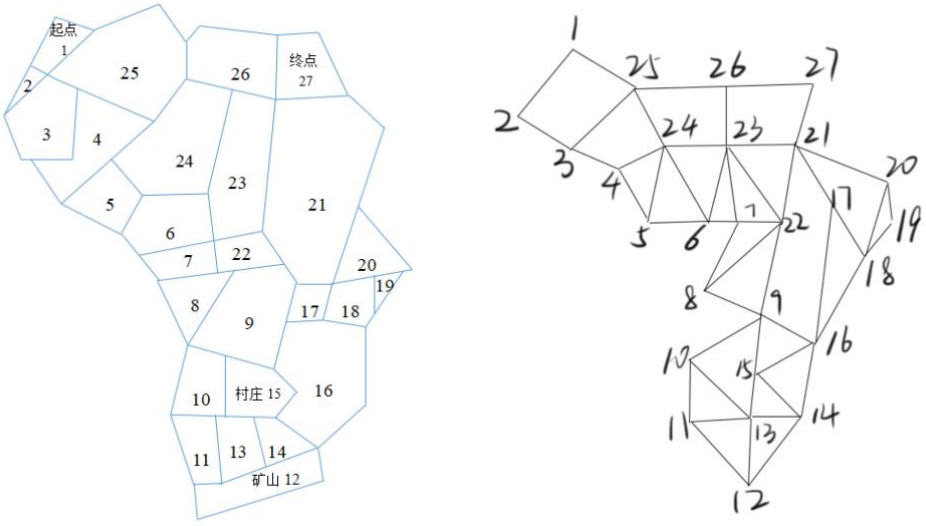

图 2 第一关地图转化为点图

2）根据 *Floyd* 算法求出起点、村庄、矿山和终点两两之间的最短路径，然后摒弃无关点。

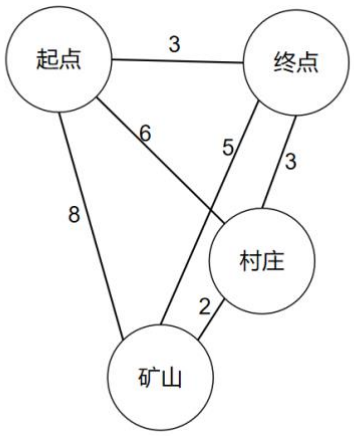

图 3 第一关地图初步化简

3）分析可以去除的多余的线，最后第一关的地图可以化简如下图所示：

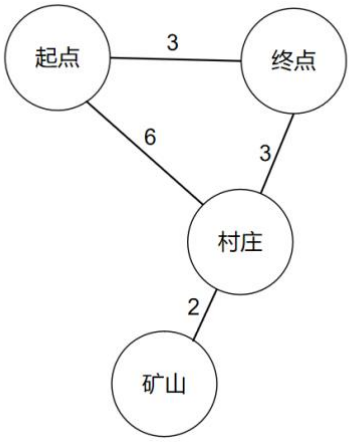

图 4 第一关地图最终化简

同理，我们可以得到第二关地图的化简:

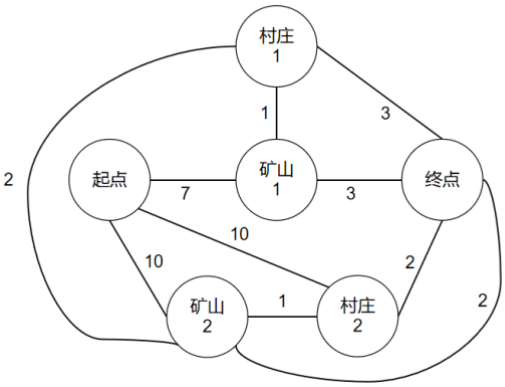

图 5 第二关地图的最终化简

1.2.2 基于动态规划的最佳策略模型的求解

根据上述模型可以对各关进行模拟仿真（代码于附录），得到各关结果如下：

1）第一关路线图如下，得到终点的最大收益为 10470 元。

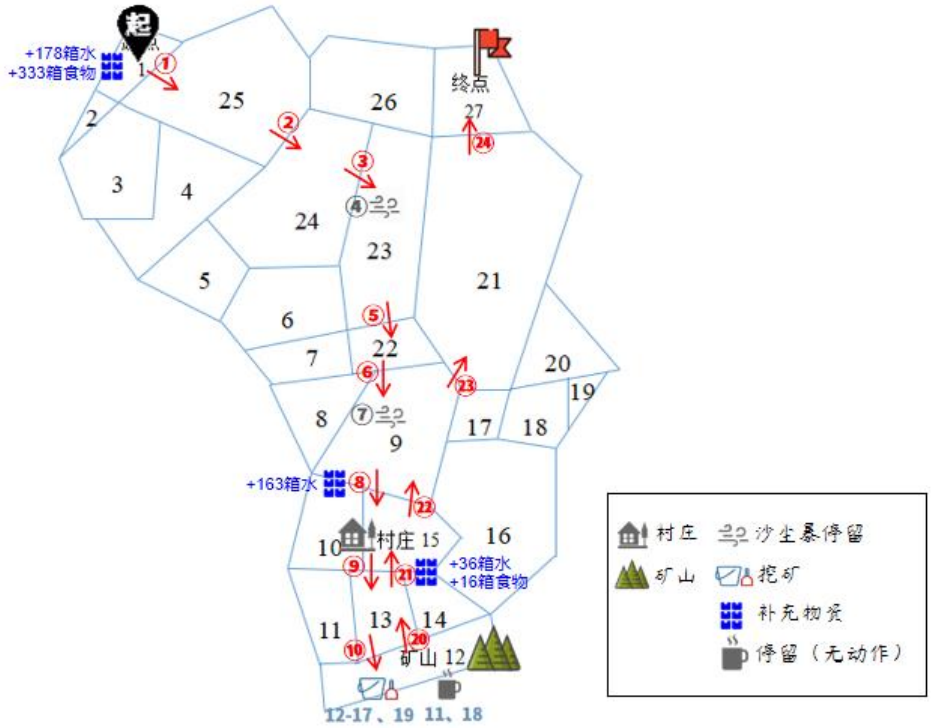

图 6 第一关路线图

2）第二关路线图如下所示，到终点的收益为 12730 元。

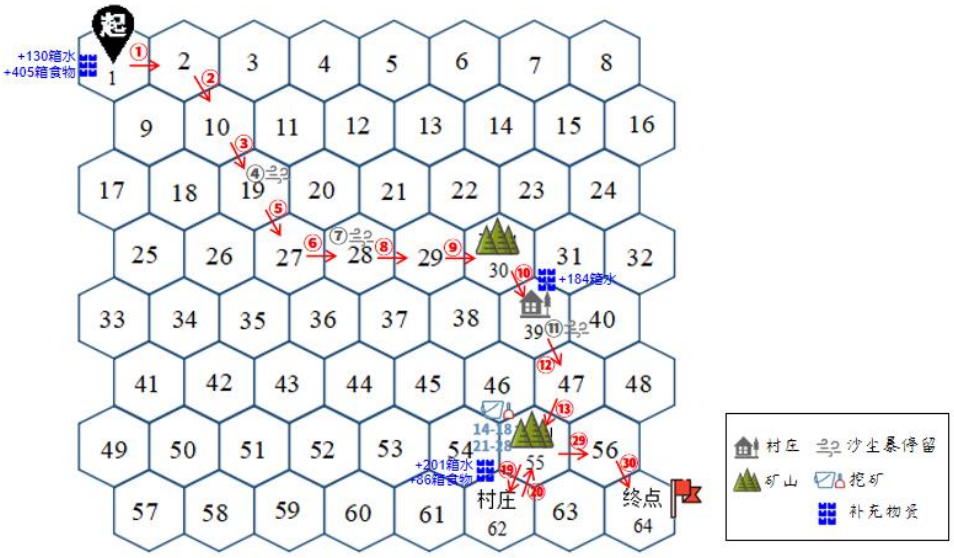

图 7 第二关路线图

2 问题二

2.1 问题二模型的建立

从全局来看，全局的天气因素能够影响玩家的全局策略；单从一天的天气来看，若玩家仅知晓一天的天气，他可以先规划一个大致的行走目标，然后根据每天的天气以及预测未来天气来及时的调整自己的策略。

在一般情况下，未来天气状况不确定，因此我们需要根据当天天气来预测未来天气状况。为了使得模型更好的拟合通用的游戏情况，我们将根据关卡一、关卡二的天气转换来分析各个天气状态相互跳转以及维持原天气的概率。再根据每个关卡不同的天气要求及时调整天气状态转换情况。

已知关卡一、关卡二天气一致，30 天天气状况表格如下：

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

根据现有条件，我们发现此问题的天气概率预测满足以下几个条件：

1)可能的天气数是有限的，包含晴朗、高温、沙尘暴。

2)从任意状态能够转变到任意状态。

3)天气状态的跳转不是一个简单的循环。

4)我们可以利用上述天气跳转表格给定各个天气之间的转移概率。

至此，该问题的分析满足马尔可夫链收敛的四大条件。因此，我们选择马尔可夫链模型进行未来天气的预测。

2.1.1 马尔可夫链天气预测模型的建立

马尔可夫链可以描述为一个随机跳跃的序列。对于游戏中的天数而言，这个跳转序列仅包含有限个离散的位置或状态的集合，且随机变量w_n——第n天的天气状态在有限的天气离散集中取值。则有：

$$w_n \in \{1,2,3\}$$

其中 1 代表晴朗天气，2 代表高温天气，3 代表沙尘暴天气。

根据关卡一、二 30 天天气状况表格，通过计算，我们可以得到各个天气之间在 30 天内的跳转次数如下：高温到高温有 6 次、高温到晴朗有 2 次、高温到沙尘暴有 3 次、晴朗到晴朗有 2 次、晴朗到高温有 4 次、晴朗到沙尘暴有 2 次、沙尘暴到沙尘暴有 1 次、沙尘暴到高温有 3 次、沙尘暴到晴朗有 2 次。则可以得出天气状态转移概率如下：

$$P_{ij}=\begin{bmatrix}P_{11} & P_{12} & P_{13}\\ P_{21} & P_{22} & P_{23}\\ P_{31} & P_{32} & P_{33}\end{bmatrix}=\begin{bmatrix}\frac{1}{4} & \frac{1}{2} & \frac{1}{4}\\ \frac{5}{14} & \frac{3}{7} & \frac{3}{14}\\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6}\end{bmatrix}$$

对应的转换图为：

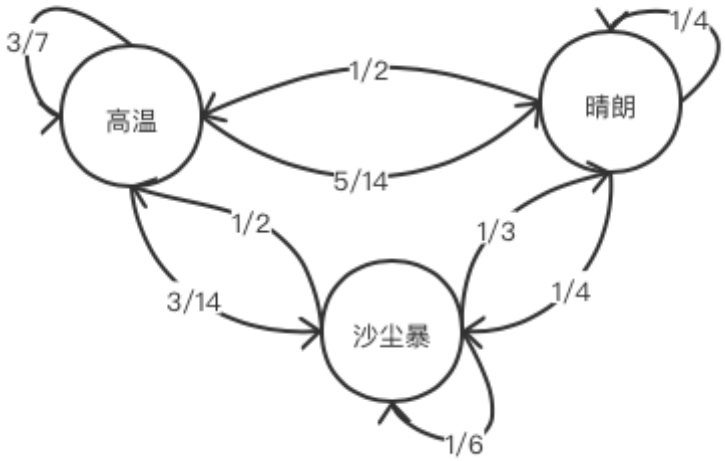

图 8 天气跳转问题的状态转移图

对于计算$w_{n+1}=j$的概率$P_r\{w_{n+1}=j\}$，满足：

$$P_r\{w_{n+1}=j\}=\sum_i P_{ij}P_r\{w_n=i\} \tag{14}$$

在第$n+1$天到达天气状态j的途径只能是在第n天到达天气状态i，然后从i跳转到j，因此，令$\pi_n(i)=P_i\{w_n=j\}$，我们可以将式 14 表达为：

$$\pi_{n+1}(j)=\sum_i P_{ij}\pi_n(i) \tag{15}$$

即为：

$$\pi_{n+1}=P\pi_n$$

根据马尔可夫链收敛规则，我们可以设置任意合理的初始情况，迭代运行式 15，都可以得到一个收敛的值，该值即为各个天气状况的概率。

至此，我们利用马尔可夫链求解出了各个天气状态的出现概率。在游戏中，玩家已知当天情况，但不知道未来天气状况。利用上述马尔可夫链模型所得结论，我们可以通过当天天气情况来预测直到游戏结束的未来的天气。不同的预测结果都对应着一个确切的概率。玩家可以根据预测概率情况结合问题一：已知全局天气情况下的决策选择模型来动态的选择自己的决策。

这种决策方法要求玩家能够根据实际情况“随机应变”，而不是固守游戏一开始时定下的策略。给定“历史”，玩家每一个行动选择开始至游戏结束都构成了一个博弈。我们将游戏的“初始状态”设定跟随玩家的选择而变化，即每一次决策后都进入一个新的将前一状态作为当前博弈初始状态的决策。

2.1.2 决策收益模型的建立

根据问题一所得的模型，我们可以将不同天气状态下获得的决策收益C表示为一个决策选择集如下：

$$\{C_{W_1}, C_{W_2}, C_{W_3}, \cdots, C_{W_m}\}$$

其中W_m表示标志为m的一个有序天气集合，C_{W_m}表示在W_m天气集情况下的最优决策所获得的收益。

对于W_m而言，我们将其存在的概率表示为P_{W_m}，则最佳决策的数学期望E_{max}满足：

$$E_{max} = max\{P_{W_1}C_{W_1}, P_{W_2}C_{W_2}, \cdots, P_{W_m}C_{W_m}\} \tag{16}$$

2.2 利用问题二模型的具体求解

根据我们的模型可画出求解流程图如下：

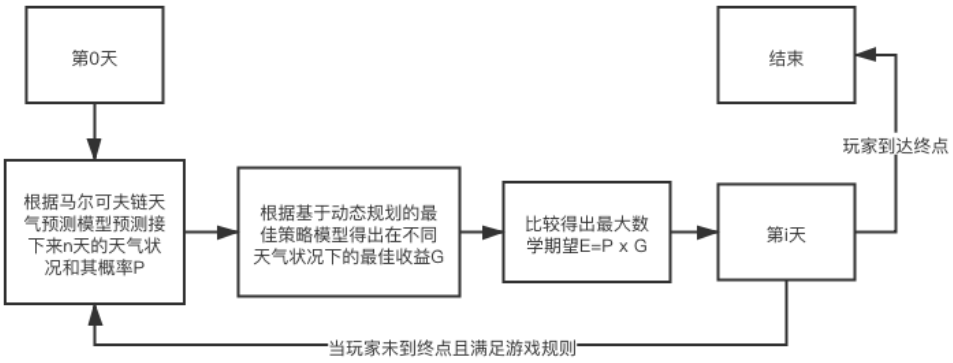

图 9 问题二求解流程图

2.2.1 关卡三求解

根据马尔可夫链收敛规则，我们可以设置任意合理的初始情况，迭代运行式 15，都可以得到一个收敛的值，该值即为各个天气状况的概率在这里我们设置初始情况为$\pi_1 = (0.5,0.4,0.1)$，迭代计算 10 次结果如下：

表 1 各天气概率收敛过程

迭代次数	晴朗	高温	沙尘暴
1	0.301190	0.471429	0.227381
2	0.319459	0.466327	0.214215
3	0.317815	0.466691	0.215494
4	0.317960	0.466665	0.215375
5	0.317948	0.466667	0.215386
6	0.317949	0.466667	0.215385
7	0.317949	0.466667	0.215385
8	0.317949	0.466667	0.215385
9	0.317949	0.466667	0.215385
10	0.317949	0.466667	0.215385

根据关卡三的天气要求：玩家仅知道当天的天气状况，但已知 10 天内不会出现沙暴天气。根据上述分析所得图 8 各天气状态跳转图，我们由关卡天气条件删去天气状态为沙尘暴的天气：

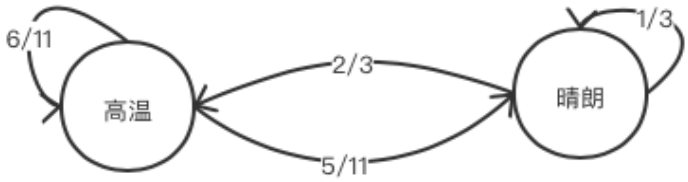

图 10 关卡三的天气跳转问题的状态转移图

接下来我们可以根据 *Floyd* 算法将关卡三的地图化简为以下形式：

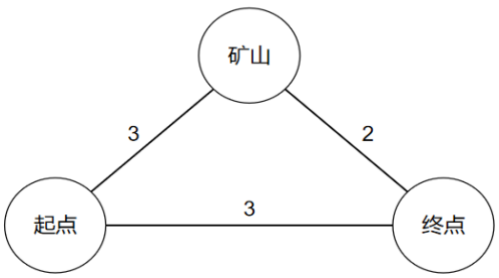

图 11 关卡三抽象地图

其次，根据图 9 的问题分析流程图，我们需要先获得各个天气情况下对应的最佳策略。通过以上天气跳转概率，我们列举了所有符合以上天气跳转概率的 10 天天气状况。运用模型一将关卡三的基础收益、基础消耗量、游戏天数等具体设定转化成对应数值带入到约束方程中。运行并分析得到的各个天气状态下的最优策略结果。

我们发现获得的最优决策路线大致分为以下两条:

$$①\text{起点}\xrightarrow{3}\text{终点}$$
$$②\text{起点}\xrightarrow{3}\text{矿山}\xrightarrow{2}\text{终点}$$

其中，路线②对应的天气情况较为特殊，为后七天均为晴朗的天气条件。其余天气状况下分析所得的最右路线均为路线①。部分运行结果整理后列举如下：

表 2 运行结果

日期 / 天气集	1	2	3	4	5	6	7	8	9	10	路线选择
W1	高温	高温	高温	晴朗	高温	晴朗	高温	晴朗	晴朗	高温	①
W2	高温	晴朗	晴朗	晴朗	高温	晴朗	高温	晴朗	晴朗	高温	①
W3	高温	晴朗	高温	晴朗	高温	晴朗	高温	高温	晴朗	高温	①
W4	晴朗	高温	高温	晴朗	高温	晴朗	晴朗	晴朗	高温	晴朗	①
W5	高温	高温	高温	晴朗	晴朗	高温	高温	高温	晴朗	高温	①
W6	高温	高温	高温	高温	高温	晴朗	高温	晴朗	高温	高温	①
W7	晴朗	高温	高温	晴朗	晴朗	晴朗	晴朗	晴朗	晴朗	高温	①
W8	高温	晴朗	高温	晴朗	晴朗	晴朗	晴朗	晴朗	晴朗	晴朗	②
W9	高温	高温	高温	晴朗	晴朗	晴朗	晴朗	晴朗	晴朗	晴朗	②
…	…	…	…	…	…	…	…	…	…	…	…

为验证根据模型一给定天气得到的决策结果的合理性，我们将结合关卡三进行具体分析。

对于路线②而言，路程消耗为 5 天，则玩家最多在矿山挖 5 天。由于这种路线选择相较路线一复杂，我们先分析路线②的收益状况。由关卡设置条件易得，玩家仅在晴朗天气下挖矿一次才有 35 元的净收益，高温天气下挖矿收益为负，因此玩家在高温下肯定选择不挖矿。

若玩家挖矿挖满 5 天，我们设路线②净收益为E_2，则有：

$$E_2 = 35d + E_1 - 135 \times (5 - d) - b \tag{17}$$

其中：d指挖矿五天中天气为晴朗的天数，b表示最后两天回终点的消耗，

E_1为路线①的净收益（路线①无挖矿，净收益为负），同路线②前三天消耗。

若路线②比路线①更优，则有：

$$E_2 > E_1$$

即：

$$35d - 135(5-d) - b > 0$$

将b取最小值 110 元——两天都在晴朗的天气下赶路，则要使路线②有收益，最好情况下d取值应大于 4.617。

由于$d \leq 5$故要使路线②收益大于路线①，则在最好情况下需要挖矿 5 天都为晴朗，即 10 天中的后 7 天都要为晴朗。与模型一分析结果相同。

下面我们将模拟玩家在关卡三起点时，用上述模型进行的决策分析过程。根据图 10，关卡三的天气跳转的状态转移图，我们可以计算出连续 7 天晴朗的概率为：0.001，则两个决策的收益期望为：

$$\{P_1C_1, P_2C_2\} = \{E_1, E_1 + (0.01 \times 65 + 0.99 \times \partial)\} \quad (18)$$

其中$\partial \leq -105$，表示次好情况，即五天中仅一天为高温时后七天的收益。

易得线路①的收益期望大于收益二线路期望，玩家选择线路①更佳。

根据图 10 中两种天气状态的跳转概率，我们可以给出符合此跳转规律的可能的 10 天游戏天气状态表格。我们将其中一种情况列举如下：

日期	1	2	3	4	5	6	7	8	9	10
天气	高温	晴朗	晴朗	高温	高温	高温	晴朗	高温	高温	晴朗

为了细化分析，我们使用以上天气状态表格作为关卡三这 10 天真实的天气情况，并选择路线一作为玩家初始策略。后面玩家再重复上述过程迭代分析出每一步的策略即可。

关卡三过程模拟结果如下：

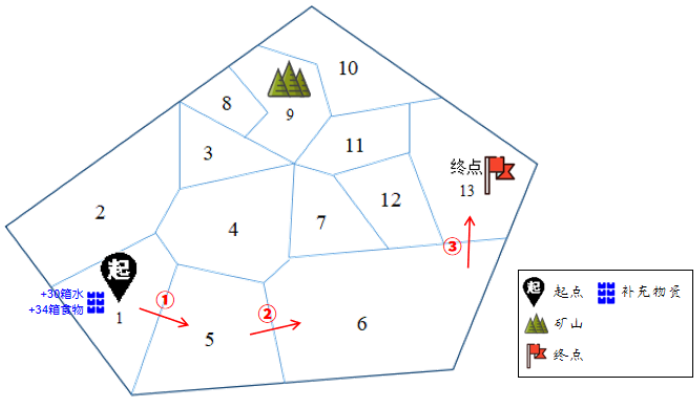

图 12 第三关路线图

根据模拟结果，由于总水量和食物量消耗为$158kg < 1200kg$，故玩家可

以在起点购买好 30 箱水以及 34 箱食物，后直达终点即可。

2.2.2 关卡四求解

与关卡三分析步骤相似，首先我们查看关卡四天气条件：玩家仅知道当天的天气状况，但已知 30 天内较少出现沙暴天气。因此，根据图 8 天气跳转问题的状态转移图，我们适当减少各个天气到沙暴天气的跳转概率。减少后的状态跳转图如下：

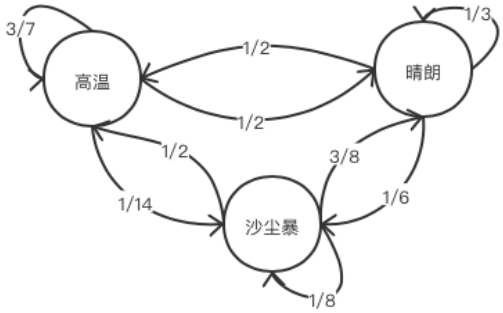

图 13 关卡四的天气跳转图

设置初始值，利用式 15，我们求得，最终收敛后的各个天气概率表示如下，可以发现，当前沙暴天气出现概率较小，达到了 30 天内较少出现沙暴天气的模拟条件：

$$\pi = (0.4160, 0.4667, 0.1173)$$

接下来，我们再根据问题一中的方法简化关卡四地图如下：

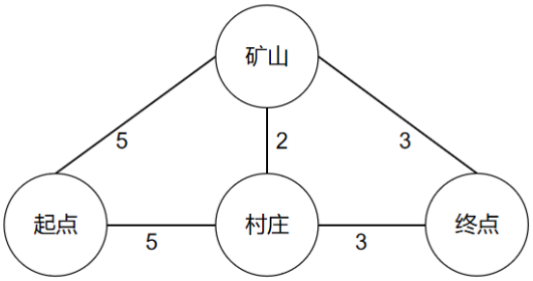

图 14 关卡四化简路线

我们假设未来天气有m种情况，其概率表示集合为：

$$\{P_{W_1}, P_{W_2}, \cdots, P_{W_m}\}$$

每种天气我们都可以利用模型一求解得到对应的一种最佳选择策略。计算各个决策的收益，根据式 16，得出最大收益期望。最终在每一点我们选择收益期望最大的决策即可。

运行算法我们可以得到，在初始情况下（即玩家位于起点，且仅知道当

天的天气）在所有路线中期望最大的一条路线为：

$$\text{起点} \xrightarrow{5} \text{村庄} \xrightarrow{2} \text{矿山} \xrightarrow{2} \text{村庄} \xrightarrow{2} \text{矿山} \xrightarrow{3} \text{终点}$$

则此时，玩家在起点将按照这个策略进行第一天的操作。接下来重复以上分析步骤，进行分析判断，直至玩家走完全程。

根据图 16 关卡四的天气跳转图，为了细化问题分析，我们假设 30 天的天气状况如下：

日期	1	2	3	4	5	6	7	8	9	10
天气	高温	高温	晴朗	高温	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	晴朗	高温	晴朗	高温	高温	沙暴	高温	晴朗	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

玩家在起点采用上述路线的策略，并在到达新的一天后迭代更新下一步策略。最终，在上述假设的真实天气情况下，玩家关卡四游戏过程如下：

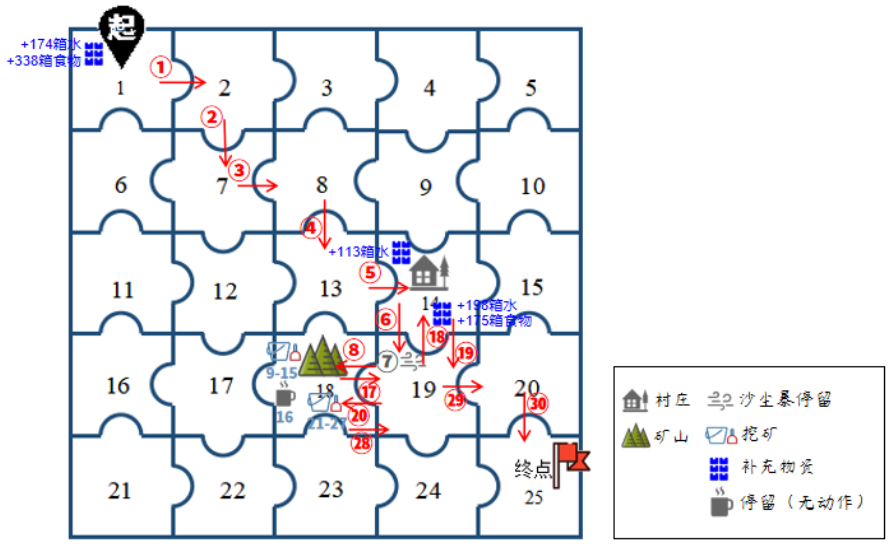

图 15 关卡四路线图

由模拟结果可得最终可以获得的收益为 13140 元。

3 问题三

3.1 问题三模型的建立

3.1.1 基于人工势能场最佳策略模型的建立

和第一第二问不同的是，第三问的游戏属于多人游戏，且玩家之间会产生影响。在第一小问中我们可以以玩家到达每一个节的平均收益来代替问题 1 中的实际收益G，将点和点转移的最大收益定为势函数，就可以得到点和点之间的转移概率，然后得到点和点转移的数学期望，最后就可以再次利用问题 1 的方法求解出最佳策略，该最佳策略指代的是玩家选择路线的最大数学期望。

我们需要求解出任意一点i在t时刻到它的相邻节点j的转移概率，从而计算出在第t天从i到j的概率（已知在第t天，节点i到节点i的概率为 1）我们利用不同时间t时，节点i到达终点的最大收益，作为该点的势函数。并根据相邻节点的势函数，确定对于处于i点的玩家到达相邻节点 j 的转移概率。

此后，计算求解出第t天下，玩家从i走到j的概率，据此根据概率计算出t时刻有n位玩家从i到达j的概率，求解出数学期望，进而通过最大的平均期望筛选出最佳的策略。

我们可以先定义第t天在点i且携带w箱水和f箱食物时的势函数为$H_{max}^{w,f,t,i}$，且满足：

$$H_{max}^{w,f,t,i} = G_{max}^{w,f,t,i} \tag{19}$$

其中$G_{max}^{w,f,t,i}$为问题一所求该状态下到达终点的最大收益。

于是我们可以得到在第t天，从点i到其邻点j的转移概率$P_{i,j}^{t}$为:

$$P_{i,j}^{t} = \frac{H_{max}^{w',f',t,j}}{\sum_{q=1}^{n} H_{max}^{w',u',t,j}} \tag{20}$$

接着可以将式 20 写成如下：

$$P_{i,j}^{t} = \sum_{k=1}^{n} P_{k,i}^{t-1} \times P_{i,j}^{t-1} \tag{21}$$

其中点k为点i的某个邻点，n为点i邻点的个数。

根据问题一所建立的模型和数学期望的公式可以得到第t天，玩家从点i到其邻点j的水和食物损耗的数学期望$E_{i,j}^{t}(L)$为：

$$E_{i,j}^{t}(L)=L_{i,j}^{t}\left[C_{n}^{n}\left(P_{i,j}^{t}\right)^{n}n+C_{n}^{n-1}\left(P_{i,j}^{t}\right)^{n-1}\left(1-P_{i,j}^{t}\right)^{1}(n-1)+\cdots+C_{n}^{1}\left(P_{i,j}^{t}\right)^{n}\left(1-P_{i,j}^{t}\right)^{n-1}+\left(1-P_{i,j}^{t}\right)^{n}\right] \quad (22)$$

其中$i\neq j$

同时，可以求出第t天，玩家从点i（点i为矿山所在处）的挖矿所得资金的数学期望$E_{i,j}^{t}(M)$为：

$$E_{i,j}^{t}(M)=L_{i,j}^{t}\left[C_{n}^{n}\left(P_{i,j}^{t}\right)^{n}\frac{1}{n}+C_{n}^{n-1}\left(P_{i,j}^{t}\right)^{n-1}\left(1-P_{i,j}^{t}\right)^{1}\frac{1}{(n-1)}+\cdots+C_{n}^{1}\left(P_{i,j}^{t}\right)^{n}\times\left(1-P_{i,j}^{t}\right)^{n-1}+\left(1-P_{i,j}^{t}\right)^{n}\right] \quad (23)$$

其中$i=j=$*矿山所在点*

根据题意，也可得每次去村庄购买的金额的数学期望$E_{i,j}^{t}(V)$为：

$$E_{i,j}^{t}(V)=V\left[4\left(1-C_{n}^{1}\times\left(P_{i,j}^{t}\right)^{n}\times\left(1-P_{i,j}^{t}\right)^{n-1}-\left(1-P_{i,j}^{t}\right)^{n}\right)+C_{n}^{1}\times\left(P_{i,j}^{t}\right)^{n}\times\left(1-P_{i,j}^{t}\right)^{n-1}+\left(1-P_{i,j}^{t}\right)^{n}\right]$$

化简得

$$E_{i,j}^{t}(V)=V\left[4-3\left(C_{n}^{1}\times\left(P_{i,j}^{t}\right)^{n}\times\left(1-P_{i,j}^{t}\right)^{n-1}+\left(1-P_{i,j}^{t}\right)^{n}\right)\right] \quad (24)$$

其中$j=$*村庄所在点*

最后，第t天，玩家从点i到其邻点j的净收益$E_{i,j}^{t}$的数学期望为：

$$E_{i,j}^{t}=E_{i,j}^{t}(M)-E_{i,j}^{t}(L)-E_{i,j}^{t}(V) \quad (25)$$

3.1.2 基于博弈论的占优策略模型模型的建立

根据题意，所有玩家仅知道当天的天气状况，从第 1 天起，每名玩家在当天行动结束后均知道其余玩家当天的行动方案和剩余的资源数量。而当玩家所处位置没有人时，玩家的决策不受他人影响，只有当玩家和他人在同一点或抵达同一点时才需要考虑。

根据问题二，我们可以得到第二天的天气概率$P=[P_1,P_2,P_3]$。在不考虑多人的情况下，对应的天气从点i到其邻点j的收益为$L=[L_1,L_2,L_3]$，则其收益的数学期望为

$$E=P_1L_1+P_2L_2+P_3L_3$$

在多人情况下，为了衡量玩家收益并选择最优策略，我们先定义一个决策空间S，满足$S=[s_1,s_2,\cdots s_m]$，m为玩家一天的决策总量。

接下来我们考虑该策略下的剩余时间t对收益函数的影响。当到达下一个点后的剩余天数和到终点所需花费到时间相距越小时，玩家对路线的选择变得会苛刻，即非最短路径的选择损失会更大。我们设这对收益函数的影响为$g_1(s,t)$，其满足关于到达终点后的剩余时间$t(s)$的负指数分布，如下所示：

$$g_1(s,t)=\alpha_1 e^{-t(s)} \tag{26}$$

其中α_1为时间系数，当$t(s)$越大，g_1越大，说明若执行这个决策s，g_1的损失很大，即迟迟不去终点，关于时间的损耗会以指数级变大。

然后我们考虑该策略下剩余的水和食物对收益函数的影响，若资源足以到达终点，村庄和最优路线对决策的影响不大；但当资源不足以到达终点时，村庄和最优路线对决策的影响会变很大。我们假设该影响函数分别为$g_2(s,w)$和$g_3(s,f)$。

$$g_2(s,w)=\alpha_2\left(e^{-w_1(s)}-e^{-w_2(s)}\right) \tag{27}$$

其中α_2为水系数，$w_1(s)$代表到终点的剩余水的箱数，$w_2(s)$为到最近村庄的剩余水的箱数，分析同上。

同理，可以定义$g_3(s,f)$满足：

$$g_3(s,f)=\alpha_3\left(e^{-f_1(s)}-e^{-f_2(s)}\right) \tag{28}$$

其中α_3为食物系数，$f_1(s)$代表到终点的剩余食物的箱数，$f_2(s)$为到最近村庄的剩余水的箱数，分析同上。

接着我们定义一个停留决策的惩罚为$g_4(s)$满足：

$$g_4(s)=\alpha_4 \tag{29}$$

α_4为停留系数，当选择停留时可能他人也选择停留，则损失了时间、资源，同时进入下一轮决策。

最后我们可以假设玩家i下一步行动的总收益函数为$A(s)$：

$$A_i(s)=N(s)E(s)+\frac{M(s)}{N(s)}+g_1(s,t)+g_2(s,w)+g_3(s,f)+g_4(s) \tag{30}$$

其中$N(s)$代表选择该决策的玩家总数，$E(s)$代表选择该决策的期望花费，$M(s)$代表选择该决策的挖矿收益。

计算出收益函数后，对上述情况进行讨论，当所有玩家对物资和时间充

足时，参与者的决策没有绝对占优的情况，因此只能通过问题 3（1）的建模思想，选择具有最小期望损失的方案。

所以我们这里只讨论有一方的物资或时间充足的情况，即对于决策空间S，参与者$A=[A_1,A_2]$，满足：

$$\begin{bmatrix} A_1(S_1),A_2(S_1) & A_1(S_1),A_2(S_2) & \cdots & A_1(S_1),A_2(S_{m-1}) & A_1(S_1),A_2(S_m) \\ A_1(S_2),A_2(S_1) & A_1(S_2),A_2(S_2) & & A_1(S_2),A_2(S_{m-1}) & A_1(S_2),A_2(S_m) \\ \vdots & & \ddots & \vdots & \\ A_1(S_{m-1}),A_2(S_1) & A_1(S_{m-1}),A_2(S_2) & \cdots & A_1(S_{m-1}),A_2(S_{m-1}) & A_1(S_{m-1}),A_2(S_m) \\ A_1(S_m),A_2(S_1) & A_1(S_m),A_2(S_2) & & A_1(S_m),A_2(S_{m-1}) & A_1(S_m),A_2(S_m) \end{bmatrix}$$

在满足上述占优假设的情况下，总有策略S_i，使得无论A_2选择何种策略，A_1选S_i的收益总是比其他策略占优，即严格占优策略。在多方参与的游戏中，则先考虑最优、次优……最差情况，对于优先级相同的玩家，则按照期望损失给予相应的策略。

3.2 问题三模型的求解

3.2.1 第五关求解

第五关中，由于没有村庄的存在，玩家在确定最优方案时，只需要购买刚好够量的资源即可，不需要考虑村庄购买资源的额外价格；虽然玩家不一定要走单人情况下的最优策略，但考虑到玩家数量只有两个，且路径返回对玩家而言是亏损的，（即不需要考虑路径返回的情况），因此，可以为玩家规划出到达终点的最优和次优方案。

（1）首先，在第五关中，由于玩家数量只有两个，若两名玩家选择了同一条到达终点的路线，则需要某一名玩家错开行走，错开的方式是绕路或停留，先到另一个点后再到达，因此，对于绕开的玩家来说需要多消耗一天的时间和行走的资源，损失比停留一天的损失大，由此，对于不在主路线（直达终点）上的区域，我们可以不必考虑，化简为如下：（每条线的权重为 1）

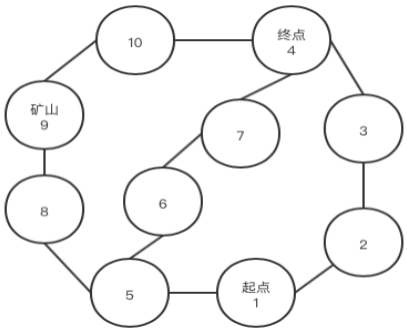

图 16 第五关化简地图

（2）经过求解可得出各个点在不同时间t到终点的最大收益如图所示：

表 3 各个点在不同时间t到终点的最大收益

日期 / 节点	1	2	3	4	5	6	7	8	9	10
1	9535	9535	9480	9425	9290	9155	9020	0	0	0
2	0	9535	9535	9480	9425	9290	9155	9020	0	0
3	0	0	9535	9535	9480	9425	9290	9155	9020	0
4	0	0	0	9535	9535	9480	9425	9290	9155	9020
5	0	9425	9425	9370	9235	9100	8965	0	0	0
6	0	0	9425	9425	9425	9370	9235	9100	0	0
7	0	0	9425	9425	9370	9235	9100	8965	0	0
8	0	9315	9315	9180	9045	8910	0	0	0	0
9	0	0	9315	9315	9180	9045	8910	0	0	0
10	0	0	0	9315	9315	9180	9045	8910	0	0

（3）据此求解出不同的时间t下，玩家从点i到其邻点j的概率，下图为第三天的结果。

表 4 玩家从点i到其邻点j的概率（第三天）

节点 / 节点’	1	2	3	4	5	6	7	8	9	10
1	0.0633	0.0637	Inf	Inf	0.0630	Inf	Inf	0.0622	Inf	Inf
2	Inf	0.1261	0.1261	Inf	Inf	Inf	Inf	Inf	Inf	Inf
3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
5	Inf	Inf	Inf	Inf	0.1246	0.1246	Inf	Inf	Inf	Inf
6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0.1232	0.1232	Inf
9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

（4）计算不同时间t下从点i到其邻点j的期望收益，如图为寻找最优策略的部分期望收益计算结果。

表 5 部分期望收益结果

节点 / 节点	1	2	3	4	5	6	7	8	9	10
1	55.000	110.028	NaN	NaN	110.028	NaN	NaN	110.026	NaN	NaN
2	NaN	55.000	110.996	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	55.000	110.437	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	55.000	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	55.000	110.973	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN	55.000	110.427	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	55.000	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	55.000	110.959	NaN
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-34.621	110.417
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	55.000

（5）最后利用第一问的计算方法求解，得到最大的平均期望收益为 9526.3165 元，最大的食物的消耗期望为 33.6316 箱，最大的水的消耗期望为 27.4737 箱。所以玩家在起点需购买 34 箱食物，28 箱水，路线如下：

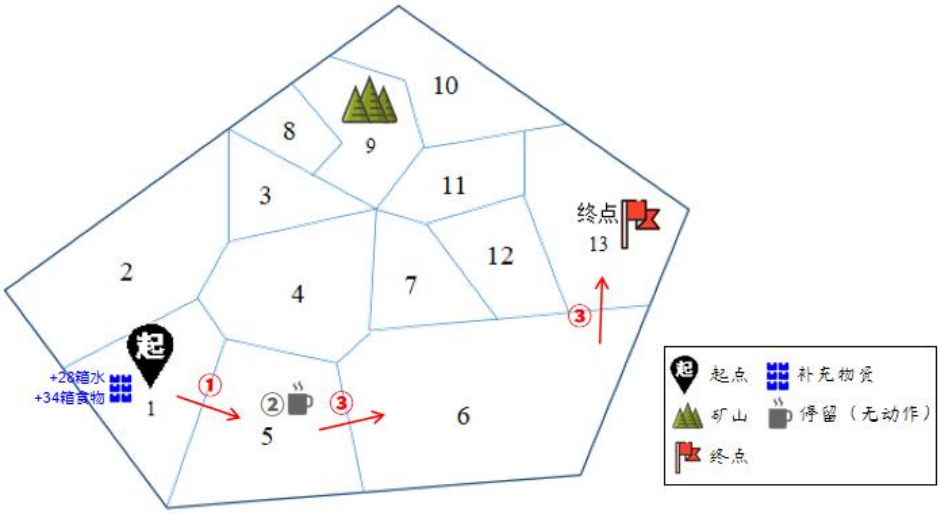

图 17 第五关路线图

由于玩家数只有两个，结果和单人游戏时的最优策略相近，是可以说明我们模型的合理性的。

3.2.2 第六关求解

我们根据 3.1.2 所建立的基于博弈论的占优策略模型，可以绘制求解流程图如下所示：

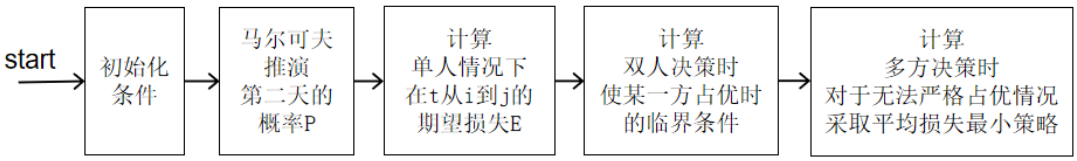

图 18 第六关求解流程图

由于在第六关中，玩家到前进方向的任意一个点时，所有邻点到终点的距离都是相等的，因此，在玩家发生冲突时，终点的损失对于玩家的收益函数的影响是相同的，经过模拟仿真，得到结果如下图所示：(w和f是剩余水和食物的箱数)

表 6 第六关策略选择

占优临界	策略
节点 13	占优势的玩家应当选择前往矿山，对于水或资源不够的玩家应当选择前往村庄补给,在有两人占优时，应选择前往矿山
$w = 28$或$f = 29$	
节点 15	占优势的玩家应当选择前往 15,对于水或资源不够的玩家应当选择前往村庄补给，在有两人占优时，应选择停留
$w = 14$或$f = 15$	
节点 18	占优势的玩家应当选择前往矿山，对于水或资源不够的玩家应当选择前往村庄补给，在有两人占优时，仍选择前往矿山
$w = 16$或$f = 17.14$	
节点 19	在时间允许的情况下，占优势的玩家应当选择前往矿山，对于水或资源不够的玩家应当选择前往村庄补给，在两人占优时，选择停留
$w = 14$或$f = 15$	
节点 20	占优势的玩家应当选择停留，对于水或资源不够的玩家应当选择前往终点，在两人占优时，仍选择停留
$w = 7$或$f = 7.5$	
节点 23	占优势的玩家应当选择前往矿山，对于水或资源不够的玩家应当选择前往节点 24，当两人占优时，选择停留
$w = 14$或$f = 15$	

节点 24	占优势的玩家应当选择停留，对于水或资源不够的玩家应当选择前往终点，有两人占优时，选择停留
$w < 7$或$f < 7.5$	

起点开始的大部分点对玩家的决策影响都是随机的，而只有在 13，15，18，19，20，23，24，如下是各点的决策方案，由图可知，在有两人参与决策时，占优势的玩家不一定选择最优的方案，如节点 20 时，占劣势的玩家需选择到达终点，而占优势的玩家选择停留，如果此时占优玩家选择前往终点，而占劣势玩家选择停留，则此时占优玩家的收益最大，但实际上该方案的风险也是最大的，而对于劣势玩家而言，有更大的概率选择前往终点，此时停留避开冲突对占优玩家而言是比较稳妥的选择，这也体现了结果的合理性。

六、 模型的讨论和合理性分析

对于问题三，我们针对参与玩家的数量n从 2 增加到 10，根据拟合的曲线可以发现，当玩家的数量逐渐增加时，玩家选择最佳路径的收益逐渐减少，且减少的趋势越来越大，这与实际情况是相吻合的，体现出我们模型的合理性。

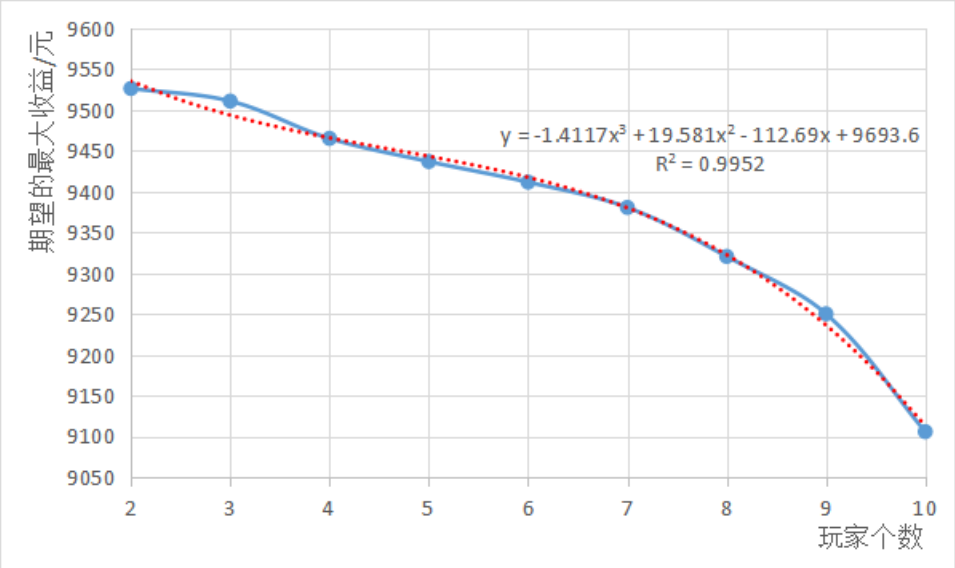

图 19 玩家人数和最大收益的拟合曲线

七、 模型评价

模型的优点：

（1）问题一的模型充分考虑了影响玩家决策的确定因素与随机因素，使模型的建立更加符合游戏设定。从玩家的切身思想出发，保证路线最优化、收益最大化。

（2）问题二的模型利用马尔可夫链，基于关卡一、二所给的 30 天天气状态计算出的天气跳转概率，在已知当天天气的条件下预测未来天气状况。使得对天气的预测更加符合游戏设定、更加合理。

（3）问题二、三的模型均以第一问的模型为基础，增加或改变了相应影响因子。使得模型整体性更强、思路清晰。

（4）问题三的模型在影响玩家决策的相关因素上又增加了对其余玩家剩余资源量的考虑，使得多人游戏下各个玩家不同决策之间的互相影响得以衡量，提高准确度。

模型的缺点：

（1）问题一的模型运算规模较大，耗时长。

（2）问题二的天气预测是基于一二关 30 天天气跳转概率给出的，游戏天气样本数量较少，存在预测的误差。

（3）问题三采用玩家到达每一个点的平均收益代替问题一中的实际收益，使得对收益的衡量误差增大，存在一定的不准确性。

八、 参考文献

[1] 张水舰, 刘学军, 杨洋. 动态随机最短路径算法研究[J]. 物理学报, 2012, 61(16): 7-16.

[2] 戴玲,夏学知.基于 Markov 天气模型的流量管理改航策略[J].舰船电子工程,2007(4):57-59.

[3] 王芳,万磊,徐玉如,等.基于改进人工势场的水下机器人路径规划[J].华中科技大学学报：自然科学版,2011(S2):184-187.

[4] 尚金成,张维存,等.一种基于博弈论的发电商竞价策略模型与算法[J].电力系统自动化,2002(9):12-15.

九、 附录

附录清单:

1. Result.xlsx
2. 问题一仿真代码
3. 问题二马尔可夫收敛代码
4. 问题三（1）仿真代码
5. 问题三（2）仿真代码

1. Result.xlsx

第一关				
日期	所在区域	剩余资金数	剩余水量	剩余食物量
0	1	5780	178	333
1	25	5780	162	321
2	24	5780	146	309
3	23	5780	136	295
4	23	5780	126	285
5	22	5780	116	271
6	9	5780	100	259
7	9	5780	90	249
8	15	4150	243	235
9	13	4150	227	223
10	12	4150	211	211
11	12	4150	201	201
12	12	5150	177	183
13	12	6150	162	162
14	12	7150	138	144
15	12	8150	114	126
16	12	9150	90	108
17	12	10150	60	78
18	12	10150	50	68
19	12	11150	26	50
20	13	11150	10	38
21	15	10470	36	40

22	9	10470	26	26
23	21	10470	10	14
24	27	10470	0	0
25				
26				
27				
28				
29				
30				

第二关				
日期	所在区域	剩余资金数	剩余水量	剩余食物量
0	1	5300	130	405
1	2	5300	114	393
2	10	5300	98	381
3	19	5300	88	367
4	19	5300	78	357
5	27	5300	68	343
6	28	5300	52	331
7	28	5300	42	321
8	29	5300	32	307
9	30	5300	16	295
10	39	3460	184	283
11	39	3460	174	273
12	47	3460	158	261
13	55	3460	148	247
14	55	4460	124	229
15	55	5460	100	211
16	55	6460	76	193
17	55	7460	46	163
18	55	8460	16	133
19	62	4730	201	207
20	55	4730	185	195
21	55	5730	170	174

22	55	6730	155	153
23	55	7730	131	135
24	55	8730	116	114
25	55	9730	86	84
26	55	10730	62	66
27	55	11730	47	45
28	55	12730	32	24
29	56	12730	16	12
30	64	12730	0	0

2. 问题一仿真代码

```
global neighbor_; %邻居
global get_ij;%收益
global w_ij;
global f_ij;
global day_ij;
global f_left;
global w_left;
neighbor_ = [1,1,1,0;1,1,1,0;1,1,1,1;0,1,1,1];
global get_mat;
get_mat = -inf(400,600,n,30);
get_mat(:,:,1,1) = 0;
mine_money = 1000;
bag = 1200; %负重 1200kg
money = 10000; %初始资金
w_weight = 3; %水的负重
f_weight = 2; %食物的负重
f_price = 10; %食物的基价
w_price = 5; %水的基价
mine = 4;
vilige = 3;
s = 1;
e = 2;
w = [5,8,10];
f = [7,6,10];%不同天气下食物和水的消耗
day = [2,2,1,3,1;2,3,1,2,2;3,2,1,2,2;2,3,3,2,2;1,1,2,1,3;2,1,1,2,2]; %30 天的天气情况
day = day';
```

```
n = 4;%端点个数
d_n = [0,3,6,8;3,0,3,5;6,3,0,2;8,5,2,0];%端点之间的距离
f_left = -inf(n,n,30);
w_left = -inf(n,n,30);
get_ij = -inf(n,n,30);
w_ij = inf(n,n,30);
f_ij = inf(n,n,30);%食物的开销
day_ij = inf(n,n,30);%需要花多少天
for k = 1:30
    temp1 = day_ij(:,:,k);
    index_1 = find(day(k:end)~=3); %找到 k 天后不是沙尘暴天气的索引
    judge = (d_n <= length(index_1)& d_n~=0); % 找到 k 天后相互之间可以到达矩阵,不包含到自身
    index_2 = find(judge == 1); %找到可以相互到达的各个坐标
    temp1(index_2) = index_1(d_n(index_2)); %一个距离走一天，需要找 x 距离个非沙尘暴天气才能走完
    temp1(find(d_n==0)) = 0;
    day_ij(:,:,k) = temp1;
    for M = 1:n
        for N = 1:n
            if(M ~= N)
                if temp1(M,N) ~= inf  %不能到达的地方不考虑
                weather = day(k:k + (temp1(M,N)-1)); %找到走的这段路的天气情况;
                w_ij(M,N,k) = length(weather==1)*(w(1))*2 + length(weather == 2)*(w(1))*2 + length(weather==3)*(w(3));
                f_ij(M,N,k) = length(weather==1)*(f(1))*2 + length(weather == 2)*(f(1))*2 + length(weather==3)*(f(3));
                    end
                else
                    if M == mine
                        f_ij(M,N,k) = day(k) * f(day(k)) * 3;
                        w_ij(M,N,k) = day(k) * w(day(k)) * 3;%挖矿的开销是基础开销的三倍
                        get_ij(M,N,k) = 1000 - f_ij(M,N,k) * f_weight * f_price * 2 - w_ij(M,N,k) * w_weight * w_price * 2;
                    else
                        f_ij(M,N,k) = day(k) * f(day(k));
```

```
                    w_ij(M,N,k) = day(k) * w(day(k));
                    get_ij(M,N,K) = - f_ij(M,N,k) * f_weight * f_price * 2 -
w_ij(M,N,k) * w_weight * w_price * 2;
                end
            end
        end
    end

    function max_get = get_max(t,e)
    global neighbor_;
    global get_mat;
    global day_ij;
    global get_ij;
    global f_ij;
    global w_ij;
    while t > 0
        for w = 1:400
            for v = 1:600
                max_get = get_mat(w,v,e,t);
                weight = w * 3 + v * 2;
                if(weight <=1200)
              neighbor = find(neighbor_(e,:)== 1); %找到邻居
        for i = 1:length(neighbor) %对每一个邻居，判断最大值
            get_day = day_ij(e,neighbor(i),t) %获取邻居到本点需要的时间
            left_w = w - w_ij(e,neighbor(i),t);
            left_f = v - f_ij(e,neighbor(i),t);
            if get_day ~=inf && left_w >=0 && left_f >=0
            temp_get = get_max(t- get_day,neighbor(i)) + get_ij(e,neighbor(i),t);
            else
                temp_get = -inf;
            end
            if max_get < temp_get
                max_get = temp_get;
            end
        end
                end
```

```
            t = t - 1;
                end
            end
        end
    end
```

3. 问题二马尔可夫收敛代码

```
clc;
clear all;
a=[1/3,0.5,1/6;1/2,3/7,1/14;3/8,1/2,1/8];
b=[0.5,0.4,0.1];
c = zeros(10,3);
for ii=1:10
    if ii==1
        d=b*a;
    else
        d=d*a;
    end
    c(ii,:) = d;
end
```

4. 问题三（1）仿真代码

```
sunny = [3,4];
hot = [9,9];
money = 10000;
weather = [1,2,1,1,1,1,2,2,2,2]; %十天的天气
result = -inf(3,10); %三条路线十天的收益
cost_day = [3,4,5];
for t = 1:10
    for i = 1:3 %三条路线
    judge   = find((10 - t)< cost_day(i)); %判断剩余天数是否能走到终点
    if sum(judge) > 0
    result(i,t) = -inf;%到不了的收益为负无穷
    else
        if t == 1
            weather_ = weather(t:t + cost_day(i) - 1);
            lost_w = sum(weather_ == 1) * 2 * sunny(1) + sum(weather_ == 2) * 2 * hot(1)
            lost_f = sum(weather_ == 1) * 2 * sunny(2) + sum(weather_ == 2) * 2 * hot(2);
```

```
                lost = lost_w * 5 + lost_f * 10;
                result(i,t) = money - lost;
            else
                weather_ = weather(1:t + cost_day(i) - 1);
                stop_day = t - 1; %停留的天数
                weather_hot = weather_(find(weather_==2)); %高温时停留最好
                if length(weather_hot) > stop_day %若停留的天数没有高温天数多，则停留的天数全为高温天
                lost_w = sum(weather_ == 1) * 2 * sunny(1) + (sum(weather_ == 2) * 2 - stop_day) * hot(1);
                lost_f = sum(weather_ == 1) * 2 * sunny(2) + (sum(weather_ == 2) * 2 - stop_day) * hot(2);
                lost = lost_w * 5 + lost_f * 10;
                result(i,t) = money - lost;
                else %停留了所有的高温天，且停留了一些晴朗的天气
                lost_w = (sum(weather_ == 1) * 2 - (stop_day - sum(weather_ == 2))) * sunny(1) + sum(weather_ == 2) * hot(1);
                lost_f = (sum(weather_ == 1) * 2 - (stop_day - sum(weather_ == 2))) * sunny(2) + sum(weather_ == 2) * hot(2);
                lost = lost_w * 5 + lost_f * 10;
                result(i,t) = money - lost;
                end
            end
        end
        end
end
res = result';
max_res = max(res);
[index,y] = find(res == max(res));
for i = 1:3
    result(i,1:index(i)) = max_res(i);
end

get_max = table2array(get_max);
p = zeros(10,10,10); %计算转移概率
n1 = [1,2,5,8];
n2 = [2,3];
```

```
n3 = [3,4];
n4 = [4];
n5 = [5,6];
n6 = [6,7];
n7 = [7,4];
n8 = [8,9];
n10 = [7,10];
n9 = [9,10]; %列出邻居
t = 9;
get_max(find(isnan(get_max))) = 0;
for i = 1:10
    index = find(get_max(:,i) ~= 0);
end
while t>0
        p(1,n1,t) = get_max(n1,t+1) ./ sum(get_max(n1,t+1));%势能转为概率
        p(2,n2,t) = get_max(n2,t+1) ./ sum(get_max(n2,t+1));%势能转为概率
        p(3,n3,t) = get_max(n3,t+1) ./ sum(get_max(n3,t+1));%势能转为概率
        p(4,n4,t) = get_max(n4,t+1) ./ sum(get_max(n4,t+1));%势能转为概率
        p(5,n5,t) = get_max(n5,t+1) ./ sum(get_max(n5,t+1));%势能转为概率
        p(6,n6,t) = get_max(n6,t+1) ./ sum(get_max(n6,t+1));%势能转为概率
        p(7,n7,t) = get_max(n7,t+1) ./ sum(get_max(n7,t+1));%势能转为概率
        p(8,n8,t) = get_max(n8,t+1) ./ sum(get_max(n8,t+1));%势能转为概率
        p(9,n9,t) = get_max(n9,t+1) ./ sum(get_max(n9,t+1));%势能转为概率
        p(10,n9,t) = get_max(10,t+1) ./ sum(get_max(n9,t+1));%势能转为概率
        t = t - 1;
end

p(find(isnan(p))) = 0;
p(4,:,:) = 0;
p_1 = zeros(10,10,11);
p_1(1,1,1) = 1;
for t = 1:10
    for i = 1:10
        for j = 1:10
            p_1(i,j,t+1) = sum(p_1(:,i,t)) * p(i,j,t);
        end
    end
```

```
end
lost_w = -inf(10,10,10);
lost_f = -inf(10,10,10);
cost_money = -inf(10,10,10);
E = -inf(10,10,10);
E_f = -inf(10,10,10);
E_w = -inf(10,10,10);
w_cost = [3,9];
f_cost = [4,9];
weather = [1,2,1,1,1,1,2,2,2,2]; %十天的天气
p_1(find(p_1 == 0)) = inf;
for t = 1:10
    for i = 1:10
        for j = 1:10
            if i ~= j
          lost_w(i,j,t) = 2 * w_cost(weather(t));
          lost_f(i,j,t) = 2 * f_cost(weather(t));
          cost_money(i,j,t) = lost_w(i,j,t)*5+ lost_f(i,j,t)*10;
          E(i,j,t) = cost_money(i,j,t)*(p_1(i,j,t+1).^2 * 2 + 1 - p_1(i,j,t+1).^2);
          E_f(i,j,t) =    lost_f(i,j,t)*(p_1(i,j,t+1).^2 * 2 + 1 - p_1(i,j,t+1).^2);
          E_w(i,j,t)= lost_w(i,j,t)*(p_1(i,j,t+1).^2 * 2 + 1 - p_1(i,j,t+1).^2);
            else
                if i == 9
           lost_w(i,j,t) = 3 * w_cost(weather(t));
          lost_f(i,j,t) = 3 * f_cost(weather(t));
          cost_money(i,j,t) = lost_w(i,j,t)*5+ lost_f(i,j,t) * 10;
          E(i,j,t) = cost_money(i,j,t) - 200*(p_1(i,j,t+1).^2 * 1/2 + 1 - p_1(i,j,t+1).^2);
          E_f(i,j,t) = lost_f(i,j,t);
          E_w(i,j,t) = lost_w(i,j,t);
                else
                    lost_w(i,j,t) = w_cost(weather(t));
          lost_f(i,j,t) = f_cost(weather(t));
          cost_money(i,j,t) = 5*lost_w(i,j,t) + 10*lost_f(i,j,t);
          E(i,j,t) = cost_money(i,j,t);
          E_w(i,j,t) = lost_w(i,j,t);
          E_f(i,j,t) = lost_f(i,j,t);
                end
```

```
            end
        end
    end
end
table1 = p_1(:,:,3);
table2 = E(:,:,3);
```

5. 问题四仿真代码

```
w = zeros(3,3); %根据行动结束当天预测明天天气
w = [1/3,1/2,1/6;1/2,3/7,1/14;3/8,1/2,1/8];
E = -inf(25,25,3);%根据当天的天气，给出期望
a_1 = 1000;
a_2 = 20000;
a_3 = 20600;

for j = 1:3
    E(:,:,j) =   w(j,1) * (3 * 2 *5  + 4 * 2 * 10)+ w(j,2) * (9 * 2 *
5 + 9 * 2 *10);
end
for i = 1:25
    E(i,i,:) =   w(j,1) * (3  *5  + 4 * 10)+ w(j,2) * (9 * 5 + 9  *10)
+ w(j,3) * (10 *5 + 10 *10);
    if i == 18
        E(i,i,:) = 3*( w(j,1) * (3  *5  + 4 * 10)+ w(j,2) * (9 * 5 +
9  *10)+ w(j,3) * (10 *5 + 10 *10)) - 1000;
    end
end
g_t = zeros(5,5,30);
g_w = zeros(5,5,100);
g_f = zeros(5,5,100);
for t = 1:30
    for i = 1:5
        for j = 1:5
            g_t(i,j,t) = a_1 * exp(-((25 -t) - (5 - i) - (5 - j)));%
计算 g1 损失
        end
    end
end
```

```
for w = 1:100
    for i = 1:5
        for j = 1:5
            g_w(i,j,w) = a_2 * exp(-(w - 7*(5 - i) - 7*(5 - j)));%计
算 g2 损失
        end
    end
end
for f = 1:100
    for i = 1:5
        for j = 1:5
            g_f(i,j,f) = a_3 * exp(-(f - 7*(5 - i) - 7*(5 - j)));%计
算 g3 损失
        end
    end
end
k = 12;%以下为对某一个点寻找占优策略
w1 = 25;
w2 = 100;
f1 = 100;
f2 = 100;
t1 = 25;
t2 = 1;
test = [2,3];
test_1 = zeros(1,2);
E_1 = zeros(1,2);
E_1(1) = E(k,(test(1))*5 + test(2),1);
E_1(2) =E(k,(test(2))*5 + test(1),1);
test_1(1)  =  g_t(test(1),test(2),t1)  +  g_w(test(1),test(2),w1)  +
g_f(test(1),test(2),w1);
test_1(2)  =  g_t(test(2),test(1),t1)  +  g_w(test(2),test(1),w1)  +
g_f(test(2),test(1),w1);
test_2 = zeros(1,2);
test_2(1) = E(k,(test(1))*5 + test(2),1) + g_t(test(1),test(2),t2) +
g_w(test(1),test(2),w2) + g_f(test(1),test(2),w2);
test_2(2) = E(k,(test(2))*5 + test(1),1) + g_t(test(2),test(1),t2) +
g_w(test(2),test(1),w2) + g_f(test(2),test(1),w2);
```

```
choose = zeros(2,2);
for i = 1:2
    for j = 1:2
        if i ~=j
           choose(i,j) = test_1(i) + E_1(i);
        else
            choose(i,j) = test_1(i) + 2 * E_1(i);
        end
    end
end
best = 0;
for i = 1:2
    temp = choose(i,:) > max(choose(:,:));
    if sum(temp) == 2
        best = i;
    end
end
```