基于优化模型的行动决策策略

摘要

本文是基于优化思想，使玩家在不同规则，约束条件下获取最大收益最优行动方案的决策问题。

针对问题一，对于一般情况下最优策略模型的建立，我们以游戏中玩家的资源负重上限，规定时间限值，资金限值为约束条件；以保留最大的最终收益为目标函数；以是否在矿山挖矿，是否在村庄买资源，买多少资源，是否停留为决策变量；以及游戏过程中玩家的资金，日消耗和转移消耗情况来综合建立最优策略模型。

对于第一二关最优策略的规划，我们基于**贪婪思想**，从玩家消耗较少资源和挖矿获取更多收益两个角度考虑保留的最大收益。我们首先利用 **dijkstra 算法**计算游戏中玩家每次转移过程的最短路径，保证转移过程中资源消耗成本最小。然后在最短路径的基础上，在资源，时间，负重上限约束条件下对挖矿时间规划，使玩家到达终点保留最大的收益。根据第一关的条件，玩家在第 24 天到达终点结束游戏，得到的最大剩余资金量为 10470 元；根据第二关的条件，玩家在第 30 天到达终点结束游戏，得到的最大剩余资金量为 12580 元。

针对问题二，对于一般情况下最佳决策模型的建立，由于玩家仅知道当天的天气情况，我们利用**多步决策方法**通过对每一步的最佳策略进行决策来得到最终的最大收益。对于第三四关的最佳策略，我们依然基于“贪婪思想”使整个游戏过程中玩家沿最短路径行动。将行动决策路线分类讨论，通过**随机仿真**天气求解每类行动决策下每步决策方案的**期望收益**值，逐步选择每一步决策方案中期望收益最大的决策方案，一直到到达终点时求解出最终最大收益的行动策略。最终得出结论：在仅知道当天天气安排的情况下，对于第三关，经过矿山的路线的最终期望收益始终高于不经过矿山的路线；对于第四关，玩家应优先选择从“起点-矿山-村庄-矿山-终点”这条期望收益最高的行动策略。

针对问题三，根据规则，一个区域若同时有多名玩家进入，则他们的每个人的正向收益会比独自进入该区域获取正向收益小。因此为获取更大收益，我们利用“**静态博弈**”的思想，从自身利益角度出发，采用博弈论中的“最大最小策略”来对玩家的行动方案进行决策，获得在最差情况下（玩家每一步行动决策都有 k 名玩家同行）的最优行动决策方案。对于第五关的最佳策略讨论，利用博弈论中的“纳什均衡[1]”，将行动策略分为“经过矿山和不经过矿山”两种，然后对玩家在过程中的收益和资源消耗情况进行综合考虑，利用规划思想，在时间，资源等条件约束下对玩家挖矿时间、时长以及过程停留时间安排进行规划，使得玩家在消耗较少资源的情况下获取更多的收益。最终我们确定的最优行动策略为：**玩家不经过矿山，在第二天高温天气原地停留 1 天后，直接沿最小路径到终点。**

对于第六关的最佳策略讨论，我们利用“**动态博弈**”的思想，根据所在位置和剩余资源数得出各名玩家的决策集合，从某名玩家角度出发，选定某一策略，计算在其余玩家所有可行的策略下可能造成的最坏影响，再选出能使最坏影响降到最低的策略作为当前的最优策略。

最后，我们进行了灵敏度分析和对模型的优缺点进行了评价，并对模型进行了改进。

关键词：贪婪思想，多步决策模型，随机仿真，期望收益，博弈论

一、问题的提出与重述

1.1 问题的提出

现有一个小游戏：玩家根据一张地图，从起点出发利用初始资金购买一定的水和食物，在沙漠行走。中途可以在矿山村庄进行资金或资源补充。最终目标是在规定时间内到达终点且尽可能保留较多资金。

1.2 问题的重述

游戏基本规则如下:

（1） 第 0 天游戏开始，玩家从起点出发，到达终点游戏结束。玩家需在规定时间内到达目的地。

（2） 玩家拥有水和食物两种资源，且数量不能超过玩家的负重上限。玩家需保证在到达终点前资源剩余不为 0，否则视为游戏失败。

（3） 玩家每天可从一个区域到达与它相邻的另一个区域，也可以选择在原地停留。玩沙暴日必须在原地停留。

（4） 基础消耗量为玩家原地停留一天的资源消耗量，行走一天的消耗量为基本消耗量的二倍。

（5） 基础收益为玩家挖矿一天赚取的收益，挖矿一天的资源消耗量为基础消耗量的三倍。玩家沙暴日也可以挖矿。

（6） 玩家可在起点和村庄购买水和食物，但不能多次在起点购买物资。在村庄购买每箱物资价格为基准价格的 2 倍。玩家可以在终点以一半的基准价格将剩余物资退回。

（7） 沙漠中一天内每个区域的天气相同。每天的天气为晴朗，高温，沙暴中的一种。

请根据上述规则解决如下问题：

问题一，现有一名玩家，在整个游戏时间段内每天天气情况已知的条件下，给出一般情况下的最优出行策略。并根据此最优策略求解第一关和第二关终点出行情况，将相应结果分别填入 Result.xlsx。

问题二，现有一名玩家，在仅知道当天的天气状况，决定当天的行动方案。请给出一般情况下的玩家出行最优策略，并对第三关和第四关下的游戏策略进行具体讨论。

问题三，现有 n 名拥有相同初始资金的玩家，同时从起点出发。若某天中有任意 k 名玩家均从区域 A 走到区域 B，则这 k 名玩家每一位的资源消耗数量均为基础消耗的 $2k$ 倍；若某天中有任意 k 名玩家在同一矿山挖矿，则他们中的每一位的资源消耗数量均为基础消耗量的 3 倍，并且每一名玩家一天中碳钢挖矿获得的资金是基础收益的 $1/k$；若某天中有任意 k 名玩家在同一村庄购买资源，那么每箱资源的价格均为基准价格的 4 倍。其他情况下玩家消耗的资源数量和资源价格与单人游戏相同。

现基于上述规则解决如下问题：

（1）在整个游戏时间段每天天气已知，每名玩家在第 0 天确定好行动方案且此后不能修改的情况下，试给出一般情况下玩家采取的行动策略，并对第五关进行具体讨论。

（2）在所有玩家仅知道当天天气的情况下，从第一天起，每名玩家当天的行动方案和资源剩余量都会在当天行动结束后公布给其他玩家，然后根据情况确定第二天的行动安排。试给出一般情况下玩家应采取的行动策略，并对第六关进行具体讨论。

二、问题的分析

针对问题一，对于在全部天气已知的情况下建立一般情况下的最佳策略模型。我们以游戏中玩家的资源负重上限，规定时间限值，资金限值为约束条件，以保留的最终收益最大为目标函数，以是否在矿山挖矿额，是否在村庄买资源，买多少资源，在地区是否停留为决策变量，建立一般情况下的最佳行动策略规划模型。

对于第一二关的最佳策略。我们基于贪婪算法思想，从玩家消耗最少资源和玩家挖矿获取更多收益两个角度考虑，使玩家最终保留的收益最大。首先从资源消耗最少方面分析，我们可以通过寻找游戏过程中玩家每次转移时的最小路径入手，保证在局部路程上消耗资源最少，以减少资源消耗成本。然后从玩家挖矿角度考虑，我们通过计算发现在三种天气情况下玩家在矿山挖矿均会有正向收益，其中晴天挖矿收益最大，沙暴挖矿收益最小。因此我们在最短路径的基础上，留有较多的挖矿时间，然后在资源，时间，负重上限约束条件下对挖矿时间安排，使玩家到达终点保留的收益最大。

针对问题二，对于一般情况下最佳决策模型的建立，由于玩家仅能知道当天的天气情况，所以我们利用多步决策方法通过对每一步的最佳策略进行决策来得到最终的最大收益。对于第三关的最佳策略，我们依然基于“贪婪思想”使整个游戏过程中玩家沿最短路径行动。然后我们将行动决策路线分为“经过矿山和不经过矿山”两类讨论，通过随机仿真天气求解每类行动决策下每步决策方案的期望收益值，然后选择每一步决策方案中期望收益最大的决策方案，一直到到达终点时求解出最终最大收益的行动策略。

对于第四关的最佳策略，与第三关的想法类似，在整个游戏过程中玩家沿最短路径行动的基础上，根据矿山，村庄，终点的位置布局，我们首先从起点先到“村庄或者矿山”的情况分为两类。然后根据已有资源逐步增加去矿山的次数。经过逐步尝试，确定出最终第四关行动路线策略数。然后对第四关玩家在不同天气下挖矿的收益情况进行分析，过随机仿真天气求解每类行动决策下每步决策方案的期望收益值，然后选择每一步决策方案中期望收益最大的决策方案，一直到到达终点时求解出最终最大收益的行动策略。

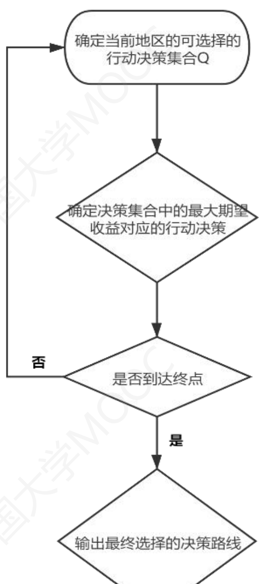

图 1 行动路线多步决策流程图

针对问题三，根据问题三的规则，一个区域若同时有玩家进入，则他们的个人正向收益会比独自进入该区域获取正向收益小。为获取更大收益，玩家应尽量避免与其他队友在同一区域同行。但由于不知道其他玩家的行动方案，我们无法实时做出行动决策。因此我们利用“静态博弈”的思想，仅考虑自身利益最大化，采用博弈论中的“最大最小策略”来对玩家的行动方案进行决策，获得在最差情况下（玩家每一步行动决策都有 k 名玩家同行）的最优收益，使最差情况下玩家的资源和资金消耗最少。

对于第五关的最佳策略讨论，第五关没有对村庄的考虑，因此我们将行动策略分为：“经过矿山和不经过矿山”两种，然后对玩家在过程中的收益和资源消耗情况进行综合考虑，在时间，资源等条件约束下对玩家挖矿时间、时长以及过程停留时间安排进行规划，使得玩家在消耗较少资源的情况下获取更多的收益。、

对于第六关的最佳策略讨论，由于每名玩家在当天行动完成后能够知道其余玩家所在区域和携带物资数，列出自身以及其余玩家所有可行的路线，从自身角度出发，其余玩家的不同路线组合均有一定的概率出现，在每一个策略下计算在其余玩家所有可行的策略下可能造成的最大风险，再选出风险最高的情况最好的策略作为当前的最优策略。

三、模型假设

1、不考虑除天气、物资以外的因素对行动的影响

2、假设沙漠中所有区域天气相同

四、符号说明

符号	符号含义	单位
C	玩家初始资金	元
P_1	每箱水的基准价格	元/箱
P_2	每箱食物的基准价格	元/箱
X_i	第 i 天买水的箱数	箱
Y_i	第 i 天买食物的箱数	箱
H	在矿场打矿一天的收益	元
D	每一关规定的天数	天
G	所属关卡的地区数	个
M_1	每箱水的质量	千克
M_2	每箱食物的重量	千克
a_i	第 i 天水的剩余量	箱
b_i	第 i 天食物的剩余量	箱
W	玩家资源可负重上限值	千克
k	玩家人数	个
R_q	决策风险率	%

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 全部天气已知的一般情况下最佳策略模型的建立

• **目标函数：**

在规定时间内到达目的地且保留最多的资金：

$$max = \mathrm{C} - [(P_1X_0 + P_2Y_0) + 2\left(P_1\sum m_iX_i + P_2\sum n_iY_i\right)] + \sum d_iH \qquad (1)$$

C 为初始资金，P_1为每箱水的基准价格，P_2为每箱食物的基准价格，X_i为第 i 天买水的箱数，Y_i为第 i 天买食物的箱数，H 为在矿场打矿一天的收益。

• **决策变量：**

x_{ij}为玩家在第 i 天是否到达第 j 个地区：

$$x_{ij} = \begin{cases} 1,\text{第 i 天到达第 j 个地区} \\ 0,\text{第 i 天不到第 j 个地区} \end{cases} \qquad (2)$$

d_i为玩家在第 i 天是否挖矿：

$$d_i = \begin{cases} 0,\text{第 i 天不挖矿} \\ 1\text{，第 i 天挖矿} \end{cases} \qquad (3)$$

m_i为玩家在第 i 天是否买水：

$$m_i = \begin{cases} 0,\text{第 i 天不买水} \\ 1\text{，第 i 天买水} \end{cases} \qquad (4)$$

n_i为玩家在第 i 天是否购买食物：

$$n_i = \begin{cases} 0,\text{第 i 天不买食物} \\ 1\text{，第 i 天买食物} \end{cases} \qquad (5)$$

• **约束条件：**

（1） 规定时间内到达终点

$$\sum x_{ij} \le D \qquad (6)$$

$$\sum_{j=1}^{G} x_{ij} = 1 \qquad (7)$$

D 为规定的天数，G 为所属关卡的地区数

（2） 资源负重约束

$$M_1a_i + M_2b_i \le W \qquad (8)$$

$$a_i \ge 0, b_i \ge 0$$

M_1为每箱水的质量，M_2为每箱食物的重量，a_i为第 i 天水的剩余量，b_i为第 i 天食物的剩余量，W 为资源负重上限值

（3） 矿山和村庄不能在一天内同时到达

$$d_i+m_i \leq 1 \tag{9}$$

$$d_i+n_i \leq 1 \tag{10}$$

（4） 资金约束

初始资金约束：

$$(a_1+g_i)P_1+(b_1+h_i)P_2 \leq C \tag{11}$$

g_i为第 i 天水的基础消耗量，h_i为第 i 天食物的基础消耗量

• **资源状态转移情况：**

（1） 资金转移状态情况：

$$B_i = B_{i-1} + Hd_i - 2(P_1 m_i X_i + P_2 n_i Y_i) \tag{12}$$

B_i为第 i 天玩家拥有的资金数

（2）不同天气下水资源转移状态情况：

$$a_i = a_{i-1} - g_i(w_i + 2d_i + 1) + m_i X_i \tag{13}$$

（3）不同天气下食物转移状态情况：

$$b_i = b_{i-1} - h_i(w_i + 2d_i + 1) + n_i Y_i \tag{14}$$

w_i为玩家是否在第 i 个地区停留：

$$w_i = \begin{cases} 0,\text{玩家在第 i 个地区停留} \\ 1,\text{玩家不在第 } i \text{ 个地区停留} \end{cases} \tag{15}$$

• **综上，**一般情况下玩家的最优策略模型如下：

$$max = \mathrm{C} - [(P_1X_0 + P_2Y_0) + 2\left(P_1\sum m_iX_i + P_2\sum n_iY_i\right)] + \sum d_iH$$

$$
s.t.\begin{cases}
\sum_{G} x_{ij} \le D \\
\sum_{j=1} x_{ij} = 1 \\
M_1 a_i + M_2 b_i \le W \\
(a_1 + g_i)P_1 + (b_1 + h_i)P_2 \le C \\
B_i = B_{i-1} + Hd_i - 2(P_1 m_i X_i + P_2 n_i Y_i) \\
a_i = a_{i-1} - g_i(w_i + 2d_i + 1) + m_i X_i \\
b_i = b_{i-1} - h_i(w_i + 2d_i + 1) + n_i Y_i \\
d_i = \begin{cases} 0,\text{第 i 天不挖矿} \\ 1\text{，第 i 天挖矿} \end{cases} \\
m_i = \begin{cases} 0,\text{第 i 天不买水} \\ 1\text{，第 i 天买水} \end{cases} \\
n_i = \begin{cases} 0,\text{第 i 天不买食物} \\ 1\text{，第 i 天买食物} \end{cases} \\
w_i = \begin{cases} 0,\text{玩家在第 i 个地区停留} \\ 1\text{，玩家不在第 } i \text{ 个地区停留} \end{cases} \\
d_i + n_i \le 1 \\
d_i + m_i \le 1 \\
a_i \ge 0, b_i \ge 0
\end{cases} \tag{16}
$$

5.1.2 第一关卡最佳行走策略的求解

最佳策略的目标是在规定时间内到达终点，且使最终保留的收益最大。我们从游戏过程消耗最少资源和赚取较多的收益两个角度进行决策。基于贪婪思想，使玩家在游戏过程中转移时均在最小路的情况下进行，以保留更多资源使玩家有更多的时间在矿山挖矿赚钱。

因此我们在资源，时间，负重上限约束条件下，寻找整个过程每一转移阶段的最短路径，规划玩家在矿山的挖矿时间，获取最大收益大。具体计算步骤如下：

Step1：我们对矿场挖矿挣钱和在矿场挖矿时付出的成本进行对比，见下表。由表可知，在沙暴天气资源消耗量最大的情况下进行挖矿，仍能保证挖矿获取的利益大于资源消耗成本。因此我们尽量保证玩家在矿山工作的时间较长使最终保留的资金更多。

表 1 挖矿的收益与成本

挖矿收益(元/天)	晴天挖矿资源消耗成本（元/天）	高温挖矿资源消耗成本（元/天）	沙暴挖矿资源消耗成本（元/天）
1000	285	300	450

Step2：利用 dijkstra 算法求解从起点到矿山的路径最小，使途中消耗的资源数最少。

考虑起点到矿山的最小路径是否经过村庄进行资源补给问题，为使玩家在矿场挖矿时间较长，因此我们选择在最小路径时经过村庄先进行资源补给，补给到玩家最大负重上限，以使在矿场的工作时长最长。最终我们选择的起点到矿山的最小路径如下：

表 2 起点到矿山最小路径表

1（起点）	25	24	23	22	9	15（村庄）	13	12（矿山）

Step3：比较表 1 中挖矿的成本与收益，我们发现在沙暴天挖矿虽然能保证有正向收益，但比较晴天和高温天气挖矿成本，在沙暴天挖矿获取收益的把性价比并不高。因此我们在总资源一定和时间允许的条件下，应该尽量将挖矿时间安排在晴天或者高温天气进行。

我们通过计算，在起点到矿山的最小路径下玩家在第 10 天到达矿山，然后从第 11 天开始在矿山挖矿。在保证资源不被用完的情况下，玩家可以从第 11 天到 19 天一直在矿山挖矿，但考虑到这一段时间有三天是沙暴天气，挖矿性价比不高。因此，我们从沙暴日挖矿天数尽可能少来的角度，将行走策略分为沙暴天气不挖矿和在其中一天沙暴天进行挖矿两种情况来考虑：

（1）在沙暴天气挖矿，我们选择在第 11 天沙暴天气挖矿，第 17,18 天玩家在原地休息等待。第 20 天玩家离开矿山到村庄进行资源补给，使资源数正好能保证玩家走完村庄到终点间的最短路。

（2）不在沙暴天气挖矿，那么在第 11 天和第 19 天期间，除沙暴日外其他日期玩家均挖矿，然后在第 20 天离开矿山到村庄进行资源补给，补充的数量正好保证玩家能走完村庄到终点间的最短路。

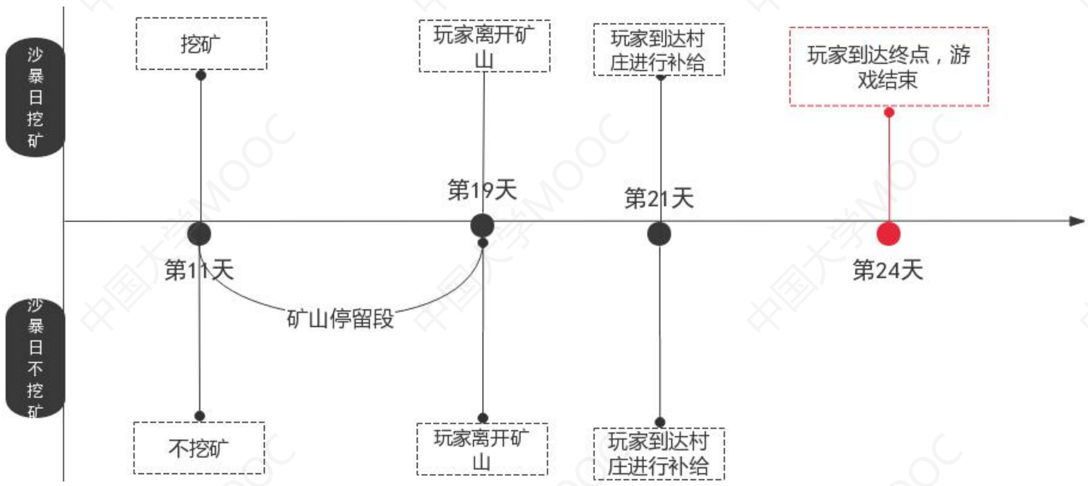

图 2 两种情况下玩家行程安排图

• 上述两种情况，均不考虑玩家第二次从村庄补给完再回矿山挖矿的情况。因为玩家在第 21 天第二次回到村庄，若补给完再回矿山，途中需要消耗 580 元的成本，而为了能按时到达目的地，玩家返回矿山后只能在矿山挖一天矿，且挖矿天气为沙暴日，挖矿一天收益为 550 元，获得的收益小于返回矿山的成本，因此不考虑二次返回矿山。

最终我们得到最优策略安排见附件，玩家在第 24 天到达终点，保留的最多收益为 10470 元。

5.1.3 第二关卡最佳行走策略的求解

基于第一关的求解思想，我们仍基于贪婪思想使起点到矿山，矿山到村庄，以及最后阶段返回终点的路径，每一过程都是在最短路径下进行。并且在资源，时间，负重上限约束条件下使玩家在矿山工作的时间尽可能的长，以获取更大的收益。

第一程：从起点到矿山（30）。为使在矿场待的时间较长，我们在起点将资源补给到玩家负重上限。起点到矿山（30）的最短路径如下：

表 3 起点到矿山（30）的最短路径

日期	0	1	2	3	4	5	6	7	8	9
所在区域	1	2	3	4	4	5	13	13	22	30

第二程：玩家在第 9 天到达矿山（30）后，根据已有的物资数，在第 10 天开始挖矿。由于第 11 天为沙暴天气，考虑到在沙暴天挖矿收益最低且资源消耗最大，因此我们选择在减小资源消耗的考虑下，先去村庄将资源补给到负重上限，然后将挖矿时间安排在晴朗和高温天气下进行，在第 13 天返回矿场（30）后开始挖矿到第 18 天，剩余资源不足以继续挖矿的情况下，第 19 天到村庄（39）进行补给，补给到负重上限。

第三程：玩家在第 19 天离开村庄（39）在第 21 天到矿山（55）继续挖矿，在留有一定资源和在规定时间内回到终点的基础上，玩家挖矿到第 28 天，然后沿矿山（55）到终点的最小路径回到终点。

最终我们在第 30 天到达终点，得到的最多收益为 12580 元。

5. 2 问题二模型的建立与求解

5. 2. 1 仅知当天天气的一般情况下最佳策略模型的建立

由于仅知道当天的天气情况，无法对整个游戏过程进行总的决策。因此我们采用多步决策的思想，先确定每一步的最佳决策，再确定最终的最佳决策方案。

多步决策模型建立如下：

Step1：建立区域邻接矩阵T_{jk}

$$T_{jk}=\begin{cases}0\text{，第 j 个地区不与第 k 个地区相邻}\\1\text{，第 j 个地区与第 k 个地区相邻}\end{cases}\tag{17}$$

Step2：根据区域邻接矩阵确定玩家下一步可到达的区域，构成决策集合 Q。

Step3：对决策集合 Q 中每一个地区选项，进行天气情况的仿真。设定仿真次数为 N。

Step4：计算每步决策下仿真的不同天气情况的收益G_t（$t=1....N$），G_t是由矿山收益 $G1$, 物资花销 $G2$ 初始起点物资的花销量 $G3$, 以及到达终点后退回物资获得的收益 $G4$ 构成。

$$G_t=G1-G2-G3+G4\tag{18}$$

最终矿山收益 $G1$ 为：

$$G1=\sum d_iH\tag{19}$$

村庄物资花销 $G2$ 为：

$$G2=2\left(P_1\sum m_iX_i+P_2\sum n_iY_i\right)\tag{20}$$

初始物资花销量 $G3$ 为：

$$G3=a_0P_1+b_0P_2\tag{21}$$

终点退回物资获得的收益 $G4$ 为：

$$G4=\frac{1}{2}(a_DP_1+b_DP_2)\tag{22}$$

每步决策下不同天气情况的收益 G 为：

$$G_t=\sum d_iH-[2\left(P_1\sum m_iX_i+P_2\sum n_iY_i\right)+(a_0P_1+b_0P_2)-\frac{1}{2}(a_DP_1+b_DP_2)](23)$$

Step5：计算每步决策方案下的期望收益E_i

$$E_i=\frac{\sum_{t=1}^{N}G_t}{N} \tag{24}$$

Step6：比较每步决策所有方案的期望收益E_i，选取最大期望收益对应的决策地区。

Step7：重复上述步骤，确定到达终点前每步决策的最大期望收益行走路线。

5.2.2 第三关最佳行动策略的讨论

基于贪婪算法的思想，整个游戏玩家的转移过程均沿最短路径进行。

- **第三关游戏行动策略分类：**

由于决策线路第一天的天气情况是未知的，因此我们根据第一天的天气情况分为“晴天和高温”两种决策情况进行讨论。并且由于地区与地区间没有距离远近的划分，因此在第一天天气情况确定的基础上将线路又分为“经过矿山和不经过矿山”两种情况。

综上，第三关行走线路策略共有四种，如下：

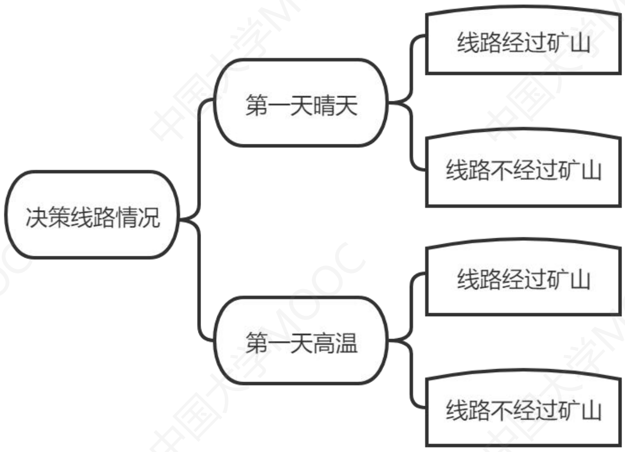

图 3 第三关行动决策分类图

- **第三关游戏规则讨论：**

（1） 玩家在游戏过程中需保证在规定时间内有足够的资源到达终点。

但由于初始资源购买量我们无法确定，因此我们考虑以每种线路决策情况下的最多资源消耗量作为每种情况下的初始资源量进行讨论。以保证无论怎样的天气安排，玩家都能保证有足够的资源到达终点。

每种情况下最多资源消耗量为最坏天气（天气全为高温）的情况下资源消耗量。

表 4 每种策略最多资源消耗量

		水消耗量（箱）	食物消耗量（箱）	资源消耗费用（元）
晴朗	经过矿山	42	44	650
	不经过矿山	78	80	1190
高温	经过矿山	54	54	810
	不经过矿山	90	90	1350

我们对晴朗和高温天气下挖矿所获取的收益进行了计算，结果如下：

表 5 不同天气挖矿收益表

天气	晴天	高温
获取收益	35 元/天	-205 元/天

由表中数据可知在高温天气下挖矿，玩家的收益损耗较大，为保证最终最大的收益，我们规定在矿山如果遇到高温天气，玩家就立刻向终点出发以减少收益损失。

因此对于第三关整个游戏过程我们需遵守的原则有：

（1） 保证有足够的资源和在规定时间内到达终点。

（2） 在矿山若遇到高温天气，玩家就立马向终点出发。

• 每种决策情况的最大收益仿真模拟：

（1）首先我们对两种天气出现的概率进行讨论。分“晴天大概率出现，高温大概率出现，晴天与高温等概率出现”三种情况讨论。

我们设定大概率为 0.7，小概率为 0.3，,然后基于上述规则对 4 中决策路线进行随机仿真实验，得到的每种决策情况下的最大收益如下：

表 6 第三关中不同决策情况下的最大收益

最终期望收益 / 天气比例	晴天		高温	
晴天：高温	路线经过矿山	路线不经过矿山	路线经过矿山	路线不经过矿山
1:1	-1052.2	-380.656	-1095.6	-380.08
7:3	-908.945	-348.136	-945.916	-347.048
3:7	-1154.4	-411.544	-1225.2	-412.072

通过比较在不同天气比例下的最终期望收益，不论是晴天还是高温情况，路线不经过矿山的期望收益要小于路线经过矿山的情况。

因此我们推出：不论在怎样的天气状况安排下，经过矿山的路线的最终期望收益始终高于不经过矿山的路线。

5.2.3 第四关最佳行动策略的讨论

• 行动路线策略的分类

和第三关最佳策略的讨论方法类似，我们首先对第四关行动策略进行分类，根据矿山，村庄，终点的位置布局，我们首先从起点先到“村庄或者矿山”的情况分为两类。然后根据已有资源逐步增加去矿山的次数。

经过逐步尝试，我们将第四关行动路线策略分为 5 种，如下：

表 7 路线策略表

策略种类	路线安排情况
1	起点-村庄-矿山-终点
2	起点-村庄-矿山-村庄-终点
3	起点-村庄-矿山-村庄-矿山-村庄
4	起点-矿山-村庄-矿山-终点
5	起点-矿山-村庄-终点

• 对第四关游戏规则进行讨论

（1）玩家在游戏过程中需保证在规定时间内有足够的资源到达终点。

但由于初始资源购买量我们无法确定，因此我们考虑以每种线路决策情况下的最多资源消耗量作为每种情况下的初始资源量进行讨论。以保证无论怎样的天气安排，玩家

都能保证有足够的资源到达终点。

每种情况下最多资源消耗量为最坏天气（天气全为高温）的情况下资源消耗量。

表 8 每种策略最多资源消耗量

	水消耗量（箱）	食物消耗量（箱）	收益（元）
第一种策略	222	242	2470
第二种策略	287	301	1190
第三种策略	439	477	4475
第四种策略	450	488	6825
第五种策略	237	259	3175

（3） 我们对不同天气下挖矿所获取的收益进行了计算，结果如下：

表 9 不同天气下的挖矿收益

挖矿收益(元/天）	晴天挖矿资源消耗成本（元/天）	高温挖矿资源消耗成本（元/天）	沙暴挖矿资源消耗成本（元/天）
1000	285	300	450

根据挖矿收益表我们可以得出不论在哪种天气情况下，玩家挖矿都会有正向收益

•每种决策情况的最大收益仿真模拟：

（1）首先我们对三种天气出现的概率进行讨论。分“高温大概率出现，沙暴小概率出现，且沙暴概率出现上限值为 0.3。晴朗，高温，沙暴天气出现概率为 0.4,0.5,0.1。

然后基于上述规则对 4 中决策路线进行随机仿真实验，得到的每种决策情况下的最大期望收益如下：

表 10 第四关每种决策情况下的最大期望收益

策略种类	期望收益（元）
第一种策略	2470
第二种策略	1190
第三种策略	4475
第四种策略	6825
第五种策略	3175

因此我们推出：从起点-矿山-村庄-矿山-终点，这条路的期望收益最高，说明在天气情况不确定的情况下，应优选选择这种行动策略。

5.3 问题三模型的建立与求解

5.3.1 已知天气状况下一般策略的讨论

此问引入了多人游戏，与前几问的单人游戏相比，若有其他玩家与自身在同一区域，会使自身的收益发生改变。所以本问从仅有的以下两种行动状态来考虑：

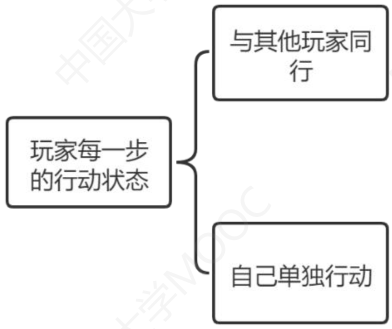

图 4 玩家行动状态分类

为获取更大收益，玩家应尽量避免与其他队友在同一区域同行。但由于不知道其他玩家的行动方案，我们无法实时做出行动决策。

因此我们利用“静态博弈”的思想，仅考虑自身利益最大化。根据题中所给规则，若有玩家与自身在同一区域，自身消耗量比单人消耗量多，收益量比单人收益量少，所以游戏全程k个玩家选择的路线均与自身一致为整个游戏过程中玩家行动策略最差的情况，选出在这种最差情况下的最优出行策略，即最差情况下收益最大的线路。

所以将问题转化为改变消耗量和收益量的第一问的模型，即目标函数写为

$$max = C - \left[4\mathrm{k}(P_1 \sum m_i X_i + P_2 \sum n_i Y_i)\right] + \frac{H}{k} \sum d_i \qquad (25)$$

资源状态转移方程写为：

（1）不同天气下水资源转移状态情况：

$$a_i = a_{i-1} - g_i(2kw_i + 2d_i + 1) + m_i X_i \qquad (26)$$

（2）不同天气下食物转移状态情况：

$$b_i = b_{i-1} - h_i(2kw_i + 2d_i + 1) + n_i Y_i \qquad (27)$$

5.3.2 第五关行动策略的讨论

第五关玩家共有两人，根据博弈论中的“纳什均衡”，两个玩家都是从自身角度考虑，使自己的收益最大化。由于两个玩家的选择会相互影响，对于每个玩家最优策略应当是风险最小的策略，使得最终博弈的行动策略是均衡解。而最差的收益情况为“两人选择同一行动方案”，此时挖矿基础收益为100元，行走时资源消耗量见下表：

表 11 行走时的资源消耗量

	晴天	高温
水消耗量（箱）	12	36
食物消耗量（箱）	16	36

玩家在不同天气下的挖矿收益见下表：

表 12 不同天气挖矿收益表

天气	晴天	高温
获取收益	-65 元/天	-540 元/天

由表中数据可知，在高温天气下挖矿会损失更多，因此我们应避免在高温天气下挖矿。同时权衡高温天气停留与行走消耗的资源和时间对挖矿情况的影响，我们对高温天气下是否原地停留也进行进一步的讨论。

我们在转移过程沿最短路径进行的条件下，将行动路线分为经过矿山和不经过矿山两种情况进行分析。

1、玩家经过矿场

根据第五关地区分布情况，我们先对第二天高温天气下玩家是否原地停留进行讨论：

（1） 玩家在第二天原地停留，此时玩家会在第 4 天到达矿场。然后对玩家是否在矿场挖矿进行考虑，如果玩家不进行挖矿，则玩家会在第六天到达终点，由于第七天后天气均为高温，玩家不能在矿山停留打工，因此，玩家从第七天开始向终点行进。

（2） 玩家不在第二天停留，此时玩家会在第 3 天到达矿山，第六天天气晴朗，玩家可以在第六天在矿山打工，从第七天开始向终点行进。

2、玩家不经过矿场

（3）玩家在第二天原地停留，然后沿最短路径向终点行进，在第四天到达终点。

（4）玩家第二天不停留，则直接沿最短路径向终点行进，在第三天到达终点。

我们对这四种行动策略的资源消耗量进行计算，见下表：

表 13 四种行动策略的资源消耗

行动策略	第一种	第二种	第三种	第四种
资源消耗（元）	-1235	-1485	-795	-980

所以，第五关选择的最优行动策略为：玩家不经过矿山，在第二天高温天气原地停留 1 天后，直接沿最小路径到终点。

5.3.3 第六关行动策略的讨论

每名玩家能够根据自身所在位置和剩余资源量得出当前所有可行决策，设每名玩家行动策略的决策集合为$Q_q(q=1,2,3)$，如从第一名玩家角度出发，选定某一决策时，其余玩家的不同决策组合会对该决策的期望收益产生影响，具体表现在选择的某一步决策相同，设第 q 位玩家的决策集合中的不同策略出现的概率为p_q，设在只有一名玩家情况下的收益为$C1_q$，在其他玩家影响下的收益为$C2_q$，则在该决策下的风险率R_q为：

$$R_q=\frac{C1_q-C2_q}{C1_q}\times 100\% \tag{28}$$

计算其余玩家决策组合出现的概率与此时造成的风险率的乘积，从中选出最大值，用于评估该策略下可能出现的最坏情况，即

$$\max\left(p_2 \cdot p_3 \cdot R_q\right) \tag{29}$$

对第一名玩家的决策集合Q_1中的每一种策略计算最大风险率，比较得出最大风险率最低的策略，即

$$min\left(max\ p_2.p_3.R_q\right) \tag{30}$$

认为此策略即为当前的最优策略。

六、灵敏度分析

我们改变了第三关玩家在不同天气下水和食物的基础消耗量，将基础消耗量扩大 2 倍见下表：

表 14 扩大 2 倍基础消耗量的表格

资源	基础消耗量（箱）	
	晴朗	高温
水	6	18
食物	8	18

表 15 资源基础消耗量变化前后的最终收益变化表

最终期望收益 / 天气比例	晴朗		高温	
晴天：高温	路线经过矿山	路线不经过矿山	路线经过矿山	路线不经过矿山
3:7	-1154.4	-411.544	-1225.2	-412.072
3:7(增大基础消耗量后的数据)	-1693.4	-689.98	-1778.8	-754.786

可以看出基础消耗量增大后，资源约束和时间约束更加紧张，各情况下的期望收益都有所减少，但不经过矿山时仍为较优路线。

七、模型的评价与改进

7.1 模型的评价

7.1.1 模型的优点

（1）研究多个玩家的行动策略时运用了博弈模型，考虑了各自目标和整体均衡性。

（2）多步决策模型与贪婪算法相结合，对每一步进行优化，最后使得全局最优。

7.1.2 模型的缺点

（1）天气分布为客观存在，进行仿真模拟时所取概率不能说明全部情况。

7.2 模型的改进

（1）针对不同天气状况，作出各种行动策略的分析，不硬性规定。

九、参考文献

[1]百度文库：https://baike.baidu.com/item/纳什平衡；
[2]姜启源，谢金星，叶俊 主编 数学模型(第三版)[M].北京:高等教育出版社.2009；

附录：（以下代码基于 MATLAB2018b 版本）

第一题：

```
clc;clear;
ls=[1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
    1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
    0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
    0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
    0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
    0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
    0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
    0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0
    0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0 1
    0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
    0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0
    0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
    1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1];
for i=1:27
    for j=1:27
        if(ls(i,j)==0)
            ls(i,j)=inf;
        end
    end
end

%起点到矿山
w=ls;start=1;terminal=12;
[min,path]=Dijkstraf(w,start,terminal)

%矿山到终点
```

```
w=ls;start=12;terminal=27;
[min,path]=Dijkstraf(w,start,terminal)
```
第二题：
```
clc;clear;
n=10000;
times=0;
E1=0;E2=0;
for v=1:n
%模拟天气,开始晴朗
%d=round(rand(1,10));
alphabet=[0 1];prob=[0.3 0.7];
d=randsrc(9,1,[alphabet; prob])';
d=[0,d];
f1=[6 18];%水
f2=[8 18];%食物
Cost1=1350;%在高温就走的规则下算出的上限，90kg+90kg
Cost2=810;%不去矿山
%去矿山
cost1=0;
cost2=0;%食物的累计消耗 kg
for i=1:3
    if d(i)==0
        cost1=cost1+2*f1(1);
        cost2=cost2+2*f2(1);
    elseif d(i)==1
        cost1=cost1+2*f1(2);
        cost2=cost2+2*f2(2);
    end
end
%
cost11=0;cost22=0;
s=[];%记录 4-8 天时第一次出现高温的日期,即走
number=find(d==1);
ql=5;gw=0;

for u=1:length(number)
    if number(u)>=4&&number(u)<=8
        s=number(u);
        break;
    end
end
if(isempty(s))
    gw=5;
end
```

```
if(~isempty(s))
    gw=s(1)-4;
end

if(isempty(s))
    cost11=cost11+ql*3*f1(1);
    cost22=cost22+ql*3*f2(1);
    for k=9:10
        if d(k)==0
            cost11=cost11+2*f1(1);
            cost22=cost22+2*f2(1);
        elseif d(k)==1
            cost11=cost11+2*f1(2);
            cost22=cost22+2*f2(2);
        end
    end
end
if(~isempty(s))
    cost11=cost11+gw*3*f1(1);
    cost22=cost22+gw*3*f2(1);
    for t=(gw+4):(gw+5)
        if d(t)==0
            cost11=cost11+2*f1(1);
            cost22=cost22+2*f2(1);
        elseif d(t)==1
            cost11=cost11+2*f1(2);
            cost22=cost22+2*f2(2);
        end
    end
end
%% 1
if cost1+cost11>90 || cost2+cost22>90
    cost11=0;cost22=0;
    ql=ql-1;
    if(isempty(s))
        cost11=cost11+ql*3*f1(1);
        cost22=cost22+ql*3*f2(1);
        for k=8:9
            if d(k)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(k)==1
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
```

```
            end
        end
    end
    gw=gw-1;
    if(~isempty(s))
        cost11=cost11+gw*3*f1(1);
        cost22=cost22+gw*3*f2(1);
        for t=(gw+4):(gw+5)
            if d(t)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(t)==1
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
            end
        end
    end
end
%% 第二次改
if cost1+cost11>90 || cost2+cost22>90
    cost11=0;cost22=0;
    ql=ql-1;
    if(isempty(s))
        cost11=cost11+ql*3*f1(1);
        cost22=cost22+ql*3*f2(1);
        for k=7:8
            if d(k)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(k)==1
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
            end
        end
    end
    gw=gw-1;
    if(~isempty(s))
        cost11=cost11+gw*3*f1(1);
        cost22=cost22+gw*3*f2(1);
        for t=(gw+4):(gw+5)
            if d(t)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(t)==1
```

```
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
            end
        end
    end
end


if(isempty(s))
    pr1=-Cost1+ql*200+(90-cost1-cost11)*5/2+(90-cost2-cost22)*10/2;
else
    pr1=-Cost1+gw*200+(90-cost1-cost11)*5/2+(90-cost2-cost22)*10/2;
end

E1=E1+pr1;

%不去矿山
d=d(1:3);
cost1=0;
cost2=0;%食物的累计消耗 kg
for i=1:3
    if d(i)==0
        cost1=cost1+2*f1(1);
        cost2=cost2+2*f2(1);
    elseif d(i)==1
        cost1=cost1+2*f1(2);
        cost2=cost2+2*f2(2);
    end
end
pr2=-Cost2+(90-cost1)*5/2+(90-cost2)*10/2;
E2=E2+pr2;
end
E1=E1/n;
E2=E2/n;

clc;clear;
n=10000;
times=0;
E1=0;E2=0;
for v=1:n
%模拟天气,开始高温
%d=round(rand(1,10));
alphabet = [0 1]; prob = [0.3 0.7];
d=randsrc(9,1,[alphabet; prob])';
```

```
d=[1,d];
f1=[6 18];%水
f2=[6 18];%食物
Cost1=1190;%在晴朗就走的规则下算出的上限，78kg+80kg
Cost2=650;%不去矿山
%去矿山
cost1=0;
cost2=0;%食物的累计消耗 kg
for i=1:3
    if d(i)==0
        cost1=cost1+2*f1(1);
        cost2=cost2+2*f2(1);
    elseif d(i)==1
        cost1=cost1+2*f1(2);
        cost2=cost2+2*f2(2);
    end
end
%
cost11=0;cost22=0;
s=[];%记录 4-8 天时第一次出现高温的日期,即走
number=find(d==1);
ql=5;gw=0;

for u=1:length(number)
    if number(u)>=4&&number(u)<=8
        s=number(u);
        break;
    end
end
if(isempty(s))
    gw=5;
end
if(~isempty(s))
    gw=s(1)-4;
end

if(isempty(s))
    cost11=cost11+ql*3*f1(1);
    cost22=cost22+ql*3*f2(1);
    for k=9:10
        if d(k)==0
            cost11=cost11+2*f1(1);
            cost22=cost22+2*f2(1);
        elseif d(k)==1
```

```
            cost11=cost11+2*f1(2);
            cost22=cost22+2*f2(2);
        end
    end
end
if(~isempty(s))
    cost11=cost11+gw*3*f1(1);
    cost22=cost22+gw*3*f2(1);
    for t=(gw+4):(gw+5)
        if d(t)==0
            cost11=cost11+2*f1(1);
            cost22=cost22+2*f2(1);
        elseif d(t)==1
            cost11=cost11+2*f1(2);
            cost22=cost22+2*f2(2);
        end
    end
end
%% 1
if cost1+cost11>78 || cost2+cost22>80
    cost11=0;cost22=0;
    ql=ql-1;
    if(isempty(s))
        cost11=cost11+ql*3*f1(1);
        cost22=cost22+ql*3*f2(1);
        for k=8:9
            if d(k)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(k)==1
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
            end
        end
    end
    gw=gw-1;
    if(~isempty(s))
        cost11=cost11+gw*3*f1(1);
        cost22=cost22+gw*3*f2(1);
        for t=(gw+4):(gw+5)
            if d(t)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(t)==1
```

```
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
            end
        end
    end
end
```

```
%% 第二次改
if cost1+cost11>78 || cost2+cost22>80
    cost11=0;cost22=0;
    ql=ql-1;
    if(isempty(s))
        cost11=cost11+ql*3*f1(1);
        cost22=cost22+ql*3*f2(1);
        for k=7:8
            if d(k)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(k)==1
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
            end
        end
    end
    gw=gw-1;
    if(~isempty(s))
        cost11=cost11+gw*3*f1(1);
        cost22=cost22+gw*3*f2(1);
        for t=(gw+4):(gw+5)
            if d(t)==0
                cost11=cost11+2*f1(1);
                cost22=cost22+2*f2(1);
            elseif d(t)==1
                cost11=cost11+2*f1(2);
                cost22=cost22+2*f2(2);
            end
        end
    end
end
```

```
if(isempty(s))
    pr1=-Cost1+ql*200+(78-cost1-cost11)*5/2+(80-cost2-cost22)*10/2;
else
    pr1=-Cost1+gw*200+(78-cost1-cost11)*5/2+(80-cost2-cost22)*10/2;
end

E1=E1+pr1;

%不去矿山
d=d(1:3);
cost1=0;
cost2=0;%食物的累计消耗 kg
for i=1:3
    if d(i)==0
        cost1=cost1+2*f1(1);
        cost2=cost2+2*f2(1);
    elseif d(i)==1
        cost1=cost1+2*f1(2);
        cost2=cost2+2*f2(2);
    end
end
pr2=-Cost2+(78-cost1)*5/2+(80-cost2)*10/2;
E2=E2+pr2;
end
E1=E1/n;
E2=E2/n;
```