

“穿越沙漠”问题的最优化策略研究

摘要

穿越沙漠游戏是一个涵盖动态优化、最短路径等多方面知识的策略问题，本文利用相关理论知识，由简至繁分析不同情况下玩家进行游戏的最优策略，建立了关于该游戏最优策略的数学模型，并探究了该模型在现实社会中可能的相关应用以及其蕴含的广泛社会意义。

对于问题一，我们将地图转化为数学意义上的图，针对不同的需求分别利用 **Floyd 算法**和 **Dijkstra 算法**求得所需的最短路径，由此推出玩家的最优策略，并针对不同情况进行细节上的讨论，从而得到玩家的最优收益。根据建立的一般策略，我们对具体的第一关和第二关进行了模型的应用并求出了具体的数据，验证了模型的可行性。

对于问题二，在问题的复杂度变高时，为方便讨论，我们给出了三个假设。为了使分析更加严谨条理，我们增加了一些限制条件，并在研究的过程中逐渐去除。针对天气的不确定性，我们引入了经验概率和合理冗余，使得模型更具完备性。

对于问题三，玩家的增加使得问题的自由度陡升，我们在分析时依据问题一、二的模型逐步深入，针对多玩家的情况引入了**博弈论**，并据此展开分析，得出了多玩家情况下的异步策略，并在两个具体的问题（第五关、第六关）中验证了策略的合理性。

关键字： 动态优化 最短路径 Floyd 算法 Dijkstra 算法 博弈论

一、问题重述

1.1 问题的提出

穿越沙漠是一种基于玩家视角的生存小游戏，玩家须在综合考虑多种外界因素后选择自己的决策行为。

玩家首先会拿到一张地图，图中标示了起点、终点、村庄和矿山，中途可经过的各个区块也用标号注明。每名玩家拥有一个背包和若干现金，其中现金可用来购买所需的资源，背包则决定了能带在路上的资源上限。中途的村庄可购买资源补给，而矿山能为玩家提供额外的财富来源。玩家不同的行为抉择和天气情况的变化均会影响玩家的消耗，在到达终点后，剩余的资源会折算成现金计入最终剩余。玩家需要在规定的时间内到达终点，同时获取尽可能高的收益。

穿越沙漠类似于大富翁游戏，同时会涉及组合优化中的背包问题等。虽然游戏开放度较高，但它规定了参与者能够行动范围和时间上限，因而存在最优化的方法。玩家需要提前规划好自己的行动路线、背包中的资源配置和行动策略。在后面的问题中，参与玩家数量增多，所给信息减少，情况更为复杂，需要运用多种方法找到一般性的解决方案。不过作为一个生存游戏，参与者将以最高优先级考虑的还是个人的生存问题，即现有资源能否支持其在规定时间内到达终点。

基于题目所给的基本规则以及附录中的地图、天气和背包参数等信息，本文将研究以下问题：

1. 基于单个玩家视角，所有日期的天气情况已知，根据初始资金和资源价格、消耗信息，建立模型分析从起点到终点的最优化方案。该问题所涉及的地图有两种，每种地图情况下都包含具体到每一天的资源数量和行走路线；
2. 基于单个玩家视角，当不存在沙暴天气，且具体天气情况未知时，考虑其中可能的天气情况及对玩家的影响；
3. 根据上述讨论分类，综合比较最终盈余的资金，得到玩家在其理性预期下在地图三上的行动路线；
4. 背包和资源的基本信息不变，行走总时间和地图发生变化，同时所有天气均有可能出现且无法提前知晓。综合以往经验对天气情况作进一步的推测，在复杂场景时作简化处理。据此列出可能的行动规划，根据模型筛选出较为优秀的策略作为一般情况下的最佳策略。
5. 基于两个或三个玩家的视角，考虑玩家之间可能的各种互动场景，根据博弈论分析玩家的配合战术，考虑自己的同时分析其他玩家的最优化决策（竞争或合作）。其他玩家不同的决策会影响到己方玩家能够获得的收益以及消耗；

6. 模拟对方可能的策略后，综合之前对天气情况的处理经验，建立模型分析玩家应采取的策略方法。

1.2 问题的意义

穿越沙漠作为一种考虑最优决策的小游戏，玩家通过把握有限的资源和时间，争取获得最大化的收益。与之类似的是背包问题，通过组合优化在背包中装下总价值最高的物品。最优化问题的应用场景十分广泛，在数学、商业和工程学中都需要相关知识。

近期，有关外卖骑手的配送问题被广泛讨论，其本质也能在这个最优化游戏中找到一点影子。这件事的起源是一篇名为《外卖骑手，困在系统里》的新闻故事，它讲述了外卖平台不断优化的算法，是如何造成“外卖员已成高危职业”这一社会问题。为了提高配送速度，外卖平台的算法不断减少骑手的配送时间，并且给“顺路”的骑手尽可能多地派单，力图达到广告中“送啥都快”的效果。可这带来的后果是，如果不采取闯红灯、逆行等违反交通法规的方式，骑手根本无法将外卖在规定时间内送至目的地。

为了便于分析上述问题，我们把外卖骑手一次性配送多单的现实情形模拟为穿越沙漠游戏中的一次穿越。骑手须要在规定时间内配送完好几单，类似于玩家要在有限日期内穿越沙漠。骑手在不同的状态下消耗时间不同，例如等餐、骑行以及等电梯上楼会花掉不同的时间，而玩家在原地停留和行走时也会消耗不同的资源。挖矿是游戏中唯一可以获利的方式，不过这会消耗许多资源，同时还会花费宝贵的时间。同样道理，骑手每多接一单也能直接获得更多收益，但存在着交通事故和超时配送的罚款等风险。只有根据具体情况合理分析，才能获得最佳的策略。

现实情况是，很多时候骑手本身不愿意接单，而外卖平台却依旧不断地塞单，这就类似于逼迫玩家使劲挖矿导致无法到达终点。挖矿看似是一项稳赚不赔的行动，可前提应为剩余资源足够到达目的地。如果基本要求没有满足，赚再多的钱也将没有任何意义。因而，外卖平台应把保证骑手生命健康、不破坏交通秩序视为商业底线，不能因为加快一两分钟不顾其所带来的严重后果。顾小利而失大局，这终究是不可取的。着眼未来，把握大势，方能行稳致远。

最优化问题看似简单，在实际操作的过程中会面对诸多复杂情况。本问题创新性十足，通过饶有趣味的游戏方式给予每一个参赛者深入思考最佳决策行为的机会。在现代社会高速发展的进程中，每一份资源都有重要的作用。因而，不论个人、企业还是国家，在做决策的时候一定要将条件和后果考虑清楚。合理预测，大胆践行，如此才能取得长远的发展。

二、问题分析

问题一是在条件确定的情况下求出数值最优解。玩家已知的信息包括具体到某一天的天气情况、各种资源的价格和重量以及不同决策的消耗或收益等。由于在本问题的框架下不需要做分类讨论和一般性分析，我们只须完成特值解，通过对算出的具体数值进行比较找出最大的剩余。

问题二在天气不确定性方面提高了难度，其中，第四关对沙暴的加入进一步考察了玩家分类讨论和期望推测的能力。由于第三关的地图中没有出现村庄，玩家需要在开始就准备充足的资源以应对可能出现的晴朗或高温天气。由于玩家不会因为天气原因暂缓前进，所要重点考虑的即为挖矿的收益问题。这也是本题的突破口之一。第四关则加入了沙暴的天气，因而要做包含最坏情况的策略打算。由于沙暴出现概率较低，天气的期望参考一、二关的具体情形较为科学。同时，第四关为玩家提供了村庄，重点分析的内容便是前往村庄补给与路途消耗以及恶劣天气影响之间的权衡。在对极端天气作简化分析后获得一般性结论。

问题三最大的特色是将游戏改为多玩家系统，这需要运用博弈论的相关知识。在第五关的双玩家条件下，每个人在考虑自己最优决策的同时，应考虑对方的可能行动。为了避免出现同时前往同一地点或同时挖矿的情况，玩家应采取相比之前更加差异化的决策。第六关所给的三玩家情况分析第五关异曲同工，值得注意的还是天气情况的变化。为了减少计算量，应同时对天气情况的其余玩家决策作简化分析，在玩家行动尽可能不同的情况下得到一般性的最优解。

三、符号说明

符号	意义
T	游戏时间（天）
p_{water}	水的基准价格（元/箱）
p_{food}	食物的基准价格（元/箱）
b_{water}	每箱水的质量（千克）
b_{food}	每箱食物的质量（千克）
r	基础收益（元）

* 未列出符号及重复的符号以出现处为准

四、问题一的模型建立与具体求解

4.1 问题的分析

问题一要求我们在只有单个玩家的情况下给出使玩家收益最大化的策略，并假设玩家知晓整个游戏时间段内每天的天气状况。首先，我们注意到在非必要情况（沙暴天气）时停留是只有损失没有收益的，所以玩家除沙暴天气外的每一天要么在区域之间按移动，要么已经到达矿山在挖矿。其次，我们要考虑在矿山挖矿的收益与消耗的资源之差，即玩家在矿山挖矿的净收益。只要玩家挖矿取得的净收益大于玩家在当天天气情况下基础消耗量的负值，那么玩家挖矿都是要好于不挖矿的，否则玩家当天不挖矿。

我们的目标是使玩家在游戏结束后收益达到最大值，因此我们要使玩家在矿山的时间尽可能多，而在路上的时间尽可能少。由于地图是已知的，我们可以通过 Dijkstra(迪杰斯特拉) 算法或者 Floyd（弗洛伊德）算法确定从起点到矿山、从矿山到终点的最短路径，又每天的天气状况已知，我们进而得到整个游戏过程中需要消耗的总资源。由题目所给条件可知，起点处购买资源的价格最便宜，所以我们尽可能在起点处购买好所需的全部资源。如果超过负重上限，我们选择一条经过村庄的最短路径，判断是否超过先前的最短路径，计算绕路损失的资源是否超过矿山与村庄往返所需的资源，并依此给出前往村庄补充资源的最优策略。

最后，我们计算出玩家在给出策略下最终获得的资金，并在具体情形下对模型进行验证。

我们的思路流程如图 1。

4.2 模型的建立

4.2.1 矿山、村庄均只有一个的情形

我们将地图转换成一个图 $G(V, E)$ ，其中 V 是顶点集， E 是边集，图中的点为地图中的区域，若地图中区域相邻，则在图中对应的点之间连一条边。我们规定 v_1 是起点， v_n 是终点， v_c 是村庄， v_k 是矿山。我们首先确定起点经矿山到终点的最短路径。

定义该无向图的权值矩阵 $\mathbf{W} = (w_{ij})_{n \times n}$ 为

$$w_{ij} = \begin{cases} 1, & \text{若 } v_i \text{ 到 } v_j \text{ 有边} \\ \infty, & \text{若 } v_i \text{ 到 } v_j \text{ 没有边} \\ 0, & \text{若 } i = j \end{cases}$$

我们利用 Floyd 算法求解起点 v_1 到矿山 v_k 的最短路径。步骤如下：

第一步 求出图 G 的权值矩阵 \mathbf{W} ，定义 d_{ij} 为顶点 v_i 到顶点 v_j 之间的最短距离，令 $d_{ij} = w_{ij}$ ， $l = 1$

第二步 对全体 i, j ，若 $d_{il} + d_{lj} < d_{ij}$ ，则令 $d_{ij} = d_{il} + d_{lj}$

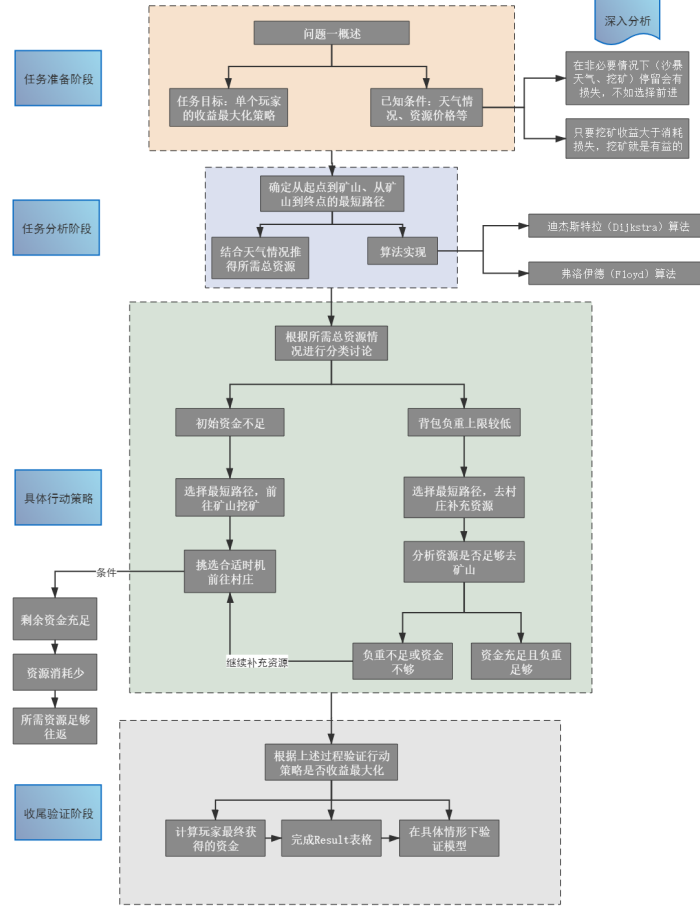


图 1 问题一分析流程图

第三步 若 $d_{ii} < 0$ ，则出现一条含有顶点 v_i 的负回路，程序停止；若 $l = k$ ，程序停止。否则回到第二步。

用同样的方法可以求出矿山 v_k 到终点 v_n 以及矿山 v_k 到村庄 v_c 的最短路径。最短路径的长度即为玩家走这段路至少要花费的天数（如果未出现沙暴天气）。

又玩家已知整个游戏时段内每天的天气情况，则玩家全过程需要的资源负重、资源价格可以求得，分别设为 B, V 。已知玩家的负重上限 H_{max} 和初始资金 F_0 ，下面进行分类讨论：

1. 若 $H_{max} \geq B$ 且 $F_0 \geq V$ ，则整个游戏时段不需要去村庄 v_c 。
2. 若 $H_{max} < B$ 但 $F_0 \geq V$ ，则在已求出的最短路径中挑选一个经过村庄 v_c 的路径以便补充资源。若不存在这样的路径，则计算绕路造成的最小可能¹损失（包含绕路的资源消耗与绕路导致不能挖矿而减少的收益）与从矿山 v_k 到村庄 v_c 补充资源造成的最小可能损失（包含往返的资源消耗与不能挖矿减少的收益）作比较，选择损失更少的方案补充资源。
3. 若 $H_{max} \geq B$ 但 $F_0 < V$ ，则计算矿山 v_k 到村庄 v_c 补充资源的最小可能损失（包括

¹即挑选天气较好的时段

从矿山到终点的最短路径包含村庄 v_c 且初始资源足够在挖矿结束后到达村庄 v_c 的情况)。

4. 若 $H_{max} < B$ 且 $F_0 < V$ ，同上。

在上述情况中，如果最小可能损失不小于补充资源带来的挖矿净收益 (= 挖矿收益 - 补充资源价格)，那么我们去村庄 v_c 补充资源，直接前往终点 v_n 。如果起点 v_1 到终点 v_n 的最短路径不包括矿山 v_k ，且去矿山 v_k 所带来的收益 (= 挖矿收益 - 挖矿消耗 - 绕路消耗) 不大于零，那么我们去矿山 v_k ，直接前往终点 v_n 。

设起点 v_1 到矿山 v_k 所需天数为 t_1 ，消耗资源为 b_1 ；矿山 v_k 到终点 v_n 所需天数为 t_2 ，消耗资源为 b_2 ；补充资源带来的最小可能损失为 D 。为方便表示，我们假设挖矿的基础收益足够大，那么以上四种情况下的最终收益 R 分别为

$$1. R = F_0 - V + r(T - t_1 - t_2)$$

2. 设 $B - H_{max}$ 重量的资源基准价格为 V' ，则有

$$R = F_0 - (V - V') + r(T - t_1 - t_2) - D - 2V' = F_0 - V - V' - D + r(T - t_1 - t_2)$$

即

$$R = F_0 - V - V' - D + r(T - t_1 - t_2)$$

$$3. R = r(T - t_1 - t_2) - F_0 - 2(V - F_0) - D = R = F_0 - 2V - D + r(T - t_1 - t_2)$$

即

$$R = F_0 - 2V - D + r(T - t_1 - t_2)$$

4. 设需要补充的资源基准价格为 V'' ，则有

$$R = F_0 + r(T - t_1 - t_2) - (V - V'') - 2V'' - D = F_0 - V - V'' - D + r(T - t_1 - t_2)$$

即

$$R = F_0 - V - V'' - D + r(T - t_1 - t_2)$$

以上即为玩家可能获得的最大收益。

当挖矿的基础收益不够大时，我们要么提前前往终点 v_n ，要么由起点 v_1 直接前往终点 v_n ，两种情况均较为简单，在此不做具体讨论。

4.2.2 矿山、村庄多于一个的情形

为方便下面的分析，我们首先作出如下的假设：

假设 1 各个矿山、村庄完全相同。

假设 2 基础收益足够大，使得矿山是必须经过的顶点。

假设 3 玩家的资源消耗足够多，使得村庄是必须经过的顶点。

由于玩家知道每一天的天气情况，因此确定路径后全程的最小可能²资源消耗进而也可求出，故问题转化为求使玩家在矿山挖矿时间最长的路径。另一方面，由假设 1 知各个矿山、村庄完全相同，所以为补充资源而消耗的行走消耗量可以更有“意义”。由于玩家最终必须要到达终点，所以我们补充资源时选择距离终点更近的村庄会减少完成挖矿后的路程，同理补充完资源后我们也选择距离终点更近的矿山，依此类推。

设 $K = \{v_{k_1}, v_{k_2}, \dots, v_{k_s}\}$ 为矿山顶点的集合， $C = \{v_{c_1}, v_{c_2}, \dots, v_{c_t}\}$ 为村庄顶点的集合。我们采用 Dijkstra 算法来求我们需要的最短路径。以起点 v_1 到集合 K 中顶点的最短路径为例：

第一步 设 G_{K_1} 为包含 $K \cup \{v_1\}$ 的最大子图，考察 G_{K_1} ，令 $l(v_1) = 0$ ，对 $v \neq v_1$ ，令 $l(v) = \infty$ ， $S_1 = \{v_1\}$ ， $i = 1$ 。

第二步 对每个 $v \in \overline{S_i}$ ($\overline{S_i} = G_{K_1} \setminus S_i$)，用 $\min_{u \in S_i} \{l(v), l(u) + w(uv)\}$ 替换 $l(v)$ ，当 u, v 不相邻时， $w(uv) = \infty$ 。计算 $\min_{v \in \overline{S_i}} \{l(v)\}$ ，把该达到最小值的顶点并入 S_i ，记作 S_{i+1} 。

第三步 若 $i = |G_{K_1}| - 1$ ，程序停止；若 $i < |G_{K_1}| - 1$ ，用 $i + 1$ 代换 i ，回到第二步。

我们之所以采用 Dijkstra 算法而非 Floyd 算法，是因为 Dijkstra 算法在求出一个顶点到图中其他顶点的最短路径的同时，还会记录最短路径的长度，方便我们进行比较。

在上述步骤后，我们选取 K 中最短路径长度最短的顶点。当需要补充资源时，我们对该顶点和集合 C 以及终点 v_n 和集合 C 进行类似的步骤，选取到该顶点与终点 v_n 最短路径长度之和最小的那个村庄。下一个矿山的选择与这个村庄的选择类似。这样我们就得到了一条优化后的最短路径，在每次移动区域时选择较好的天气时段，即可使玩家的收益最大化。

值得注意的一点是，在资源充足的情况下，我们不移动，因为矿山是完全相同的，移动没有意义。

4.3 具体问题的求解（“第一关”与“第二关”）

4.3.1 第一关的求解

由图可知，矿山不在起点与终点的连线上。分析计算可知共三种最有可能成为最优解的情况，不挖矿，仅一段时间持续挖矿，有两段时间持续挖矿。又因为天气状况已知，我们完全有理由认为，根据效益最大化，最后物资剩余为 0。

1. 不挖矿，直接由起点抵达终点，沿最短路径 $1 \rightarrow 25 \rightarrow 26 \rightarrow 27$ ，仅需三日抵达。过程如下：

- 1) 从起点出发，携带 42 箱水和 38 箱食物，负重 202kg，支出 $210 + 380 = 590$ 元。
- 2) 三日后到达终点，此时物资无剩余，保留资金总计 $10000 - 590 = 9410$ 元。

²挑选相对较好的天气

2. 仅一段时间持续挖矿，过程如下：
- 1) 从起点出发，携带 180 箱水和 330 箱食物，负重 1200kg，支出 4200 元。
 - 2) 行走八日后达到村庄，补充 163 箱水，食物剩余 232 箱，此时共携带 245 箱水和 232 箱食物，负重 1199kg, 支出 1630 元。
 - 3) 继续行走两日抵达矿山，挖矿七天，驻留一天，收入 7000 元。
 - 4) 选择最优路径 12→13→15→9→21→27 抵达终点，途经村庄，剩余 21 箱食物，补充 36 箱水和 19 箱食物，此时共携带 36 箱水和 40 箱食物，负重 188kg, 支出 740 元。抵达终点无物资剩余，最终保留资金总计 $10000+7000-4200-1630-740=10430$ 元。
3. 有两段时间持续挖矿，过程如下：
- 1)——3) 与情况二一致
 - 4) 选择矿山与村庄之间的最优路径 12→13→15 到达村庄，剩余 21 箱食物，补充 157 箱水和 140 箱食物，此时共携带 157 箱水和 161 箱食物，负重 793kg, 支出 4370 元。
 - 5) 行走两日回到矿山，挖矿三日，收入 3000 元。
 - 6) 沿最优路径抵达终点，抵达终点的时间恰好为游戏最后一日，无物资剩余。最终保留资金总计 $10000+7000+3000-4200-1630-4370=9800$ 元。

比较三种情况可得，情况二为最优策略，最终保留资本为 10430 元。

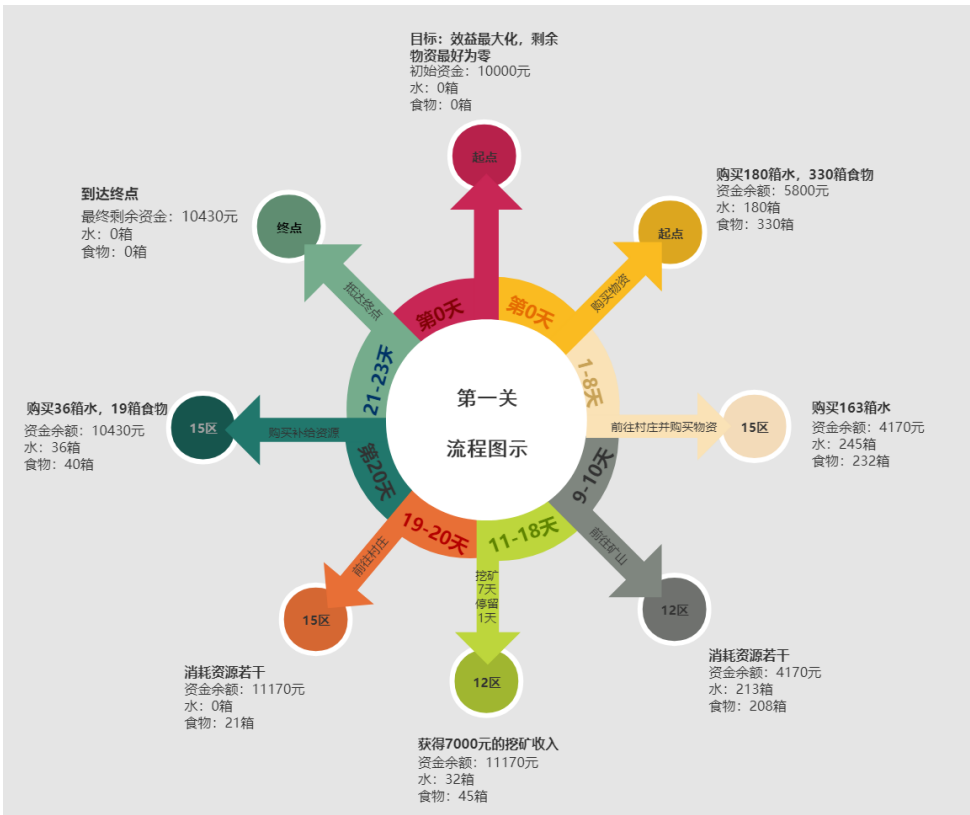


图 2 第一关图解

4.3.2 第二关的求解

由图可知共有两处矿山，经分析计算可得共六种可能成为最优解的情况，不挖矿，在矿山 30 仅一段时间持续挖矿，先在矿山 30 挖矿再去矿山 55 挖矿（又分两种情况），在矿山 55 仅一段时间持续挖矿，在矿山 55 有两段时间持续挖矿。又因为天气状况已知，我们完全有理由认为，根据效益最大化，最后物资剩余为 0。

1. 不挖矿，直接沿最优路径由起点抵达终点。过程如下：

- 1) 从起点出发，携带 182 箱水，170 箱食物，负重 886kg，支出 2160 元
- 2) 十四日后到达终点，此时物资无剩余，保留资金总计 $10000-2160=7840$ 元。

2. 在矿山 30 仅一段时间持续挖矿，过程如下：

- 1) 从起点出发，携带 223 箱水和 265 箱食物，负重 1199kg，支出 3765 元。
- 2) 行走 9 日后达到矿山，挖矿四日，收入 4000 元。
- 3) 选择最优路径 $30 \rightarrow 39 \rightarrow 47 \rightarrow 56 \rightarrow 64$ 抵达终点，途经村庄，剩余 56 箱食物，补充 68 箱水，此时共携带 68 箱水和 56 箱食物，负重 316kg，支出 680 元。
- 抵达终点无物资剩余，最终保留资金总计 $10000+4000-3765-680=9555$ 元。

3. 先在矿山 30 挖矿再去矿山 55 挖矿，过程如下：

I

- 1) 从起点出发，携带 130 箱水和 405 箱食物，负重 1200kg，支出 4700 元。
- 2) 行走 10 日后达到村庄，并在村庄停驻一日，补充 189 箱水，食物剩余 283 箱，此时共携带 189 箱水和 283 箱食物，负重 1133kg，支出 1890 元。
- 3) 继续行走 1 日抵达矿山，挖矿六天，收入 6000 元。
- 4) 行走一日到村庄补充资源，补充 196 箱水和 86 箱食物，此时共携带 196 箱水和 200 箱食物，负重 988kg，支出 3680 元。
- 5) 行走两日到达矿山 55，挖矿七天，收入 7000 元。
- 6) 沿最优路径抵达终点，抵达终点无物资剩余，最终保留资金总计 $10000+6000+7000-4700-1890-3680=12730$ 元。

II

- 1) 从起点出发，携带 184 箱水和 324 箱食物，负重 1200kg，支出 4160 元。
- 2) 行走 9 日抵达矿山 30，挖矿两天，收入 2000 元。
- 3) 行走 1 日到达村庄，补充 150 箱水，食物剩余 154 箱，此时共携带 150 箱水和 154 箱食物，负重 758kg，支出 1500 元。
- 4) 行走两日，抵达矿山 55，挖矿两四天，收入 4000 元。
- 5) 行走一日到达村庄，补充 201 箱水，和 187 箱食物，此时共携带 201 箱水和 207 箱食物，负重 1017kg，支出 5750 元。
- 6) 行走一日回到矿山，挖矿八天，收入 8000 元。
- 7) 沿最优路径抵达终点，抵达终点无物资剩余，最终保留资金总计 $10000+2000+4000+8000-$

$4160-1500-5750=12590$ 元。

4. 在矿山 55 仅一段时间持续挖矿，过程如下：

- 1) 从起点出发，携带 163 箱水和 355 箱食物，负重 1199kg，支出 4365 元。
- 2) 行走 12 日后达到村庄，补充 223 箱水，食物剩余 211 箱，此时共携带 223 箱水和 211 箱食物，负重 1091kg，支出 2230 元。
- 3) 继续行走 1 日抵达矿山，挖矿八天，收入 8000 元。
- 4) 沿最优路径抵达终点，抵达终点无物资剩余，最终保留资金总计 $10000+8000-4365-2230=11405$ 元。

5. 在矿山 55 有两段时间持续挖矿，过程如下：

- 1) 从起点出发，携带 156 箱水和 366 箱食物，负重 1200kg，支出 4440 元。
- 2) 行走 12 日后达到村庄，补充 158 箱水，食物剩余 222 箱，此时共携带 158 箱水和 222 箱食物，负重 918kg，支出 1580 元。
- 3) 继续行走 1 日抵达矿山，挖矿五天，收入 5000 元。
- 4) 行走一日到村庄补充资源，补充 201 箱水和 125 箱食物，此时共携带 201 箱水和 207 箱食物，负重 1017kg，支出 4510 元。
- 5) 行走一日回到矿山，挖矿八天，收入 8000 元。
- 6) 沿最优路径抵达终点，抵达终点无物资剩余，最终保留资金总计 $10000+5000+8000-4440-1580-4510=12470$ 元。

综上分析，情况 3I 为最佳策略，最终保留资本为 12730 元。

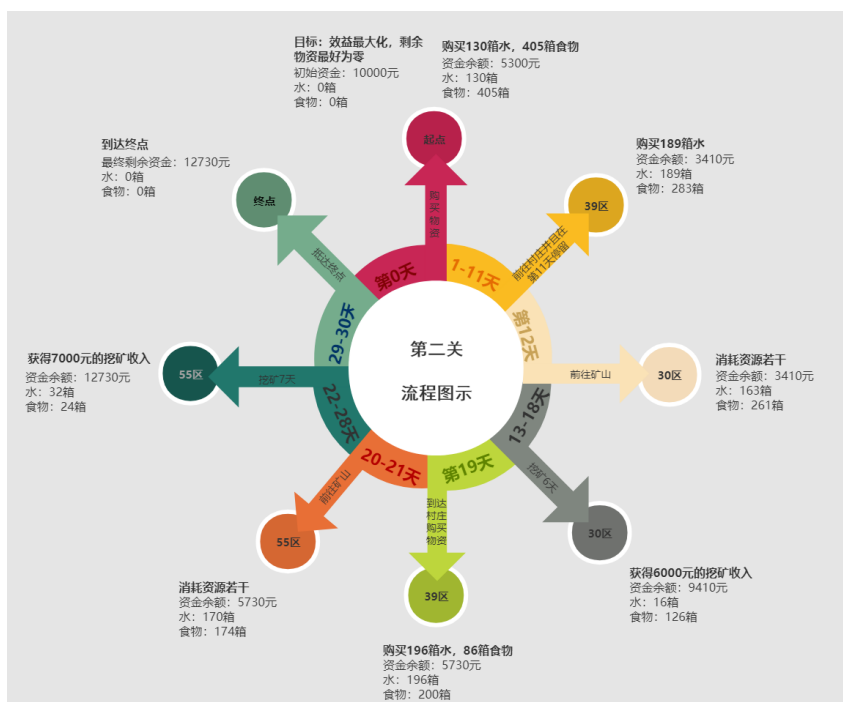


图 3 第二关图解

五、问题二的模型建立与具体求解

5.1 问题的分析

问题二与问题一相比，只有天气已知的条件发生改变，变为只知道当天的天气情况。这意味着我们无法提前对整个游戏时段做出最优化的安排，只能对部分时段做出局部最优的安排。首先，我们要判断矿山是否仍是必须经过的顶点。当天气情况变得不确定时，只有基础收益足够大，矿山才值得去，否则在极端情况下会有无法在游戏时段内到达终点的风险。只有在矿山的净收益大于绕路导致的资源消耗时，我们才需要前往矿山，否则我们直接前往终点。其次，我们在起点处购买资源的策略也需要改变。在确定好路径后，我们要按照最极端的天气情况来购买资源，为自己留下冗余。

如果我们根据过往的天气情况来预测未来可能出现天气的概率，并根据预测的概率制定起点处购买资源的策略，那我们便可以减少冗余，从而使玩家的收益更大。另外，如果考虑沙暴天气，那么沙暴出现在不同的时段也会对玩家造成不同的影响，需要进行分类讨论。

为了方便模型的建立，我们将按照如下的步骤逐个讨论：

- 不考虑沙暴，不预测天气的概率，不存在村庄
- 不考虑沙暴，预测天气的概率，不存在村庄
- 考虑沙暴，预测天气的概率，存在村庄

我们的思路流程如图 4。

5.2 模型的建立

假设 矿山、村庄至多各存在一个。

5.2.1 不考虑沙暴的情形

首先我们讨论是否前往矿山。假设不存在村庄，将 T' 定义为玩家进行游戏的时间。设路途中（不在矿山的时间段）消耗的资源价值为 V_R ，其余时段消耗的资源价值为 V_K ，完成由起点直接到终点的最短路径消耗的资源价值为 V'_R ，由上一节的分析可得前往矿山的条件为

$$r(T' - t_1 - t_2) - V_R - V_K > -V'_R$$

值得注意的是，在进一步的分析中，该式的误差是可以部分消除的。在不考虑沙暴的前提下，路径的长度在数值上就是走完整个路径所需的天数。设起点到矿山的最短路径长 l_1 ，起点到终点的最短路径长 l_2 。考虑 $l_1 - l_2$ ：

1. 若 $l_1 - l_2 > 0$ ，则从第 l_2 天起考察 $r(T' - t_1 - t_2) - (V_R - V'_R) - V_K > 0$ 是否成立， $V_R - V'_R$ 部分的精确度提高。

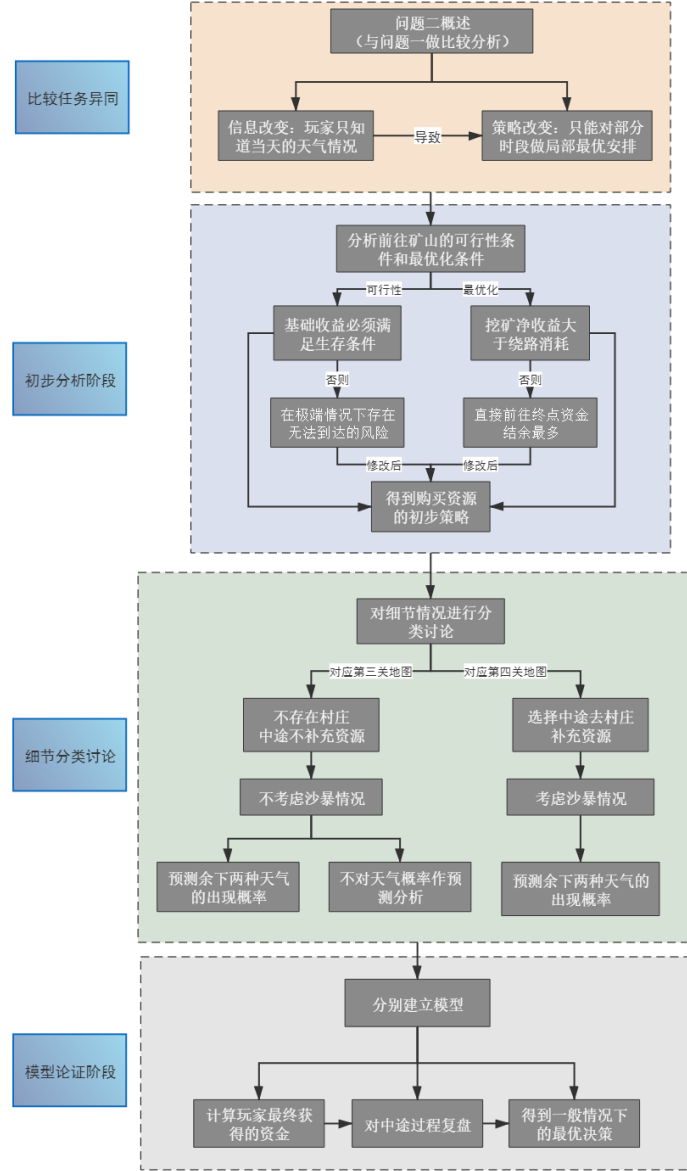


图 4 问题二分析流程图

2. 若 $l_1 - l_2 < 0$, 则玩家第 l_1 天后已经开始挖矿。又玩家挖矿的消耗是 3 倍基础消耗量, 行走的消耗量是 2 倍基础消耗量, 那么到达矿山后 $l_2 - l_1$ 天的消耗即为行走 $l_2 - l_1$ 天与原地休息 $l_2 - l_1$ 天之和, 故可认为玩家到达矿山后 $l_2 - l_1$ 天的消耗是基础消耗量, 考察 $r(T' - t_1 - t_2) - (V_R - V'_R) - V_K > 0$ 是否成立, $V_R - V'_R$ 部分的精确度提高。

接着我们考虑初始物资的购买。 x 表示基础消耗量, 角标 1, 2, 3 分别对应天气晴朗、高温、沙暴。假设不存在村庄, 我们按照高温天气下行走的消耗量购买资源, 则玩家可能的游戏时段为

$$\min \left\{ T, \min \left\{ \left\lfloor \frac{B}{2(x_{food2}b_{food} + x_{water2}b_{water})} \right\rfloor, \left\lfloor \frac{F_0}{2(x_{food2}p_{food} + x_{water2}p_{water})} \right\rfloor \right\} \right\}$$

我们选择高温天气下的行走消耗是一种折衷的选择, 在为自己留下一定冗余的同

时，使得在游戏结束后冗余带来的损失尽可能小。我们依据此可能时段来初步制定我们的策略。

在矿山挖矿时，我们的退出策略（即资源剩余多少时停止挖矿，前往终点）为：剩余资源可以在高温天气下由矿山到终点的最短路径到达终点。在终点处，如果存在剩余资源，我们进行兑换。从而玩家最终的收益为初始资金 - 购买资源消耗的资金 + 挖矿收益 + 退回剩余资源的资金。这是玩家相对最优的策略。

另外，我们注意到由于冗余的存在，玩家的游戏时段很大概率会比计算得到的可能游戏时段长。为了减少冗余带来的损失，我们引入预测的天气概率。根据以往的天气情况，可以得到经验高温概率 $a\%$ ，则晴朗概率为 $1 - a\%$ ，再引入一个新的变量——合理冗余 ε 。每日消耗量按行走消耗量计算，则可能游戏时段的计算式中，两分数的分母分别变为

$$a\%[2(x_{food_2}b_{food} + x_{water_2}b_{water})] + (1 - a\%)[2(x_{food_1}b_{food} + x_{water_1}b_{water})] + \varepsilon_1$$

$$a\%[2(x_{food_2}p_{food} + x_{water_2}p_{water})] + (1 - a\%)[2(x_{food_1}p_{food} + x_{water_1}p_{water})] + \varepsilon_2$$

退出策略的改变类似，按经验概率分配天气后增加一个合理冗余。需要特别注意的是，上面两个合理冗余的角标只是为了区分计量单位的不同，对应的资源是相同的，而退出策略的合理冗余完全不同，是由退出时的情况决定的。

5.2.2 考虑沙暴的情形

由于沙暴是较为罕见的天气，所以其经验概率相较于其他两种情况非常小，以至于可以忽略不计。因此我们在分配天气时，可以将发生沙暴的情况归入合理冗余中。但如果沙暴发生在路途中时，除造成额外的资源损耗（包含在合理冗余内）之外，还导致了一天时间的损失，这意味着一天挖矿净收益的损失，为方便起见，我们将其视为沙暴天气下一天挖矿收益的净损失。我们将这样的损失称为沙暴天气下合理冗余外的损失，记作 S 。我们在最后计算玩家的总收益时，减去这部分损失。

存在村庄时，我们要考虑是否前往村庄，给出判断的策略。由于天气情况未知，我们尽可能选择晴朗的一天出发，以最大程度减少路上的资源消耗。当前往村庄购买资源带来的净收益（挖矿净收益 - 补充资源价格 - 往返路程消耗资源的价格）大于零时，我们前往村庄补充资源，否则我们直接前往终点，结束游戏。需要注意的是，计算往返路程消耗资源的价格时，要根据经验概率进行天气的分配。

退出策略与上一节类似，这里不再赘述。

5.3 具体问题的求解（“第三关”与“第四关”）

5.3.1 第三关的求解

由图可知，矿山不在起点与终点的连线上。分析计算可知共两种最有可能成为最优解的情况，不挖矿，仅一段时间持续挖矿。因玩家只知当天天气状况，且无沙暴天气。由第一关数据分析可得，十天内出现高温天气的概率为 0.625，出现晴朗天气的概率为 0.375。所以，水的基础消耗量的期望值为 7，食物的基础消耗量的期望值为 7，用期望值计算，我们有理由假设最终均无物资剩余。

1. 不挖矿，直接由起点抵达终点，沿最短路径 $1 \rightarrow 5 \rightarrow 6 \rightarrow 13$ ，仅需三日抵达。过程如下：
 - 1) 从起点出发，携带 42 箱水和 42 箱食物，负重 210kg，支出 $210+420=630$ 元。
 - 2) 三日后到达终点，此时物资无剩余，保留资金总计 $10000-630=9370$ 元。
2. 仅一段时间持续挖矿，过程如下：
 - 1) 从起点出发，携带 175 箱水和 175 箱食物，负重 875kg，支出 2625 元。
 - 2) 行走三日后抵达矿山，挖矿五天，收入 1000 元。
 - 3) 选择最优路径行走两日后抵达终点。抵达终点无物资剩余，最终保留资金总计 $10000+1000-2625=8375$ 元。

经比较可得，情况一为最优策略，保留资金总计 9370 元。

5.3.2 第四关的求解

由图可知，矿山在起点与终点的连线上。分析计算可知共三种最有可能成为最优解的情况，不挖矿，仅一段时间持续挖矿，有两段时间持续挖矿。因玩家只知当天天气状况，又已知 30 天内较少出现沙暴天气。我们不妨认为在游戏过程中只有晴朗和高温天气，只在到达终点后游戏再延续三天，即多置备三天物资。当若干情况均多置备三天物资时，不影响比较分析。因此我们完全有理由假设游戏过程只出现晴朗和高温天气。

由第一关数据分析可得，出现高温天气的概率为 0.625，出现晴朗天气的概率为 0.375。所以，水的基础消耗量的期望值为 7，食物的基础消耗量的期望值为 7，用期望值计算，我们有理由假设最终均无物资剩余。

1. 不挖矿，直接由起点抵达终点，沿最优路径 $1 \rightarrow 6 \rightarrow 7 \rightarrow 12 \rightarrow 13 \rightarrow 18 \rightarrow 19 \rightarrow 24 \rightarrow 25$ ，仅需八日抵达。过程如下：
 - 1) 从起点出发，携带 112 箱水和 112 箱食物，负重 560kg，支出 $560+1120=1680$ 元。
 - 2) 八日后到达终点，此时物资无剩余，保留资金总计 $10000-1680=8320$ 元。
2. 仅一段时间持续挖矿，过程如下：
 - 1) 从起点出发，携带 238 箱水和 238 箱食物，负重 1190kg，支出 3570 元。
 - 2) 行走五日后抵达矿山，挖矿六天，收入 6000 元。

3) 选择最优路径行走三日后抵达终点。抵达终点无物资剩余，最终保留资金总计 $10000+6000-3570=12430$ 元。

3. 有两段时间持续挖矿，过程如下：

1) 从起点出发，携带 224 箱水和 264 箱食物，负重 1200kg，支出 3760 元。

2) 继续行走五日抵达矿山，挖矿六天，收入 6000 元。

3) 行走两日后达到村庄，补充 238 箱水和 198 箱食物，此时共携带 238 箱水和 238 箱食物，负重 1190kg，支出 6340 元。

4) 再行走两日回到矿山，挖矿八日，收入 8000 元。

5) 沿最优路径抵达终点，无物资剩余。最终保留资金总计 $10000+6000+8000-3760-6340=13900$ 元。

经比较可得，情况三为最优策略，保留资金总计 13900 元。

六、问题三的模型建立与具体求解

6.1 问题的分析

问题三相较于前两问增加了玩家的人数，这时我们要引入博弈论的知识，讨论多玩家之间的竞争与合作关系。由于多玩家同时同一条路径上移动会带来额外的资源消耗，在矿山同时挖矿又会减少收益，在村庄同时补充资源又会提升价格，所以玩家之间的博弈对于最终的收益是至关重要的。我们需要讨论在收益减少的情况下，前往矿山是否仍具有价值？在移动的过程中，怎样移动是最优的？如果选择前去挖矿，采取什么策略可以使我的收益最大化？如果需要补充资源，何时去村庄是最优的？

本问的两个问题分别在其余条件一定（整个游戏时段天气与全体玩家策略已知）和其余条件不确定（只知道当天的天气和全体玩家策略）的情况下讨论。由于问题的自由度过高，在合理且不失一般性的前提下，我们在讨论时会对问题提出不同的假设加以限制。对于上述问题，我们会给出判断的策略。显然，由题目给出的条件不难注意到，所有的策略都将有相同的准则：要与尽可能少的玩家同时做出同样的行动。

我们的思路流程如图 5。

6.2 模型的建立

6.2.1 第（1）问的情形

首先我们确定 n 个玩家之间的竞争合作关系。假设 n 个玩家在第 0 天达成某种合约，使得全体玩家的总收益最优，那么对于每个玩家有两种选择：遵守或背弃。如果背弃合约带来的额外收益足够大，那么背弃是有利可图的。不失一般性，我们考虑两名玩家的情况。对于单个玩家，假设遵守带来的收益是 1，背弃带来的收益是 1.5，但另一个

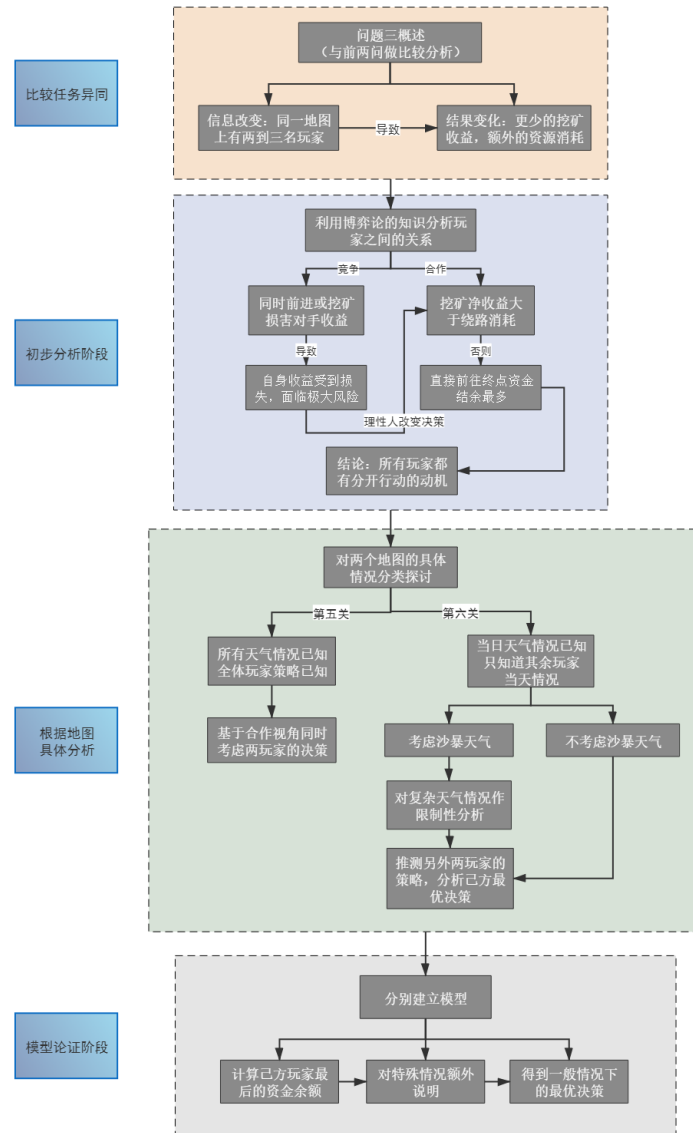


图 5 问题三分析流程图

玩家的收益将会成为 0.5, 如果两人都背弃, 则两人的收益均为 0.5。于是玩家都倾向于背弃合约, 这便是博弈的“囚徒困境”。这时, 尽可能快地完成游戏 (即从起点直接前往终点) 是最佳策略, 使损耗达到最小值。

表 1 博弈的“囚徒困境”

A \ B	遵守	背弃
遵守	(1,1)	(0.5,1.5)
背弃	(1.5,0.5)	(0.5,0.5)

想要摆脱囚徒困境, 必须使规定的合约中不存在背弃的收益, 即设定背弃的惩罚机

制。如果合约中存在为了使整体收益最大化导致部分玩家收益不能达到最优的情况，我们就设定补偿机制，使得玩家之间的收益达到均衡。当然，如果合约中没有背弃的收益甚至背弃会带来更多的损耗，囚徒困境不存在。

接着我们确定玩家在区域间移动的策略。由于同时走同一路径的玩家越多，行走的消耗量越大，所以我们尽可能不走同一条线路。如果图中我们要求的最短路径结果不止一条，那么我们让玩家分别走不同的最短路径。如果多个玩家移动的最短路径有重复，我们需要判断在 t 时刻走还是 $t+1$ 时刻走消耗的资源更少。设 c 为玩家的基础消耗量， k 为除我以外在该时刻移动的玩家，选择 $t+1$ 时刻离开条件为

$$2(k_t + 1)c_t > 2(k_{t+1} + 1)c_{t+1} + c_t$$

即

$$(2k_t + 1)c_t > 2(k_{t+1} + 1)c_{t+1}$$

前往村庄补充资源的策略显然为有且仅有我一人时前往，否则资源价格过高，补充资源带来的收益不足以弥补购买它造成的损耗。

由于每名玩家的行动方案预先给定，因此在基础收益足够大³的前提下，我们是否挖矿的策略为：当 $\frac{r}{k} - 3c > 0$ 时，我们进行挖矿，否则不挖矿，执行退出策略，离开矿山前往终点。

6.2.2 第（2）问的情形

对于天气未知情况的讨论与问题二相同，下面对玩家方案未知的情况做讨论。

我们采取一种新的策略，称其为“异步策略”，即任两个玩家不在同一时刻做同一动作。这个策略要求我们要么在同一时刻与其他玩家做不同的动作，要么在不同时刻与其他玩家做出同一动作。首先，我们求所给图的最短路径，若最短路径不止一条，采取同时刻不同动作的策略；若最短路径重复，采取不同时刻同动作的策略。类似地，我们可以得到对于挖矿动作和去村庄补充资源动作的策略。异步策略以一个相对较小的代价，换来一个局部最优的结果，在第（2）问的情形下，是玩家所能采取的最佳策略。

6.3 具体问题的求解（“第五关”与“第六关”）

6.3.1 第五关的求解

由第三关可知，当只有一个玩家，且无沙暴天气时，最优选择为 $1 \rightarrow 5 \rightarrow 6 \rightarrow 13$ 。设两个玩家分别为甲和乙，甲乙独立做决策。

1. 若甲乙二人选择合作，合作时二者的最优行动路线分别为， $1 \rightarrow 5 \rightarrow 6 \rightarrow 13$ ， $1 \rightarrow 1 \rightarrow 5 \rightarrow 6 \rightarrow 13$ ，到达终点后，甲乙两人的保留资金平均分配，实现均衡。

³ 矿山是必经的顶点

此时甲的行动过程为：

- 1) 从起点出发，携带 30 箱水和 34 箱食物，负重 158kg，支出 490 元。
- 2) 三日后到达终点，此时物资无剩余，保留资金总计 $10000-490=9510$ 元。

此时乙的行动过程为：

- 1) 在起点停留一日，携带 33 箱水和 38 箱食物，负重 175kg，支出 545 元。
- 2) 三日后到达终点，此时物资无剩余，保留资金总计 $10000-545=9455$ 元。

最终，甲乙两人的保留资金实现均衡，均为 9482.5 元。

2. 若甲乙两人为竞争关系，在独立决策时都会选择最优路线，且二者互不相让。

此时甲乙行动过程一致：

- 1) 从起点出发，携带 60 箱水和 68 箱食物，负重 316kg，支出 980 元。
- 2) 三日后到达终点，此时物资无剩余，保留资金总计 $10000-980=9020$ 元。

最终，甲乙两人的保留资金均为 9020 元。

比较两种情况，可知甲乙两人应通过合作实现最优，最终保留资金均为 9482.5 元。

6.3.2 第六关的求解

由第五关分析可知，在利益相互影响的时候，应选择合作实现效益最优。依据第四关数据可得，当单独做决策时，挖矿的保留资金大于不挖矿的保留资金。当三人同时挖矿时，总收益 1000 元，总消耗 945 元；两人同时挖矿，总收益 1000 元，总消耗 630 元；一人挖矿，总收益 1000 元，总消耗 315 元。因此挖矿人数越少，净收益越高。

设三人分别为甲、乙、丙。经分析知，效益最优的情况为，甲直接从起点到达终点；乙与甲同时出发，到达矿山，仅一段时间持续挖矿；丙先在起点停留一段时间，并在乙离开矿山的时候在矿山挖矿。

1. 甲的行动过程为：

- 1) 从起点出发，携带 112 箱水和 112 箱食物，负重 560kg，支出 $560+1120=1680$ 元。
- 2) 沿最优路径 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 25$ ，八日后到达终点，此时物资无剩余，保留资金总计 $10000-1680=8320$ 元。

2. 乙的行动过程为：

- 1) 从起点出发，携带 238 箱水和 238 箱食物，负重 1190kg，支出 3570 元。
- 2) 行走五日后抵达矿山，挖矿六天，收入 6000 元。
- 3) 选择最优路径行走三日后抵达终点。抵达终点无物资剩余，最终保留资金总计 $10000+6000-3570=12430$ 元。

3. 丙的行动过程为：

- 1) 在起点停留七天，第一天携带 49 箱水和 49 箱食物，负重 245kg，支出 735 元。第七天时补充 238 箱水和 238 箱食物，负重 1190kg，支出 3570 元。
- 2) 行走五日后抵达矿山，挖矿六天，收入 6000 元。

3) 选择最优路径行走三日后抵达终点。抵达终点无物资剩余,最终保留资金总计 $10000+6000-3570-735=11695$ 元。

最终, 甲乙丙三人的保留资金实现均衡, 均为 10815 元。

七、总结

通过以上的讨论和探究, 我们得到了一个相对完备的模型, 给出了不同情况下玩家的最优策略。在建立模型的过程中, 我们通过合理假设与合理建参大大简化了问题的复杂性, 并通过讨论有效减少了模型的误差。但对于自由度和复杂性更高的问题, 模型未能给出较好的结果, 仍需要进一步的改进。

参考文献

- [1] Yijing Chen. Application of Improved Dijkstra Algorithm in Coastal Tourism Route Planning[J]. Journal of Coastal Research,2020,106(sp1).
- [2] 左秀峰, 沈万杰. 基于 Floyd 算法的多重最短路问题的改进算法 [J]. 计算机科学,2017,44(05):232-234+267.
- [3] 王海英. 图论算法及其 MATLAB 实现, 北京: 北京航空航天大学出版社, 2010.2

附录 A 源程序

1.1 Floyd 算法

```
1  function[P u] = f_ path(W)
2  n = length(W);
3  U = W;
4  m = 1;
5  while m<= n
6      for i=1:n
7          for j=1;n
8              if U(i, j) > U(i, m) + U(m, j)
9                  U(i, j) = U(i, m) + U(m, j);
10             end
11         end
12     end
13     m = m+1;
14 end
15 u = U(1,n);
16 % 输出最短路的顶点
17 P1 = zeros(1,n) ;
18 k = 1;
19 P1(k) = n;
20 V = ones(1,n) * inf;
21 kk=n;
22 while kk ~= 1
23     for i = 1:n
24         V(1,i) = U(1,kk) - W(i,kk);
25         if V(1,i) == U(1,i)
26             P1(k+1) = i;
27             kk = i;
28             k = k+1;
29         end
30     end
31 end
32 k = 1;
```

```

33 wrow = find(P1 ~= 0);
34 for j = length(wrow) : (-1) : 1
35     P(k) = P1(wrow(j));
36     k = k+1;
37 end
38 P;

```

1.2 Dijkstra 算法

```

1 function [d index1 index2] = Dijkf(a)
2 M= max(max(a));
3 pb(1: length(a)) = 0;
4 pb(1) = 1;
5 index1 = 1;
6 index2 = ones(1,length(a));
7 d(1: length(a)) = M; d(1) = 0; temp = 1;
8
9 while sum(pb) < length(a)
10     tb = find(pb==0);
11     d(tb) = min(d(tb), d(temp) + a(temp,tb));
12     tmpb = find(d(tb) == min(d(tb)));
13     temp = tb(tmpb(1));
14     pb(temp) = 1;
15     index1= [index1, temp];
16     index= index1(find(d(index1) == d(temp)- a(temp, index1)));
17     if length( index)>= 2
18         index = index(1);
19     end
20     index2(temp) = index;
21 end
22 d;
23 index1;
24 index2;

```

附录 B 问题一的 Result.xlsx

第一关					第二关				
日期	所在区域	剩余资金数	剩余水量	剩余食物量	日期	所在区域	剩余资金数	剩余水量	剩余食物量
0	1	5800	180	330	0	1	5300	130	405
1	25	5800	164	318	1	9	5300	114	393
2	24	5800	148	306	2	18	5300	98	381
3	23	5800	138	292	3	26	5300	88	367
4	23	5800	128	282	4	26	5300	78	357
5	22	5800	118	268	5	27	5300	68	343
6	9	5800	102	256	6	28	5300	52	331
7	9	5800	92	246	7	28	5300	42	321
8	15	4170	245	232	8	29	5300	32	307
9	13	4170	229	220	9	38	5300	16	295
10	12	4170	213	208	10	39	3410	189	283
11	12	5170	183	178	11	39	3410	179	273
12	12	6170	159	160	12	30	3410	163	261
13	12	7170	144	139	13	30	4410	148	240
14	12	8170	120	121	14	30	5410	124	222
15	12	9170	96	103	15	30	6410	100	204
16	12	10170	72	85	16	30	7410	76	186
17	12	11170	42	55	17	30	8410	46	156
18	12	11170	32	45	18	30	9410	16	126
19	13	11170	16	33	19	39	5730	196	200
20	15	10430	36	40	20	47	5730	180	188
21	9	10430	26	26	21	55	5730	170	174
22	21	10430	16	12	22	55	6730	155	153
23	27	10430	0	0	23	55	7730	131	135
24					24	55	8730	116	114
25					25	55	9730	86	84
26					26	55	10730	62	66
27					27	55	11730	47	45
28					28	55	12730	32	24
29					29	56	12730	16	12
30					30	64	12730	0	0

图 6 Result.xlsx