Dynamic Programming and Game-Theoretic Modeling for Resource-Constrained Decision Optimization: A Case Study of the Desert-Crossing Game

Xingyu Zhou[1*], Shuhan Li[2*]

[1]Falcuty of Computer Science and Technology Information, University Malaya, Kuala Lumpur, Malaysia

[2]Falcuty of Science, University Malaya, Kuala Lumpur, Malaysia

Abstract

This paper explores strategic decision-making challenges arising from a desert-crossing simulation game, which serves as a representative model for resource-constrained routing under uncertainty and competition. Multiple scenarios are examined, including single-player settings with known or unknown weather conditions, and multi-player environments with either pre-defined or real-time strategic interactions. Dynamic programming (DP), Dijkstra's algorithm, knapsack problem formulations, and game-theoretic models form the foundation of the analysis. Computational techniques, such as LINGO optimization and Monte Carlo simulation, are applied to obtain and compare solutions. Under fully deterministic weather, a cyclic mining and supply-return route emerges as the optimal strategy. In stochastic environments, simplified heuristics based on expected monetary outcomes are shown to be effective. For multi-agent scenarios, integrating optimal and suboptimal routes reduces resource competition. Dynamic games further benefit from day-by-day route adjustments aimed at minimizing resource consumption while maintaining equitable returns. The study concludes with an evaluation of model performance, identifies current limitations such as validation needs and scenario specificity, and outlines directions for future research, including application to logistics planning and multi-agent system coordination and extensions to more complex network structures and real-world logistics or supply chain problems where resource allocation and pathfinding under uncertainty are crucial.[1]

Keywords—Dynamic programming, Dijkstra's shortest path algorithm, Knapsack problem, Game theory, Nash equilibrium, Stochastic simulation

1 Introduction

1.1 Literature Review

In multistage decision problems, the principle of optimality underpins dynamic programming (DP), first formalized by Bellman[3]. For deterministic routing on nonnegative-weighted graphs, Dijkstra's algorithm efficiently computes shortest paths in $O\big(|E| + |V|\log|V|\big)$ time, and may itself be viewed as a DP instance[6].

Modern routing problems frequently extend these ideas to orienteering variants, where one maximizes collected rewards under time or capacity constraints. For example, Tang et al. embed DP subroutines within reinforcement-learning frameworks to improve orienteering solutions[7]. When uncertainty enters—e.g. stochastic travel times—Monte Carlo Tree Search has been used to solve chance-constrained orienteering problems[5], and rolling-horizon DP with sample-average approximation tackles time-dependent stochastic travel[10].

Meanwhile, noncooperative routing games model competition over shared network resources.[9] Altman and Wynter analyze how self-interested agents reach equilibrium in congested flows via fixed-point iterations akin to DP updates[2]. In discrete route-selection settings, Xu et al. integrate DP-based knapsack solvers into a Nash-equilibrium search, capturing strategic detours under resource contention[8]. While individual techniques are well-established, their integrated application here provides a structured framework for analyzing multi-stage, resource-constrained decisions under varying information structures and competition, offering a testbed for strategies potentially applicable to domains like vehicle routing with refueling or distributed resource gathering.

The "Crossing the Desert" game synthesizes these themes: each stage's cost depends on stochastic weather, inventory constraints resemble knapsack structures, and multiple players compete over limited resources. To our knowledge, no prior work has jointly applied DP, Dijkstra subroutines, and noncooperative game theory in this setting. This paper fills that gap by:

1. Developing single-agent DP models under known and unknown weather scenarios;
2. Embedding these into discrete-time game models for multi-agent competition;
3. Validating strategies via LINGO optimization and C++ Monte Carlo simulation.

1.2 Game Overview

With a map, players use their initial funds to buy a certain amount of water and food (including food and other daily necessities), start from the starting point, and walk in the desert. The goal is to reach the end point within the specified time and keep as much money as possible. You will encounter different weather conditions along the way, and you can also replenish funds or resources in mines and villages.

1.3 Basic rules of the game

1. Single player game rules

(1) The basic time unit is day. When the game starts, the player is at the starting point, which is recorded as day 0. The player must reach the end point on or before the deadline. After reaching the end point , the game ends for that player.

(2) The weather in all areas of the desert is the same every day, which is one of the three conditions: ”clear”, ”hot” or ”sandstorm”.

(3) Every day, players can move from one area of the map to another adjacent area, or they can stay where they are. On a sandstorm day, players must stay where they are. (In a map, two areas with a common border are considered adjacent, and two areas with only a common vertex are not considered adjacent.)

(4) Crossing the desert requires two resources: water and food. The minimum unit of measurement is a box. The total weight of water and food a player has each day cannot exceed the upper limit. If the water or food runs out before reaching the end, the game is considered a failure.

(5) The amount of resources consumed by a player staying in one place for one day is the basic consumption, and the amount of resources consumed by a player walking for one day is twice the basic consumption .

(6) When players stay at the mine, they can earn money through mining. The amount of money earned from mining for one day is called basic income. If mining, the amount of resources consumed is three times the basic consumption; if not mining, the amount of resources consumed is the basic consumption. (Mining is not allowed on the day of arrival at the mine, but mining is also possible on sandstorm days.)

(7) On day 0, the player can use the initial funds to purchase water and food at the base price at the starting point. The player can stay at the starting point or return to the starting point, but cannot purchase resources at the starting point multiple times. When the player passes through or stays in the village, he can use the remaining initial funds or the funds obtained from mining to purchase water and food at any time. The price per box is twice the base price. After reaching the end point, the player can return the remaining resources at half the base price.

2. Multiplayer Game Rules

(1) If any of the players walk from area A to area B on a certain day, the amount of resources consumed by any of them is k times the basic consumption.

(2) If any of the players mine in the same mine on a certain day, the amount of resources consumed by any of them is 3 times the basic consumption, and the funds that each player can obtain through mining in a day are the basic income .

(3) If any of the players purchase resources in the same village on a certain day, the price of each box will be 4 times the base price.

(4) In other cases, the amount of resources consumed and the price of resources are the same as those in a single-player game.

1.4 Problem

Question 1: There is only one player, and the weather conditions are known in advance every day during the entire game period. Give the optimal strategy for the player under normal circumstances. Solve the ”First Level” and ”Second Level” in the attachment. The remaining funds (remaining water, remaining food) refers to the funds (water, food) after all the resources required for the day have been consumed. If there is any purchase behavior on the same day, it refers to the funds (water, food) after the purchase is completed.

Question 2: There is only one player, and the player only knows the weather conditions of the day. He can decide the action plan for the day based on this. Give the best strategy for the player under normal circumstances, and discuss the ”third level” and ”fourth level” in the attachment in detail.

Question 3: There are n players who have the same initial capital and start from the starting point at the same time.

(1) Assume that the weather conditions for each day of the game are known in advance. Each player's action plan must be determined on day 0 and cannot be changed thereafter . Give the strategies that players should take in general, and discuss the ”fifth level” in detail.

(2) Assume that all players only know the weather conditions for the day. Starting from the first day , each player knows the action plans of other players and the amount of resources left after completing the action for the day, and then determines their own action plan for the next day. Give the strategies that players should adopt in general, and discuss the ”Sixth Level” in detail.

2 Model Development and Solution Approaches

2.1 Explanation of symbols

2.2 Analysis of Question 1

According to the rules of the game, the game is roughly calculated and analyzed, and the optimal strategy is determined according to different situations. Thus, multiple plans are determined, such as going directly from the starting point to the end point, from the starting point to the mine or village, and then back to the end point. Then one or more plans are selected according to the specific situation of different levels, and the remaining funds of the player after reaching the end point under different plans are calculated, and the optimal plan is obtained by comparison.

2.3 Analysis of Question 2

Since players can only know the weather conditions of the day, in general, they can simply estimate the local weather condi-

Table 1: List of Symbols and Definitions

Symbol	Meaning
R_w	Total expenditure amount
R_{w_1}	Expenditure on goods purchased in the initial village
R_{w_2r}	Expenditure on goods purchased in the village during the r-th loop
R_{w_2}	Total expenditure on goods purchased during transit
W_0	Initial capital
d_i	Unit price of item i
k_i	Quantity of item i purchased
P	Base income
t_j	Number of days required in stage j
t_{jq}	Number of sunny days in stage j
t_{jg}	Number of hot days in stage j
t_{js}	Number of sandstorm days in stage j
X_{ij}	Amount of item i consumed in stage j
b_{iq}	Consumption of item i on a sunny day
b_{ig}	Consumption of item i on a hot day
b_{is}	Consumption of item i on a sandstorm day
c_i	Weight of item i
P_{ij}	Plan option j under situation i
M_i	Maximum number of boxes for item i
$f[i][v]$	DP state variable in the knapsack problem
R'_w	Revenue from residual goods at the destination
R_{mn}	Position of person m after the n selection
W_{mn}	Remaining capital of person m after the n selection

tions based on the number of days, choose the specific direction of movement for the day, and buy more resources based on the shortest path from the starting point to the end point to provide more options for the player's movement. For the third level, two solutions can be obtained through analysis. Under the condition of randomly generating enough weather patterns, the expected value of the funds left over from the two solutions after n games with different weather conditions is calculated to determine the optimal solution. For the fourth level, based on the basic model of problem one, four solutions for problem four are obtained, and then the four solutions are simplified using the characteristics of the graph and known conditions. Using similar ideas to the third level, the expected value of each solution under enough weather conditions is used to measure the advantages and disadvantages of the solution.

2.4 Analysis of Question 3

In the first question, since there are multiple players participating in the game at the same time, the optimal route when there is only one player is no longer the optimal route. Therefore, the general strategy of the player is to randomly choose one of the optimal and suboptimal routes as the action plan for this game on day 0. When solving the "fifth level", first consider the shortest route, three days, four days, and five days. Considering that the two players are in a competitive relationship, both players consider routes with less consumption or more benefits, so only consider both choosing a three-day route, both choosing a four-day route, or one of them choosing a three-day route and one choosing a four-day route, or both choosing a five-day mining route. Introducing random numbers and using C++ for simulation, we obtained the amount of funds left when the player reaches the end in all the cases we listed.

In the second question, because after the end of each day's game, players will determine tomorrow's action plan based on their own funds and resources, and know other players' funds and resources to infer other players' action plans. In this case, players need to choose the movement plan that consumes the least funds and resources for the next day. When conducting a specific analysis and discussion of the "Sixth Level", 0-1 variables are introduced to control the weather. Let the unknown location of the mth person after the nth iteration be , combined with the map of the "Sixth Level", dynamic data prediction for players is carried out, and the restrictions on players' mining, path selection, and whether to enter the village are discussed.

3 Scenario 1: Single Player with Deterministic Weather

3.1 Determination of the Optimal Strategy

The overall decision direction of the player is: staying in areas other than the mine will consume player resources and bring no benefits, so it is not considered that the player will stay in other areas in sunny or hot weather, and the player will not return to the starting point after departure.

If you decide to go mining, the player's general decision direction is: while ensuring that water and food are consumed roughly at the same time, try to buy more food at the starting point, buy more water and less food in the village; after reaching the end, try to avoid having surplus resources.

Case 1: At the starting point, decide whether to travel to the mine to extract resources based on the mine's basic daily income.

Decision 1: If the basic daily income of the mine is far less than the funds required by the player to go to the mine, the optimal strategy is to purchase only the resources that the player needs from the starting point to the end point at the starting point, without going to the mine, and go directly to the end point.

The decision to undertake a mining detour is formulated as a cost-benefit analysis. Let ΔC_{mine} represent the **additional cost** (in monetary units) of traveling to the mine, mining for m days, and then proceeding to the final destination, as compared to taking the most direct path to the destination. This cost is derived from the weather-dependent consumption of resources (water and food) along the longer path and during mining activities.

The strategy of choosing to mine is optimal if the total revenue generated exceeds this additional cost. This is formalized by the following inequality:

$$p \cdot m > \Delta C_{\text{mine}} \tag{1}$$

where:

- p is the base daily income from mining (a constant given in the game rules).
- m is the number of days the player chooses to spend mining ($m \geq 1$).
- ΔC_{mine} is the net additional cost of the mining detour.

Decision 2: The player should opt for the mining strategy if and only if there exists an integer $m \geq 1$ such that inequality (1) holds and the player's initial resources and the time constraint allow for this detour.

Case 2: At the starting point, calculate the funds needed for the player to go to the village and to the mine based on the weather. Based on the comparison of the funds, decide whether to go to the mine or the village first.

Decision 3: If the funds required to reach the mine are small, then go to the mine first. If the remaining resources after reaching the mine are sufficient to support the player to reach the village to replenish resources after digging for at least one day, then the best decision is to reach the mine first.

Decision 4: If the funds required to reach the village are less, then go to the village first.

Case 3: The player makes the best decision based on the actual situation in the mine or village. And the player can go back and forth between the mine and the village.

Situation 3-1: In the mine, on the premise of ensuring that the player can reach the end point from the mine, the player decides whether to go to the village to replenish resources based on the remaining resources, funds and weather conditions of the remaining days.

Decision 5: If the remaining resources, funds, and days are sufficient to support the player to go to the village to replenish resources and the income from returning to the mine to mine is greater than the player's capital consumption to replenish resources, then the optimal decision is to go to the village to replenish resources and return to the mine to mine.

Decision 6: If the player goes to the village to replenish resources at the end of the remaining days, and the income from returning to the mine to mine is less than the cost of replenishing resources, then the optimal strategy is for the player not to replenish resources and return directly to the end point.

Situation 3-2: In the village, decide whether to go back to the mine to mine based on the remaining resources, funds and number of days.

Decision 7: If the player's mining income is greater than the capital expenditure, the optimal strategy is for the player to travel to the mine to extract resources after replenishing resources in the village.

Decision 8: If the player's mining income is less than the capital expenditure, the optimal strategy is for the player not to mine and return to the end point.

3.2 Establishment of general model

3.2.1 Model preparation - establishment of description matrix

Based on the map that the player gets when starting the game, use a 0-1 matrix to describe whether area A and area B are adjacent.

$$\alpha_{ij} \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ share a common boundary} \\ 0 & \text{if } v_i \text{ and } v_j \text{ do not share a common boundary} \end{cases}$$

The matrix A_0 can be obtained as follows:

$$A_0 = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

3.2.2 Model from the starting point to the end point directly

If the player does not consider going to the mine to mine , then he can directly consider the shortest path from the starting point to the end point . The shortest path is certain, and the money to be spent on buying water and food is also certain , so after determining the shortest path , the player's money at the end point can also be determined accordingly. (In the case where the shortest path passes through a mine or a village, we will divide it into the following scheme for discussion).

Using Dijkstra's shortest path graph theory algorithm, we can get the following model:

Model 1 (All-Pairs Shortest Path): Let $A^{(k)}(i, j)$ represent the length of the shortest path from node i to node j where only nodes $1, 2, \ldots, k$ are allowed as intermediate nodes.

$$A^{(k)}(i,j) = \min\left(A^{(k-1)}(i,j),\ A^{(k-1)}(i,k) + A^{(k-1)}(k,j)\right) \tag{2}$$

for $k = 1$ to n, where n is the total number of nodes (areas) on the map.

The initial state is:

$$A^{(0)}(i,j) = \begin{cases} 0 & \text{if } i = j \\ w(i,j) & \text{if } i \neq j \text{ and an edge exists with weight } \\ \infty & \text{otherwise (no direct connection)} \end{cases}$$

3.2.3 Model from the starting point to the end point via a mine or village

1. Preliminary classification of models

Based on the above optimal strategy, we first preliminarily determined three options, namely, going to the mine first and then to the village, going to the village first and then to the mine, and going back and forth between the mine and the village.

A simple analysis shows that the goal of the game is to reach the end within the specified time and keep as much money as possible. Considering that the basic income of the mine is relatively rich, walking or staying in areas other than the mine will only consume resources, so we consider excluding the necessary time to move between the starting point, the mine and the village, and the end point, so that the player can only stay in the mine as much as possible. We establish mathematical models corresponding to these three solutions respectively. When solving the levels, we add, reduce or exchange the intermediate links according to the specific situation, and introduce known information to simplify the model of each level.
The first level is shown in the Fig.1.

2. Model establishment for the scenario where the player goes to the village first and then to the mine

Model 2: The player's walking diagram is as follows: The objective function is to maximize the remaining funds at the endpoint:

$$max\ w = w_0 + p(t_0 - 1) + R'_w - R_w \tag{3}$$

where w_0 is the initial funds, P is the player's basic income at the mine, R'_w is the funds obtained by returning resources at the endpoint, and R_w is the total funds spent on purchasing resources.

$$R_w = R_{w_1} + R_{w_2} \tag{4}$$

Here, $R_{w_1} = \sum_{i=1}^{2} d_i k_i$ represents the funds spent on purchasing k_1 boxes of water and k_2 boxes of food at the starting point, and $R_{w_2} = 2\sum_{i=1}^{2} d_i k_{i+2}$ represents the funds spent on purchasing k_3 boxes of water and k_4 boxes of food in the village. d_i denotes the unit price. The total funds spent must not exceed the initial funds.

Using the knapsack problem in dynamic programming, determine the ratio of water and food the player should purchase

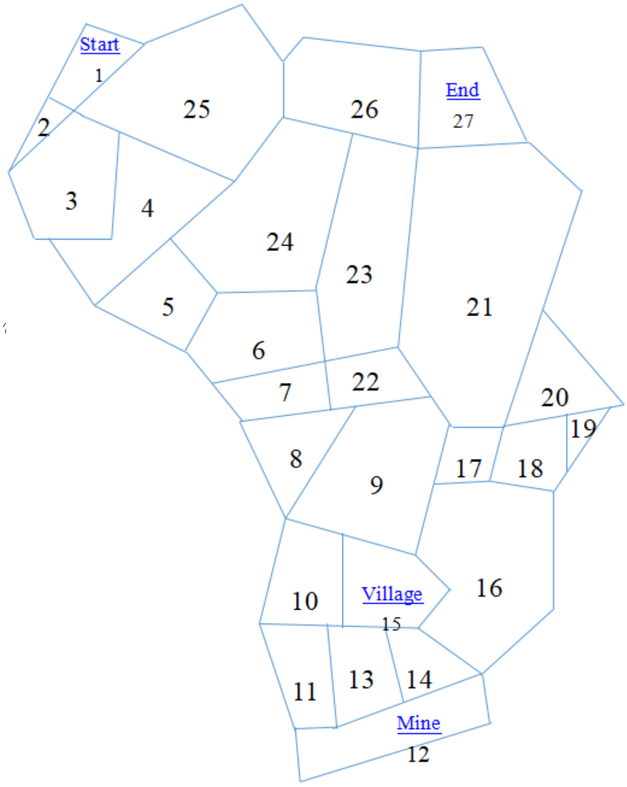

Figure 1: "Level 1" map

at the starting point:

$$f[i][v] = \max\left\{ f[i-1][v - k_i C_i] + k_i w_i \;\middle|\; 0 \le k_i \le M_i \right\} \tag{5}$$

$$\text{where } i \in \{1,2\}, \quad m \in \{1,2,3,4\}$$

The player's game duration is 30 days:

$$t_j = t_{jq} + t_{jg} + t_{js}, \quad j \in \{1,2,3,4\}, \quad t_j \le 30 \tag{6}$$

The resource consumption boxes without mining are:

$$X_{ij} = 2(b_{iq}t_{jq} + b_{ig}t_{jg}) + b_{is}t_{js}, \quad j \in \{1,2,3\} \tag{7}$$

The resource consumption boxes for mining control are:

$$X_{ij} = 3(b_{qj}t_{jj} + b_{gj}t_{fg} + b_{hj}t_{hs}), \quad j = 4 \tag{8}$$

At the endpoint, the player can return remaining resources at half price, with the resulting income:

$$R'_w = \frac{1}{2}\sum_{i=1}^{2} d_i \left(k_i + k_{i+2} - \sum_{j=1}^{4} X_{ij} C_j \right) \tag{9}$$

The total weight of resources owned by the player must not exceed the load capacity limit:

$$\sum_{i=1}^{2} C_i (k_i + k_{i+2}) - \sum_{j=1}^{4} X_{ij} C_j \le M \tag{10}$$

For clarity, the core step in many shortest path algorithms involves comparing paths through intermediate nodes. A standard formulation for computing the shortest path between all pairs of nodes (like the Floyd-Warshall algorithm) is: This formulation iteratively improves the shortest path estimate between nodes *i* and *j* by considering paths through an intermediate node *k*. Additionally, although this scenario does not follow the shortest path, imposing a shortest path constraint restricts unnecessary player movements and reduces resource waste:

Let $A^{(k)}(i,j)$ represent the length of the shortest path from node i to node j where only nodes $1, 2, \dots, k$ are allowed as intermediate nodes.

$$A^{(k)}(i,j) = \min\left(A^{(k-1)}(i,j),\ A^{(k-1)}(i,k) + A^{(k-1)}(k,j)\right) \tag{11}$$

for $k = 1$ to n, where n is the total number of nodes (areas) on the map.

The initial state is:

$$A^{(0)}(i,j) = \begin{cases} 0 & \text{if } i = j \\ w(i,j) & \text{if } i \neq j \text{ and an edge exists with weight } w(i,j) \\ \infty & \text{otherwise (no direct connection)} \end{cases}$$

3. Modeling of the scenario where the player goes to the mine first and then to the village

The player's walking diagram is as follows:

Model 3: Similarly, the following model can be established:

$$\max w = w_0 + p(t_4 - 1) + R'_w - R_w$$

$$R_w = \sum_{i=1}^{2} d_i k_i + 2\sum_{i=1}^{2} d_i k_{i+2} \leq w_0$$

$$f[i][v] = \max\{f[i-1][v - k_i C_i] + k_i w_i, \geq 0\}$$

$$i \in \{1,2\}, \quad m \in \{1,2,3,4\}$$

$$0 \leq k_i \leq M, \quad k_i - \sum_{j=1}^{m} X_{ij}$$

$$R_{w1} \leq w_0$$

$$R_{w2} \leq w_0 - R_{w1} + p(t_4 - 1)$$

$$t_j = t_{jq} + t_{jg} + t_{js}, \quad j \in \{1,2,3,4\}$$

$$X_{ij} = 2\left(b_{qf_{jq}} + b_{gf_{jg}}\right) + b_{gf_{jg}}, \quad j \in \{1,2,3\}$$

$$X_{ij} = 3\left(b_{qf_{jq}} + b_{gf_{jg}}\right) + b_{gf_{jg}}, \quad j = 4$$

$$R'_w = \frac{1}{2}\sum_{i=1}^{2} d_i \left(k_i + k_{i+2} - \sum_{j=1}^{4} X_{ij}\right)$$

$$\sum_{i=1}^{2} C_i(k_i + k_{i+2}) - \sum_{i=1}^{2}\sum_{j=1}^{4} X_{ij} C_i \leq M$$

$$A^{(k)}(i,j) = \min\left(A^{(k-1)}(i,j),\ A^{(k-1)}(i,k) + A^{(k-1)}(k,j)\right)$$

The differences from **Model 3** compared to **Model 2** are:

1. Constraint 1:

$$R_{v1} \leq w_0$$

This means the funds spent by the player at the starting point cannot exceed the initial funds.

2. Constraint 2:

$$R_{v2} \leq w_0 - R_{w1} + p(t_4 - 1)$$

This indicates that the funds spent by the player in the village must not exceed the sum of the remaining funds after starting point purchases and the income earned from mining.

3. Dynamic Programming Adjustment:

$$f[j][v] = \max\{f[i-1][v - k_i C_t] + k_i w_t, \quad 0 \leq k_i \leq M_t,\}$$

This formula specifies that the resources purchased at the village must support the player's journey from the starting point to the mine, mining activities, and resource replenishment from the mine to the village.

4. The establishment of a model for players to travel back and forth between the mine and the village

The player's walking diagram is as follows, in which players can travel back and forth between the mine and the village based on their existing funds, resources, remaining days and weather, without losing funds.

On the basis of **Model 2** and **Model 3**, we introduce the parameter r to represent the number of times the player goes back and forth between the mine and the village, and establish the mathematical model of this scheme:

Model 4:

$$\max w = w_0 + p\sum_{i=1}^{2}(t_{4r} - 1) + R_w - R_w$$

$$R_w = R_{w_1} + R_{w_2}$$

$$R_{v_1} \leq w_0$$

$$R_{W_3} = 2\sum_{i=1}^{2} d_i k_{(i+2)} r$$

$$R_{v_2} r \leq w_0 - R_{v_1} + p(t_{4r} - 1) - 2\sum_{i=1}^{2} d_i k_{(i+2)(r-1)}$$

$$f[i][v] = \max\{f[i-1][v - k_i C_i] + k_i w_i, \quad 0 \leq k_i \leq M_i\}$$

$$i \in \{1,2\}, \quad m \in \{1,2,3,4\}$$

$$t_j = t_{jq} + t_{jg} + t_{js}, \quad j \in \{1,2,3,4\}$$

$$t = t_1 + t_2 + r(t_3 + t_4) \leq 30$$

$$X_{ij} = 2(b_{iq} t_{jq} + b_{ig} t_{jg}) + b_{is} t_{js}, \quad j \in \{1,2,3\}$$

$$X_{ij} = 3(b_{iq} t_{jq} + b_{ig} t_{jg} + b_{is} t_{js}), \quad j = 4$$

$$k_{3n} - \sum_{i=2}^{3} X_{3i} \geq 0, \quad k_{4n} - \sum_{i=2}^{3} X_{4i} \geq 0 \quad (r = n)$$

$$R'_w = \frac{1}{2}\sum_{i=1}^{2}\sum_{r=1}^{n} d_i \left(k_{ir} + k_{(i+2)r} - \sum_{j=1}^{4} X_{ij}\right)$$

$$\sum_{i=1}^{2} C_i(k_i - X_{i1}) + \sum_{i=1}^{2} X_{i1}k_{(i+2)r} - r\sum_{i=1}^{2}\sum_{\substack{j=2\\ j\neq 3}}^{4} X_{ij}C_i \leq M$$

$$A^{(k)}(i,j) = \min\left(A^{(k-1)}(i,j),\ A^{(k-1)}(i,k) + A^{(k-1)}(k,j)\right)$$

Where:

1. $R_{T_2} = 2\sum_{i=1}^{2} d_i k_{(i+2)} r$ represents the money spent on purchasing resources in the village r times;

2. $R_{v2}r \leq w_0 - R_{w1} + p(t_{4r} - 1) - 2\sum_{i=1}^{2} d_i k_{(i+2)(r-1)}$ (when $r = 0, R_{v2}r = R_{w1}$) where means that the funds spent on purchasing resources in the village each time cannot exceed the sum of the remaining funds after purchasing resources at the starting point and the income from the mine this time minus the funds spent on the last purchase in the village.

3. $t = t_1 + t_2 + r(t_3 + t_4) \leq 30$ which means the total number of game days cannot exceed 30 days;

4. $k_{3n} - \sum_{i=2}^{3} X_{3i} \geq 0, \quad k_{4n} - \sum_{i=2}^{3} X_{4i} \geq 0 \quad (r = n)$ which means that the player had enough water and food when he returned to the end point from the village or mine for the last time.

3.3 Model building and solution of the "First level"

3.3.1 Description Matrix for details)

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.3.2 Model solution and result display

1. Solution steps

Use LINGO software to solve the model:

Step 1: Use Dijstra's shortest path algorithm to find the five shortest paths from the starting point to the village or mine, from the end point to the village or mine, and from the starting point to the end point.

Step 2: Substitute the shortest path into the four basic models to calculate the optimal solution among the four solutions. When solving, the description matrix needs to be flipped and solved twice.

Step 3: Sort out the optimal route according to the output of Model 4 of the optimal solution

2. Results display

The following table can be obtained after sorting.

The player returned to the starting point on the 23rd day. Draw the route on the map of "Level 1" as shown below Fig. 2. As shown below Table. 2.

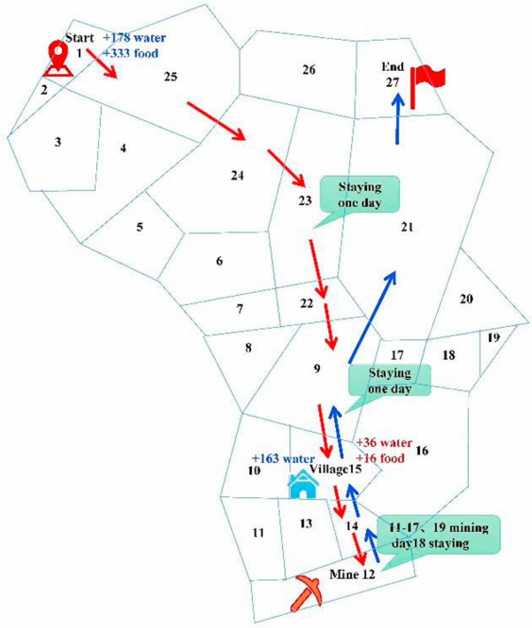

Figure 2: The best route map for the "Level 1"

3.3.3 Model building and solution of the "Second level"

1. Description matrix:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

2. Model solution steps and model display

The Second level is shown in the Fig. 3.

The solution steps are similar to those of the first level, except that the description matrix and the total number of map areas are replaced. We also use LINGO to solve the model according to the route Table. 3.

Draw the route on the map of "Level 2" as shown below Fig. 4.

The line in the picture is the walking path, and the area is the area where the player needs to stay.

4 Scenario 2: Single Player with Stochastic Weather

4.1 Model establishment and solution of problem 2: Optimal Strategy

Since the player can only know the weather conditions on the current day, the weather on the day the player plays the game and before that is known.

1. If there are few game days, players cannot make a simple prediction of the weather in that place. Then players can only

Table 2: ”First Level” Results Table

Day	Location	Remaining Funds	Remaining Water
0	1	5800	180
1	25	5800	164
2	24	5800	148
⋮	⋮	⋮	⋮
21	9	10430	26
22	21	10430	16
23	27	10430	0
24			
25			
26			
27			
28			
29			
30			

Table 3: ”Second Level” Results Table

Day	Location	Remaining Funds	Remaining Water
0	1	6475	247
1	2	6475	231
2	3	6475	215
⋮	⋮	⋮	⋮
28	55	14775	32
29	63	12365	16
30	64	12365	0

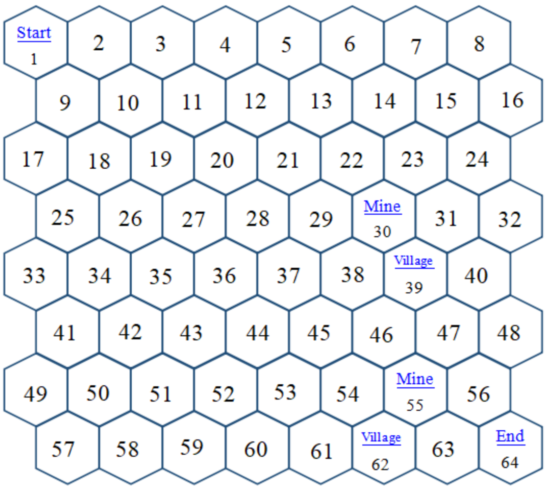

Figure 3: ”Level 2” map

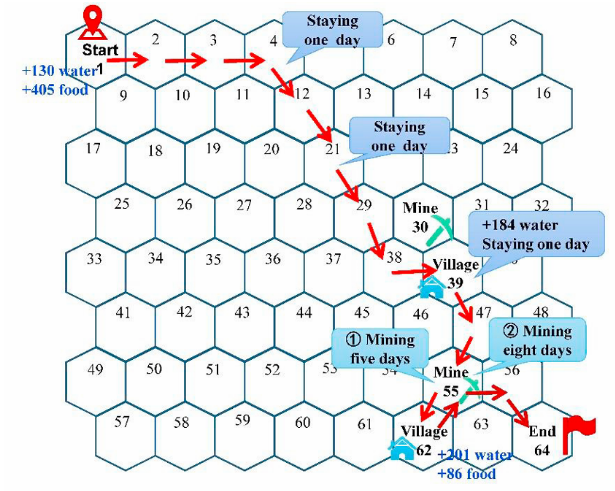

Figure 4: The best route map for the ”Level 2”

make decisions based on the conditions of the day. If it is sunny, choose a moving direction that is close to the end point and the village or mine. If it is hot weather, players choose to move in the shortest direction from their current location to the end point.

2. If there are many game days, move according to the strategy in the first few days of the game. When the player has played the game for a certain number of days and can make a rough estimate of the weather in that place, he can decide whether to go to the mine or village based on his own resources and funds, or directly return to the end point by the shortest path from his current location.

3. If the player has reached the village or mine, he can make the best decision based on situation 3 in question 1.

4. Therefore, at the starting point, players need to buy an appropriate amount of resources that are more than what is needed for the shortest path based on the length of the map.

4.2 Establishment and solution of the “third level” model

1. Graphic analysis and solution determination

In question one, we gave four basic solutions: going directly from the starting point to the end point, from the starting point to the mine or village and then to the end point. By analyzing the image characteristics of the third level below, we found that there is no village on the Fig. 5.

Therefore, we simplify the four basic solutions into two solutions suitable for this level according to the specific situation:

Solution 1: Take the shortest path from the starting point to the end point;

Option 2: Take the shortest path from the starting point to the mine for mining and then take the shortest path back to the end point.

2. Establishment of the corresponding solution model

It is known that there will be no sandstorm weather within 10 days, so we define t_1 as days of high-temperature weather and t_2 as days of clear weather:

$$t_1 + t_2 = 10 \tag{12}$$

Since the graph is relatively simple, we directly obtain the shortest path from the starting point to the endpoint (3 days) and from the mine back to the endpoint (2 days). Assume the

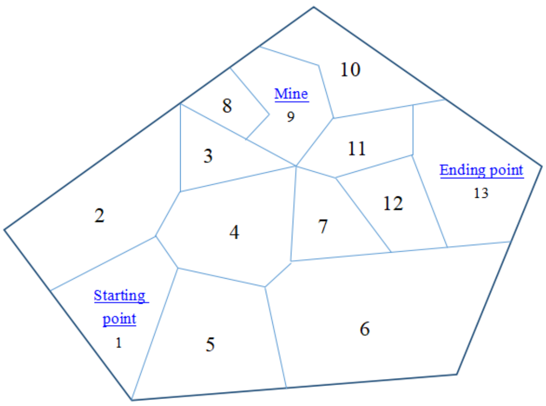

Figure 5: "Level 3" map

first 3 days consist of t_3 days of high-temperature weather and t_4 days of sunny weather:

$$t_3 + t_4 = 3 \tag{13}$$

Model for Scheme 1:

$$w_1 = w_0 - 2\sum_{i=1}^{2}(d_i b_{ig} t_3 + d_i b_{ig} t_4) \tag{14}$$

$$t_3 + t_4 = 3$$

Assume mining days include t_5 days of high temperatures and t_6 days of clear skies, ensuring the player can return smoothly from the mine to the endpoint:

$$t_5 + t_6 \leq 5 \tag{15}$$

Model for Scheme 2:

$$w_2 = w_0 - 2\sum_{i=1}^{2}[d_i b_{ig}(t_1 - t_5) - d_i b_{ig}(t_2 - t_6)] \tag{16}$$

$$t_1 + t_2 = 10$$
$$t_5 + t_6 \leq 5$$

3. Comparison of the two solutions determined

Since the player only knows the current day's weather condition, the specific weather pattern throughout the game cannot be predetermined. Thus, direct comparison of w_1 and w_2 is infeasible. We randomly generate n weather patterns, including all high-temperature days, all clear days, first 5 days as high-temperature followed by 5 clear days, etc.:

$$t_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}, \quad (i \in \{1, 2, 3, 4, 5, 6\})$$

where t_i and t_{i+1} satisfy Equations 12 and 13.

For each weather pattern, the player follows both **Scheme 1** and **Scheme 2** in the game. The expected remaining funds for each scheme after n simulations are calculated as:

$$\mathbb{E}[w_1] = \frac{\sum_{i=1}^{n} w_{1i}}{n}, \quad \mathbb{E}[w_2] = \frac{\sum_{i=1}^{n} w_{2i}}{n}$$

When n is sufficiently large, the relative magnitudes of these expectations reflect the superiority of the two schemes.

4. Model solution and results

Based on the model of the first question, a random factor is introduced to describe the weather changes, and the description matrix and the total number of map areas are also changed. The map description matrix of the "third level" is:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Scheme 1: Mean remaining funds $\mu_1 \approx 9925$, Standard Deviation $\sigma_1 \approx 25$;
Scheme 2: Mean remaining funds $\mu_2 \approx 9139.25$, Standard Deviation $\sigma_2 \approx 20$;
Since $\mu_1 > \mu_2$, and considering the variances, Scheme 1 appears statistically better under the simulated conditions.
After 50 random operations, we get:

$$\overline{W}_1 \approx 9925, \quad \overline{W}_2 \approx 9139.25, \quad \overline{W}_1 > \overline{W}_2,$$

so plan 1 is better.

4.3 Establishment and analysis of the "fourth level" model

1. Determination of the plan

Since the map of the fourth level has both villages and mines, it conforms to the four basic solutions we determined in question one. However, since we only know the weather on that day in this question, in order to introduce the parameter t, we adjust the four basic solutions and obtain four solutions that are more in line with this question.

Plan 1: Take the shortest path from the starting point to the end point;

Plan 2: Start from the starting point, go to the mine and then go to the end point;

Plan 3: Start from the starting point to the mine and then go to the end point;

Plan 4: From the starting point to the mine or village, travel back and forth between the mine and the village according to actual conditions, and finally go from the village or mine to the end point.

2. Establishment of the corresponding solution model

Under the premise of the four basic schemes in Problem 1, introduce the parameter t.

a. Assume there are t_1 days of high-temperature weather, t_2 days of clear/cloudy weather, and t_3 days of sandstorm weather.

Model for Scheme 1:

$$w_1 = w_0 - \sum_{i=1}^{2} d_i k_i$$

$$\begin{cases}\sum_{m=1}^{2} t_m = 8\\ k_i - 2\left(b_{iq}t_1 + b_{ig}t_2\right) - b_{is}t_3 \geq 0, \quad i \in \{1,2\}, \quad 0 \leq k_i \leq M_i\end{cases}$$

b. Assume mining includes t_4 days of high-temperature weather, t_5 days of clear/cloudy weather, and t_6 days of sandstorm weather.

Model for Scheme 2:

$$w_2 = w_0 - \sum_{i=1}^{2} d_i k_i + p(t_4 + t_5 + t_6 - 1)$$

$$\begin{cases}\sum_{m=1}^{3} t_m \leq 30\\ k_i = 2\left[b_{iq}(t_1 - t_4) - b_{ig}(t_2 - t_5)\right] - b_{is}(t_3 - t_6) \geq 0, \quad i \in \{1,2\}\end{cases}$$

c. Assume the player purchases water k_3 and food k_4 in the village.

Model for Scheme 3:

$$w_3 = w_0 - \sum_{i=1}^{2} d_i k_i - 2\sum_{i=1}^{2} d_i k_{i+2} + \frac{1}{2}\sum_{i=1}^{2}\left(k_i + k_{i+2} - \sum_{j=1}^{3} X_{ij}\right)$$

$$\begin{cases}\sum_{m=1}^{3} t_m \leq 30\\ k_1 + k_3 - 2\left(b_{1q}t_1 + b_{1g}t_2\right) - b_{1s}t_3 \geq 0\\ k_2 + k_4 - 2\left(b_{2q}t_2 + b_{2g}t_2\right) - b_{2s}t_3 \geq 0, \quad 0 \leq k_i \leq M\\ X_{ij} = 2\left(b_{iq}t_{jq} + b_{ig}t_{gj}\right) + b_{is}t_{js}, \quad j \in \{1,2,3\}\\ \sum_{i=1}^{2} C_i\left[k_i + k_{i+2} - 2\left(b_{iq}t_1 + b_{ig}t_2\right) - b_{is}t_3\right] \leq M\end{cases}$$

d. Assume mining includes t_4 days of clear weather, t_5 days of high-temperature weather, t_6 days of sandstorm weather, with $1 \leq r \leq n$.

Model for Scheme 4:

$$w_4 = w_0 + p\sum_{r=1}^{n}\left(t_{4r} + t_{5r} + t_{6r} - 1\right) - \sum_{i=1}^{n} d_i k_i$$

$$\begin{cases}\sum_{m=1}^{3} t_m \leq 30\\ k_1 + k_{3r} - 2\left(b_{1q}t_{1(r-1)} + b_{1g}t_{2(r-1)}\right) - b_{1s}t_{s(r-1)} \geq 0\\ k_2 + k_{4r} - 2\left(b_{2q}t_{2(r-1)} + b_{2g}t_{2(r-1)}\right) - b_{2s}t_{s(r-1)} \geq 0\\ X_{ij} = 2\left(b_{iq}t_{jq} + b_{ig}t_{js}\right) + b_{is}t_{js}, \quad j \in \{1,2,3\}\\ X_{ij} = 3\left(b_{iq}t_{jq} + b_{ig}t_{js} + b_{is}t_{js}\right), \quad j \in \{4\}\end{cases}$$

3. Comparison of four solutions

So we randomly generate n weather patterns, including all high-temperature days, all clear days, etc.:

$$t_i \in \{0,1,2,3,4,5,6,7,8,9,10\}, \quad (i \in \{1,2,3,4,5,6\})$$

Since Level 4 rarely experiences sandstorms within 30 days, we impose:

$$t_1 = t_2 = 10t_3$$

For each weather pattern, the player follows all four schemes in the game. The expected remaining funds for each scheme after n simulations ($\mathbb{E}[w_1], \mathbb{E}[w_2], \mathbb{E}[w_3], \mathbb{E}[w_4]$) are calculated. When n is sufficiently large, the relative magnitudes of these expectations reflect the superiority of the four schemes.

4. Specific analysis of the combination of model and map

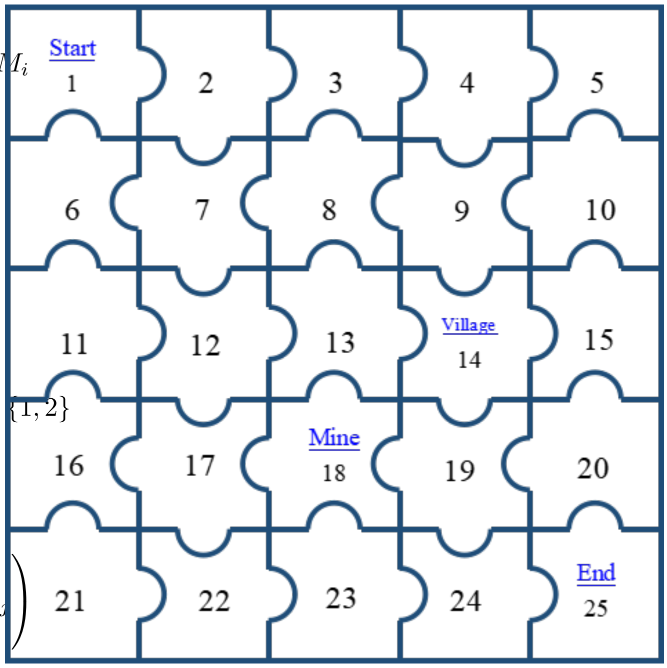

Figure 6: "Level 4" Map

(1) Since the village and the mine are both on the shortest path from the player's starting point to the midpoint, and the price of materials in the village is twice that of the starting point, we do not consider option 3, that is, we do not consider going directly back to the end point after shopping in the village from the starting point. At the same time, the income from the mine is relatively high, so we do not consider option 1, which is to take the shortest path directly to the end point. Draw the route on the map of "Level 4" as shown below **Fig. 6**

(2) Analysis shows that the shortest paths from the starting point to the mine and the village are the same. Due to the uncertainty of the weather, the player cannot decide the quantity of resources to purchase if he goes to the village first, so the player considers going to the mine first to mine.

(3) In the mine, the player is guaranteed to reach the end point by calculating the remaining resources, funds and weather conditions of the remaining days.

a. If the remaining resources, funds, and days are sufficient to support the player to go to the village to replenish resources and the income from returning to the mine to mine is greater than the player's capital consumption to replenish resources, then the optimal decision is to go to the village to replenish resources and return to the mine to mine.

b. If the player goes to the village to replenish resources at the end of the remaining days, and the income from returning to the mine to mine is less than the cost of replenishing resources, the optimal strategy is for the player not to replenish resources and return directly to the end point.

5. Model solution and results

The description matrix of the “fourth level” is:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

From the above analysis, we know that plan 4 is better. We use LINGO to calculate the model, and the results are consistent with our analysis. The specific plan is: travel to the mine to extract resources first, then go to the village to replenish resources, then travel to the mine to extract resources, and finally return to the end. When the player returns to the end, the remaining funds are 13625 yuan.

5 Scenario 3: Multi-player Competitive Setting

5.1 If the weather is known in advance and the plan is not changed after the starting point is determined

5.1.1 Strategies that players should generally adopt

Since multiple players participate in the game simultaneously, the following rules apply:

1. If k $(2 \leq k \leq n)$ players move from area A to B on the same day, the resource consumption becomes $2k$ times of the base consumption.

2. If k $(2 \leq k \leq n)$ players mine in the same mine on the same day, each player’s resource consumption is 3 times of the base consumption, and their daily mining income is reduced to $\frac{1}{k}$ times of the base income.

3. If k $(2 \leq k \leq n)$ players purchase resources in the same village on the same day, the price per box increases to 4 times of the base price.

This will cause the optimal route when there is only one player to no longer be the optimal route, so players should consider the suboptimal route in the range of options when choosing a plan on day 0.

Therefore, the general strategy of the player is to randomly choose one of the optimal route and the suboptimal route as the action plan for this game on day 0. This resembles a mixed strategy in game theory, where players randomize over pure strategies (routes) to maximize their expected payoff given the potential actions of others, aiming for a Nash equilibrium where no player can benefit by unilaterally changing their strategy.

Without considering other reasons for longer routes, the probability of players choosing a longer route is small, and choosing a longer route does not effectively reduce the probability of meeting other players, so the resources consumed by players with a high probability of choosing a longer route will increase.Draw the route on the map of ”Level 5” as shown below Fig. 7.

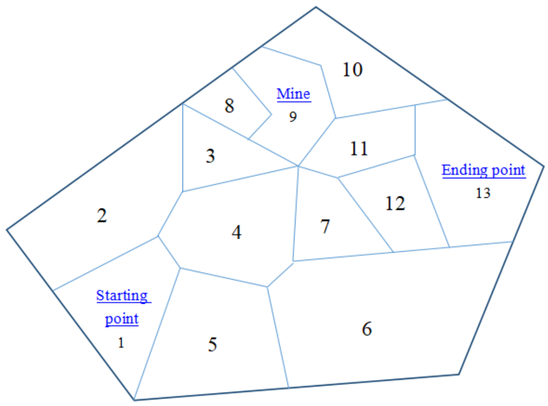

Figure 7: ”Level 5” map

5.1.2 Establishment and solution of the “Fifth level” model

1. Determination of route

Considering only the shortest routes, the **3-day routes** are:

$$P_1 = \begin{pmatrix} 1, 5, 6, 13 \\ 1, 4, 6, 13 \end{pmatrix}$$

The **4-day routes** include:

$$P_2 = \begin{pmatrix} 1, 4, 7, 12, 13 \\ 1, 4, 6, 12, 13 \\ 1, 4, 7, 11, 13 \\ 1, 5, 6, 12, 13 \end{pmatrix}$$

Since this question involves mines, we also consider **5-day routes**, categorized as:

With mining opportunities (P_3):

$$P_3 = \begin{pmatrix} 1, 2, 3, 9, 10, 13 \\ 1, 2, 3, 9, 11, 13 \\ 1, 4, 3, 9, 10, 13 \\ 1, 4, 3, 9, 11, 13 \end{pmatrix}$$

Without mining opportunities (P_4):

$$P_4 = \begin{pmatrix} 1, 4, 7, 11, 12, 13 \\ 1, 5, 6, 7, 12, 13 \\ 1, 5, 6, 12, 11, 13 \end{pmatrix}$$

Each row in P represents a distinct route.

2. Determination of the plan

Since the two players are in a competitive relationship, both players consider routes with less consumption or more benefits, so they only consider both choosing the three-day route, both choosing the four-day route, or one of them choosing the three-day route and the other choosing the four-day route, or both choosing the five-day mining route.

(1) Both players choose the 3-day route:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(b_{iq} + b_{ig}) - 4\sum_{i=1}^{2} d_i b_{iq}$$

(2) Both players choose the 4-day route:
a. Choosing P_1 and P_2:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(b_{iq} + b_{ig}) - 4\sum_{i=1}^{2} d_{iq}b_{ig}$$

b. Choosing P_1 and P_3:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_{iq}b_{ig} - 4\sum_{i=1}^{2} d_i(b_{iq} + b_{ig})$$

c. Choosing P_1 and P_4:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(2b_{iq} + b_{ig}) - 4\sum_{i=1}^{2} d_ib_{iq}$$

d. Choosing P_2 and P_4:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(2b_{iq} + b_{ig}) - 4\sum_{i=1}^{2} d_ib_{ig}$$

e. Choosing P_3 and P_4:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(2b_{iq} + b_{ig}) - 4\sum_{i=1}^{2} d_ib_{ig}$$

f. Choosing P_2 and P_4:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(3b_{iq} + b_{ig})$$

(3) One chooses a 3-day route, the other chooses a 4-day route:
a. P_1 and P_2:

$$W = W_1 + W_2 = 2W_0 - 4\sum_{i=1}^{2} d_i(b_{iq} + b_{ig}) - 2\sum_{i=1}^{2} d_ib_{ig}$$

b. P_1 and P_2, P_3:

$$W = W_1 + W_2 = 2W_0 - 4\sum_{i=1}^{2} d_ib_{iq} - 2\sum_{i=1}^{2} d_i(b_{iq} + b_{ig})$$

c. P_1 and P_4:

$$W = W_1 + W_2 = 2W_0 - 4\sum_{i=1}^{2} d_i(b_{iq} + b_{ig}) - 2\sum_{i=1}^{2} 3d_ib_{iq}$$

d. Other situations:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(b_{iq} + b_{ig})$$

(4) Both go mining, with the optimal strategy being P_2, P_3:

$$W = W_1 + W_2 = 2W_0 - 2\sum_{i=1}^{2} d_i(b_{iq} + b_{ig}) - 4\sum_{i=1}^{2} d_ib_{iq}$$

For "Scenario 3," the best option for both players is the above strategy, which yields the maximum value W.

3. Model solution

Use C++ to solve 12 situations and perform simulation tests. The results of the 12 cases are:

19080	18540	19020	19290	19180	19290
19400	18910	19180	18910	19290	19400

Use the RAND function and the system time as the random seed to generate random integers from 1 to 12. Use C++ to perform 100 simulations which preliminary analysis showed was sufficient for the mean outcome to stabilize and take the average as the simulation result, which is 10147.

5.2 Only know the weather of the day and the player's condition after the day ends

5.2.1 Strategies that players should generally adopt

Because after the end of each day's game, players will determine tomorrow's action plan based on their own funds and resources, and at the same time know other players' funds and resources to infer other players' action plans. In this case, players need to choose the movement plan that consumes the least funds and resources for the next day.

5.2.2 Specific Analysis and Discussion of the "Sixth Level"

Since the map of the sixth level is the same as that of the fourth level, and only the weather conditions of the day are known, players can determine the action plan for the next day based on the general strategy adopted by players in the face of unknown weather in question 2. Assume that the three players in the sixth level are in a competitive relationship with each other, and there is no alliance between players.

1. Introduce 0-1 variables to control the weather

$$x_1 + x_2 + x_3 = 1, \quad x_1, x_2, x_3 \in \{0, 1\}$$

2. Dynamic data prediction for players

Assume that the unknown location of the m person after the nth iteration is R_{mn}, and combine the map of the "sixth level" to predict the dynamic data of the player. Draw the route on the map of "Level 6" as shown below Fig. 8.

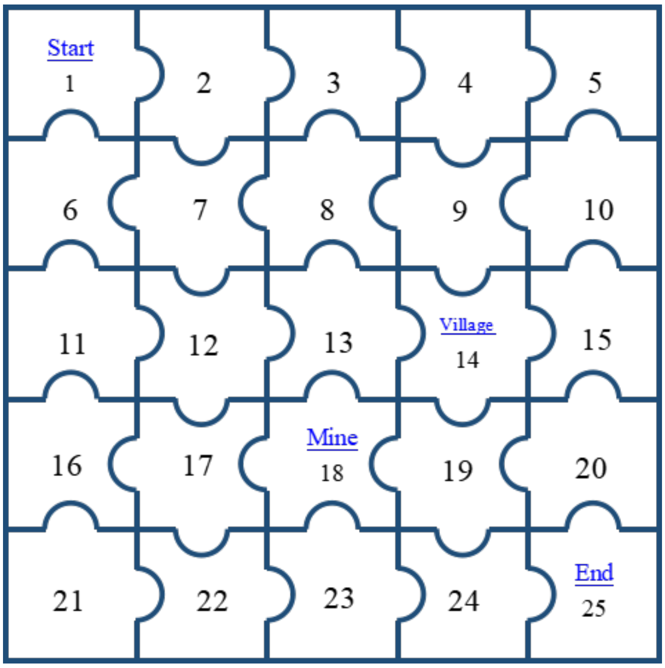

Figure 8: "Level 6" map

(1) Mining Constraints

a. After the n iteration, the player is around the mine:

$$|18-R_{on}| = 5 \text{ or } 1, \quad |18-R_{on}| = 5 \text{ or } 1, \quad a,b \in \{1,2,3\}$$

b. The player has enough funds to go to the mine and mine for at least one day:

$$W_{on} \geq 6\sum_{i=1}^{2}(x_i b_{ip} d_i + x_i b_{ig} d_i + x_i b_{iq} d_i)$$

or

$$W_{on} \geq 6\sum_{i=1}^{2}(x_i b_{ip} d_i + x_i b_{ig} d_i + x_i b_{iq} d_i), \quad a,b \in \{1,2,3\}$$

Players can go mining $R_{n(n+1)} = 18$ or $R_{n(n+1)} = 18$ if conditions a.b. are met.

(2) Path Restriction

When two players are in the same area and are ready to move to the next area:

$$|R_{on} - R_{on}| = 0$$

$$W_{on} \geq 2\sum_{i=1}^{2}(x_i b_i p_i d_i + x_i b_i p_i d_i + x_i b_i p_i d_i)$$

$$W_{on} \geq 2\sum_{i=1}^{2}(x_i b_i p_i d_i + x_i b_i p_i d_i + x_i b_i p_i d_i)$$

The two players do not move to the same area:

$$R_{n(n+1)} \neq R_{b(n+1)}$$

(3) Village Restrictions

a. After the n iteration, the player is around the village:

$$|14-R_{on}| = 5 \text{ or } 1, \quad |14-R_{on}| = 5 \text{ or } 1, \quad a,b \in \{1,2,3\}$$

b. The player has enough funds to go to the village, and must go to the village to replenish resources before moving:

$$2\sum_{i=1}^{2}(x_i b_{ip} d_i + x_i b_{ig} d_i + x_i b_{iq} d_i) \leq W_{an}$$

$$W_{an} \leq 4\sum_{i=1}^{2}(x_i b_{ip} d_i + x_i b_{ig} d_i + x_i b_{iq} d_i)$$

$$\text{or} \quad 2\sum_{i=1}^{2}(x_i b_{ip} d_i + x_i b_{ig} d_i + x_i b_{iq} d_i) \leq W_{bn}$$

$$W_{bn} \leq 4\sum_{i=1}^{2}(x_i b_{ip} d_i + x_i b_{ig} d_i + x_i b_{iq} d_i), \quad a,b \in \{1,2,3\}$$

(4) Competition between players who reach a special point at the same time

a. Arrive at the mine at the same time. Assume that people with sufficient funds will enter the mine to mine at this time, that is , players with less funds are unwilling to halve the mining benefits (players will lose money).

b. When they arrive at the village at the same time, those who do not have enough resources will definitely enter the village, while those who have enough resources will not enter the village.

c. Players all consider their own interests in the game. If the level differences among players are not big, the final value will be smaller after the game ends.

6 Conclusions

1. Advantages

(1) Using dynamic programming to solve the knapsack problem. Dynamic programming has memory, and the subproblems needed in the new problem can be directly extracted, avoiding repeated calculations and thus saving time.

(2) The knowledge of game theory is used to simplify the abstract concepts of the fifth and sixth levels to facilitate the solution and analysis of random problems.

(3) Converting graph theory problems into dynamic optimization problems provides a structured and computable framework for solving complex routing decisions. This integration greatly enhances the practicality of the model by transforming an intuitive but vague map-based decision into a series of well-defined optimization steps with clear objectives and constraints. The process is illustrated in Figure 9, which shows how the spatial problem (the map and weather) is abstracted into a graph, then into a state-space model, and finally solved via DP to yield an optimal policy. This methodological pipeline is a core contribution of our work.

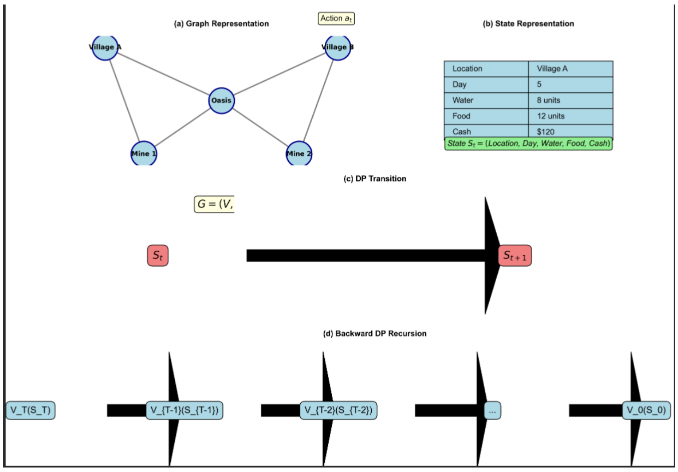

Figure 9: Modeling Framework: From Graph Theory to Dynamic Optimization

2. Disadvantages:

(1) There is a lack of empirical validation of model results against alternative strategies (e.g., greedy algorithms) or real-world data, which results in uncertain error margins compared with actual processes or simpler benchmarks. Future work should include such validation.

(2) In the cycle analysis of mines and villages, the accuracy of the solution is reduced due to the unknown number of cycles. Heuristics were used to approximate the optimal number.

(3) The models are developed specifically for the desert game's ruleset (specific resources, weather, map structures). Generalizing the framework to other domains (e.g., logistics with fuel constraints, multi-robot exploration with charging stations) would require adapting the cost functions, constraints, and state definitions, but the core methodology of combining DP for planning, graph search for routing, and game theory for interaction remains promising. Future research should explore this generalization.

(4) Scalability to larger maps or more players needs assessment, potentially requiring more efficient algorithms or approximations.

References

[1] Contemporary undergraduate mathematical contest in modeling (cumcm). Accessed 26 April 2021.

[2] Altman, E., and Wynter, L. Equilibrium, games, and pricing in transportation and telecommunication networks. *Networks and Spatial Economics* **4** (2004), 7–21.

[3] Bellman, R. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[4] Cao, Z. Exploration of exhaustive, search, and dynamic programming algorithms for the 0-1 knapsack problem. *Computer Knowledge and Technology* **12** (2009), 3193–3194.

[5] Carpin, S. Solving stochastic orienteering problems with chance constraints using monte carlo tree search. *arXiv:2409.03170*, 2024.

[6] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik* **1**(1) (1959), 269–271.

[7] Tang, J., Liu, Y., and Zhu, X. Reinforcement learning approaches for the orienteering problem. *Transportation Science* (2022). Forthcoming.

[8] Xu, J., Liu, K., and Liu, J. Competitive knapsack routing: A nash-dp approach. *Computers & Operations Research* **135** (2021), 113–125.

[9] Zhang, F., Liu, J., and Li, Q. An optimized shortest-path algorithm based on dijkstra's method. *Remote Sensing Information* **000**(002) (2004), 38–41.

[10] Zhang, Y., Wang, L., and Li, D. Data-driven time-dependent orienteering with stochastic travel times. *Computers & Operations Research* **145** (2023), 105–123.

7 List of Appendix

Appendix 1: List of all supporting documents
Appendix 2: "Level 1" description matrix
Appendix 3: LINGO program for solving the "first level"
Appendix 4: Results table of the "first level"
Appendix 5: "Level 2" description matrix
Appendix 6: Results table of the "Second Level"
Appendix 7: "Level 3" description matrix
Appendix 8: Descriptive matrix for the "Fourth Level"
Appendix 9: LINGO program for solving the fourth level
Appendix 10: LINGO program for solving the fifth level

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1
10	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
11	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
13	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0
14	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
15	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1
16	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0
17	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10: "Level 1" description matrix

Sets:

$$a = \{1, 2\}$$
$$a_1 = \{1, 2, 3\}$$
$$b = \{1, 2, 3, 4\}$$
$$\text{link}(a, b)$$

Data:

$d = [5, 10]$, $\quad bq = [8, 6]$, $\quad bg = [5, 7]$, $\quad bs = [10, 10]$,
$w = [5, 10]$, $\quad p = 1000$

Initialization:

$$fv(1) = 0$$

Revenue and Resource Use Definitions:

$$\begin{aligned}
rw &= rw_1 + rw_2 \\
rw_1 &= \sum_{i \in a} d_i \cdot k_i \\
rw_2 &= 2 \sum_{i \in a} d_i \cdot k_{i+2} \\
rw_1 &< w_0 \\
rw_2 &< w_0 - rw_1 + p \cdot (t_4 - 1)
\end{aligned}$$

Capacity Constraint:

$$\sum_{i \in a} c_i \cdot (k_i + k_{i+2}) - \sum_{i \in a} (x_{i1} + x_{i4}) \leq M_1$$

Total Time at Each Stage:

$$t_j = tg_j + tq_j + ts_j, \quad \forall j \in b$$

Resource Consumption:

$$x_{ij} = \begin{cases} 3 \cdot (bg_i \cdot tg_j + bq_i \cdot tq_j + bs_i \cdot ts_j), & \text{if } j = 4 \\ 2 \cdot (bg_i \cdot tg_j + bq_i \cdot tq_j) + bs_i \cdot ts_j, & \text{otherwise} \end{cases}$$

Residual Value:

$$rwp = \frac{1}{2} \sum_{i \in a} d_i \cdot \left(k_i + k_{i+2} - \sum_{j \in b} x_{ij} \right)$$

Forward Value Computation:

$$fv_{i+1} = \max \left(fv1_i + k_i \cdot w_i \right), \quad \forall i \in a$$

Variable Constraints:

$$\begin{aligned}
k_i &\leq m_i, \quad \forall i \in a \\
k_i &- \sum_{j \in \{1,2,3\} \cup \{4\}} x_{ij} > 0, \quad \forall i \in a
\end{aligned}$$

Objective Function (to Maximize):

$$\max Z = w_0 + p \cdot (t_4 - 1) + rwp - rw$$

0	1	5800	180	330
1	25	5800	164	318
2	24	5800	148	306
3	23	5800	138	292
4	23	5800	128	282
5	22	5800	118	268
6	9	5800	102	256
7	9	5800	92	246
8	15	4170	245	232
9	14	4170	229	220
10	12	4170	213	208
11	12	5170	183	178
12	12	6170	159	160
13	12	7170	144	139
14	12	8170	120	121
15	12	9170	96	103
16	12	10170	72	85
17	12	11170	42	55
18	12	11170	32	45
19	14	11170	16	33
20	15	10430	36	40
21	9	10430	26	26
22	21	10430	16	12
23	27	10430	0	0
24				
25				
26				
27				
28				
29				
30				

Figure 11: Results table of the “first level”

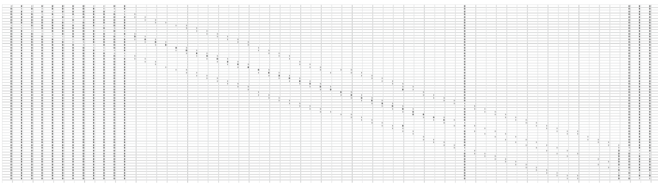

Figure 12: ”Level 2” description matrix

Sets:

$$\begin{aligned}
&a = \{1, 2\} \quad \text{: parameters } d, bq, bg, bs, m \\
&b = \{1, 2, 3\} \\
&b_1 = \{1, 2, 3, 4\} \quad \text{: with } k(i) \\
&c = \{1, 2, \ldots, 9\} \quad \text{: with time variables } t(n) \\
&\text{link}(a, b_1) : x(i, j)
\end{aligned}$$

Data:

$$\begin{aligned}
w_0 &= 10000 \\
d &= [5, 10] \\
bq &= [3, 4] \\
bg &= [9, 9] \\
bs &= [10, 10] \\
p &= 1000
\end{aligned}$$

0	1	6475	247	229
1	2	6475	231	217
2	3	6475	215	205
3	4	6475	205	191
4	4	6475	195	181
5	5	6475	185	167
6	13	6475	169	155
7	13	6475	159	145
8	22	6475	149	131
9	30	6475	133	119
10	30	7475	109	101
11	30	8475	79	71
12	30	9475	55	53
13	30	10475	40	32
14	39	11475	16	14
15	46	6775	241	237
16	55	4365	225	225
17	55	4365	215	215
18	55	4365	205	205
19	55	4365	189	193
20	55	7775	165	175
21	55	8775	150	154
22	55	9775	135	133
23	55	10775	111	115
24	55	11775	96	94
25	55	9365	86	84
26	55	12775	62	66
27	55	13775	47	45
28	55	14775	32	24
29	63	12365	16	12
30	64	12365	0	0

Figure 13: Results table of the “Second Level”

0	1	0	1	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	1	1	0	0	0	0
1	1	1	0	1	1	1	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	0	0	0	1	1
0	0	0	1	0	1	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	1	0	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	0	1
0	0	0	0	0	0	1	0	1	1	0	1	1
0	0	0	0	0	1	1	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0	0	1	1	0

Figure 14: ”Level 3” description matrix

(1) Total supply constraint:

$$\sum_{i \in a} m(i) < 1200$$

(2) Time assignment for last 3 periods:

$$\sum_{\substack{n \in c \\ n \geq 7}} t(n) = 8$$

(3) First-stage remaining wealth:

$$w_1 = w_0 - \sum_{i \in a} d(i) \cdot k(i)$$

(4) Total working time constraint:

$$\sum_{n \in b} t(n) \leq 30$$

(5) Maximum mining quantity:

$$k(i) \leq m(i), \quad \forall i \in a$$

0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	0	1	0	0	0	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	0	0	0	0	0	1	0	0	0
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	0	0	0	0	0	1	0
0	0	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	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	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	0	0	0

Figure 15: Descriptive matrix for the ”Fourth Level”

(6) Resource availability constraint (stage-wise):

$$\begin{aligned} k(i) - 2\left[bq(i)(t(1) - t(4)) + bg(i)(t(2) - t(5))\right] - bs(i)(t(3) - t(6)) \\ - 3\left[bq(i)t(4) + bg(i)t(5) + bs(i)t(6)\right] \geq 0, \quad \forall i \in a \end{aligned}$$

(7) Wealth after mining and profit:

$$w_1 = w_0 - \sum_{i \in a} d(i) \cdot k(i) + p \cdot (t(4) + t(5) + t(6) - 1)$$

(8) First player's individual constraint:

$$k(1) + k(3) - 2 \cdot (bg(1) + bq(1) - bs(1) \cdot t(3)) \geq 0$$

(9) Second player's individual constraint:

$$k(2) + k(4) - 2 \cdot (bg(2) + bq(2) - bs(2) \cdot t(3)) \geq 0$$

(10) Resource consumption calculation:

$$x(i,j) = 2 \cdot (bq(i)t(j) + bg(i)t(j) + bs(i)t(j)), \quad \forall (i,j) \in \text{link}(a, b_1)$$

$$\max Z = w_0 - \sum_{i \in a} d(i) \cdot k(i) - \sum_{i \in a} d(i) \cdot k(i+2)$$

$$+ \frac{1}{2} \sum_{i \in a} \left[k(i) + k(i+2) - \sum_{j \in b_1} x(i,j) \right]$$

Listing 1: C++ Program for Mining Strategy Evaluation

```
#include<iostream>
#include<iomanip>
#include<cstdlib>
#include<ctime>
#define random(a,b) (rand()%(b-a)+a)

using namespace std;

int sigmal(int *a, int *b, int *c)
{
    int result = 0;
    for (int i = 0; i < 2; i++)
    {
        result = result + (b[i] + c[i]) * a[i];
    }
```

```
    return result;
}

int sigma2(int *a, int *b)
{
    int result = 0;
    for (int i = 0; i < 2; i++)
    {
        result = result + a[i] * b[i];
    }
    return result;
}

int main()
{
    int f[100], sum = 0, av;
    ((int)time(0)); // Generate random seeds. Replace 0 with NULL and you will see 2 rows.

    for (int i = 0; i < 100; i++)
    {
        f[i] = (rand() % (11 - 0 + 1)) + 0;
    }

    int bg[2] = {9, 9}, bq[2] = {3, 4}, bq2[2] = {6, 8}, bq3[2] = {9, 12}, bq4[2] = {12, 16}
    int w0 = 10000, w[20], d[2] = {5, 10};

    w[0]  = 2 * w0 - sigma1(d, bg, bq) * 2 - 4 * sigma2(bg, d);
    w[1]  = 2 * w0 - sigma1(d, bg, bq) * 2 - 8 * sigma2(bg, d);
    w[2]  = 2 * w0 - sigma1(d, bg, bq) * 4 - 4 * sigma2(bq, d);
    w[3]  = 2 * w0 - sigma1(d, bg, bq2) * 2 - 4 * sigma2(bq, d);
    w[4]  = 2 * w0 - sigma1(d, bg, bq) * 2 - 4 * sigma2(bq2, d);
    w[5]  = 2 * w0 - sigma1(d, bg, bq2) * 2 - 4 * sigma2(bq, d);
    w[6]  = 2 * w0 - sigma1(d, bg, bq3) * 2;
    w[7]  = 2 * w0 - sigma1(d, bg, bq) * 4 - 2 * sigma2(bq3, d);
    w[8]  = 2 * w0 - sigma1(d, bg, bq3) * 2 - 4 * sigma2(bq, d);
    w[9]  = 2 * w0 - sigma1(d, bg, bq) * 4 - 2 * sigma2(bq3, d);
    w[10] = 2 * w0 - sigma1(d, bg, bq4) * 2;
    w[11] = 2 * w0 - 2 * sigma1(d, bg, bq) - 4 * sigma2(d, bq);

    for (int i = 0; i <= 11; i++)
        cout << w[i] << '\t';

    for (int k = 0; k < 100; k++)
    {
        sum = sum + w[f[k]];
        av = sum / (k + 1);
    }

    cout << av;
    return 0;
}
```