Project Proposals: Swim Back ... Now? (or) Pull Your Goalie Now or Later?

Laura Roettges — Ryan Moreno

July 2024

1 Option 1: Swim Back ... Now?

1.1 Problem Overview

Deep Sea Adventure is a game where you have to balance risk and reward. In the game, players are deep sea divers seeking treasure, the deeper they dive, the richer (higher in value) the treasures they will find. Whoever brings home the greatest value of riches wins, so what's the catch? You, as the players, are all rivals but you're poor, so you're forced to share a rented submarine and a **single tank of air**. If you don't get back to the sub before that tank of air runs out, you'll have to drop all the treasure you've collected in order to make it back on board alive.

The game lasts over a series of 3 rounds i.e. deep sea dives. Each dive begins with a full tank of air and all divers on the submarine. In order to gain points, you must bring Undersea Treasure Chips back to the Submarine. You can only return to the Submarine once per dive, and the dive/round ends when all Divers have returned to the sub or the air runs out. To better understand how the air runs out, the details about treasure value, treasure chip layout, etc review these rules as well as this video from [0:48 to 5:58] for a good demonstration of the game play. So, as a player, you have 2 main questions every round: **when should you pick up treasure and when should you turn around?**

1.2 Data

The data is largely dictated by the rule book. Some key components of the data are:

- 2 Special Dice: 1-3 dots, 2x on each Die
- 12 Blank Undersea Chips (become zero worth spots)
- 3 rounds

- 32 Undersea Treasure Chips: 8 each of 4 types
  - Level 1: Triangular, 0-3 points (2 each)
  - Level 2: Square, 4-7 points (2 each)
  - Level 3: Pentagonal, 8-11 points (2 each)
  - Level 4: Hexagonal, 12-15 points (2 each)

Data needed for modeling:

- T - which turn, $t \exists \{1, 2, ..., t_T\}$ are we on
- r - which round are we on where $r \exists \{1, 2, 3\}$
- p - how many players are actively playing. For simplicity we will start by assuming just 1.
- N - nodes for each chip on the board and a node for the submarine where for n_i i $\exists\{1, 2, ..., 33\}$ with 33 being the submarine (may need to consider adding a dummy finish node)
- C - chip/treasure type on each position i $c_i \exists \{triangular,\ square,\ pentagon,\ hexagon,\ blank,\ none\}$ where none is for the submarine home node this varies from round to round and even turn to turn pending what players pick up
- v_c - average value of treasure type $v_{triangle} = 1.5$, $v_{square} = 5.5$, $v_{pentagon} = 9.5$, $v_{hexagon} = 13.5$, $v_{blank} = 0$, $v_{none} = 0$
- X - arcs with distances between nodes (all connected if and only if only 1 player is on the board, else players skip over the tile that another player is occupying)
- m_i - demand with the submarine being the sink and all other nodes being sources
- s_i - supply of nodes 1 for every treasure chip, 0 otherwise.
- o_t - how much oxygen is left at the end of each turn
- o_0 - initial oxygen level at teh start of the round
- $prob_{ctr}$ - the probability of landing on chip type c, per turn, per round.
- $prob_{roll,t,r}$ - the average expected roll per turn per round which varies based on q (aka how many treasure chips each player has).

1.3 Model Details

We will model this as an **integer program** with some factors similar to a multiperiod planning issue and some factors of a transshipment issue (though it is not as simple, there will definitely need to be multiple if-then conditional statements). A problem with the model we will need to address is the amount of turns that we can take decreases as a factor of how many treasure chips each player has collected so this is not a linear relationship.

There are a multitude of factors that could increase the complexity of the model and solution. At least at first **we plan to start building this model as if you could play alone** to sort through an 'ideal world' scenario to maximize their expected treasure return. We could organize our model by 'turn,' e.g. on which of your turns should you pick up a tile and which turn should you turn around. This would make it easier to model using probabilities, such as, how far on average would you travel on a given turn, on average what treasure chip type would you land on for that turn, and on average what value would a chip of a certain type be (since the chip values are not known to the players at the time of decision making). Modeling in this way may however be challenging to include the factor of the game where you can abandon a treasure chip in order to conserve air, since that relies on landing on a blank marker where treasure had already been taken, so we may need to leave that factor of the game out if we take this approach. This model will also have a temporal factor not only from the turns, but also the rounds as the board chip location/composition changes based on the previous rounds' results.

Variables:

- x_{iptr} - on which node, i, per turn, t, per round, r, does a player, p, pick up a treasure chip
- ret_{iptr} - on which node, i, turn, t, and round, r, does player, p, decide to continue going forward or turn around,
- q_{cptr} - how many treasure chips of each type c, does each player, p, have in their possession at the start of each turn, t, in round r

Constraints:

- oxygen level must be ≥ 0
- can't go further than the last chip on the board (bottom of the sea)
- can't continue rolling after you return to the submarine
- demand $\leq$ supply
- supply *leq* 32

- total rounds = 3
- T total number of turns in a given round $\leq\ o_0 -\ \sum_t \sum_c q_{cptr}$ *aka. Turn total can not exceed oxygen level - total chips all players have at the start of each turn summed over all their turns*
- (if we ever model with multiple players) you can not land on the same chip another player is currently on
- You can not pick up anything with supply ≤ 0

Objective: $max \sum_{r=1}^{3}\ v_c * q_{cpTr}$ where T is the last turn of each round

1.4 Additional Considerations:

Handling the dice rolls:

- We can start by assuming that a roll of 1 die is on average a of value 2.
- Alternately we could compose multiple sets of possible rolls in sequence for each character and just use those sequences for the model. Running the model with multiple sequences should on average yield similar results, we could then look at the averages and variance across the multiple runs to try and draw conclusions.

Model implications and considerations when adding in players:

- If we add players, they may need to be characterized as greedy vs conservative and given a probability of picking up a treasure chip that varies based on how many chips they have.
- Including other players impacts the arc connections between nodes because if you roll a 2 (and have no chips) and another player is 2 spaces in front of you, you actually move in front of them to the space 3 ahead of you.
- Including other players impacts the probability of you landing on a treasure chip worth any value (if they already took the chip and you land on that space it is worthless).
- We would need to sort through how to determine if they dropped their treasure to the bottom of the sea, as it impacts the board setup for the next round, and it impacts the value of the last node on the board.

2 Option 2: When to Pull the Hockey Goalie

2.1 Problem Overview

In hockey, it is common practice to "pull the goalie" in the last few minutes of a game if a team is losing. Pulling the goalie entails subbing out the goalie for an additional offensive player. This way, the losing team has 6 skaters on the ice (and no goalie) instead of the usual 5 skaters and 1 goalie. The key idea behind this is that winning/losing the game is solely decided by which team ends with more points, not *how many* points they win/lose by. So, if team A is down by 2 points with 1 minute left in the game, there is a very small chance of them catching up by the end of the game if the play remains 5v5. Although pulling the goalie makes it more likely that team A will be scored on (called an "empty net" goal), there is minimal cost associated with getting scored on at this point in the game, because team A was almost certainly going to lose anyways. Pulling the goalie also makes it more likely that team A will score a goal, since they have 6 offensive skaters. Even though there is higher risk of getting scored on, the reward associated with potentially scoring outweighs the risk since scoring could bring team A to a tie or win. In essence, after a goalie is pulled, there is higher variance in the number of goals scored (on both sides). Since team A is about to lose the game, this higher variance is to their benefit.

Our question is, given the score of a hockey game and the number of minutes left, when (if ever) is the optimal time for team A to pull their goalie?

2.2 Model Details

Simplifications/assumptions

- For each minute of the game, either 0 or 1 goals are scored by a given team, according to a given probability
- Each team has the same skill level (i.e., the probabilities of scoring are the same)
- The opposing team will always be at full strength and will never pull their goalie
- Once the goalie has been pulled, we cannot switch back to full strength.

Data

- $p_{\hat{g}vg}$: probability of a team scoring during a given minute if they have pulled their goalie (from NHL data)
- p_{gvg}: probability of a team scoring during a given minute if both teams have full strength (from NHL data)
- $p_{gv\hat{g}}$: probability of a team scoring during a given minute if they have full strength and their opponents have pulled their goalie (from NHL data)
- t: number of minutes left in the game
- s_a: current score for team A
- s_b: current score for team B

Decision variables

- $g_i, \forall i \in \{1, 2, ...t\}$: $g_i = 0$ iff the goalie has been pulled and $g_i = 1$ otherwise
- $p_{iA}, \forall i \in \{1, 2, ...t\}$: probability that team A scores a goal in the ith minute remaining.
- $p_{iB}, \forall i \in \{1, 2, ...t\}$: probability that team B scores a goal in the ith minute remaining.
- Note: as we saw in multi-period planning with the decision variable for inventory between months, it may be easiest to include p_{iA} and p_{iB} as decision variables despite the fact that they can be directly derived from g_i. We can enforce their relationship to g_i using constraints.

Expressions

- f_a: number of goals scored by team A at the end of the game (probabilistic)
- f_b: number of goals scored by team B at the end of the game (probabilistic)
- $\mathbb{P}[f_a - f_b \geq 0]$: probability that team A beats team B
- Note: We will need to create mathematical definitions for these based on our decision variables.

Objective

- Max $\mathbb{P}[f_a - f_b \geq 0]$ (We are including ties as a "win" for team A since it still achieved the goal of catching up with team B's score.

Constraints

- $0 \leq g_i \leq 1$ and $g_i \in \mathbb{Z} \forall i \in \{1, 2, ...t\}$ (binary variable)
- $g_{i+1} \leq g_i \forall i \in \{1, 2, ...t-1\}$ (once the goalie has been pulled this cannot be reverted)
- $p_{iA} = \begin{cases} p_{gvg}, & g_i == 1 \\ p_{\hat{g}vg}, & g_i == 0 \end{cases}$ (probabilities of scoring based on NHL data)
- $p_{iB} = \begin{cases} p_{gvg}, & g_i == 1 \\ p_{gv\hat{g}}, & g_i == 0 \end{cases}$ (probabilities of scoring based on NHL data)

Model type

Our choice of model may change as we refine our definitions of f_a, f_b, and Max $\mathbb{P}[f_a - f_b \geq 0]$. To start with, we are considering integer programming because we have the constraint $g_i \in \mathbb{Z}$ (and since p_{iA} and p_{iB} are non-integer, total unimodularity may not apply).

2.3 Additional Considerations:

- **Probabilities**. We need to decide how to handle probabilities in our model. At first glance it may seem reasonable to work with expected values (e.g., expected number of goals scored by team A throughout the remainder of the game). However, this is not sufficient because the point of pulling a goalie is that it changes the variance in goals scored in a way that increases the probability of a win, not that it increases the expected net change in score. (In fact, it likely decreases the expected net change in score since the team pulling the goalie is much more likely to get scored on.) We will need to define $\mathbb{P}[f_a - f_b \geq 0]$ such that it incorporates the probability of each possible final score.

- **Integer requirements**. Perhaps we can relax the constraints $g_i \in \mathbb{Z}$ and allow the assignment of "partial goalies" (analogous to partial flow in network flow). In this case we would have constraints such as $p_{ai} = (g_i)p_{gvg} + (1 - g_i)p_{\hat{g}vg}$.

- **Computing the likelihood of scoring**. When computing the probability of a team scoring in a given minute from NHL stats, we will need to take into account that the data is inherently biased by the game abruptly ending. If we simply considered the number of minutes between goals, we would be overestimating the probability of a team scoring in a given minute because we would fail to include any long stretches without goals at the end of the game. (This is particularly relevant for $p_{\hat{g}vg}$ and $p_{gv\hat{g}}$ because common practice is that goalies are only pulled in the last few minutes of a game, so there are often no goals scored while the goalie is pulled.)