Surface Reclamation using POMDPs

Gabriel Agostine GABRIEL.AGOSTINE@COLORADO.EDU
Department of Aerospace Engineering Sciences
University of Colorado Boulder

Abstract

Humanity is on the brink of extinction and has been forced to survive underground after the relentless assault of an unknown force that has descended on humanity. This document describes a Partially Observable Markov Decision Process (POMDP) formulation for a resource management reinforcement learning agent whose goal is to reclaim the surface from said unknown force. Operating with a handful of squads, the agent must learn to manage limited resources, deploy teams to explore different regions, make strategic decisions under uncertainty, and maintain the sanity and comfort of its people.

Introduction

In this project, we develop a reinforcement learning agent that must optimize resource management and strategic decisions in a partially observable environment. The agent controls an underground facility and must allocate resources, deploy exploration teams, conduct research, and adapt to threats to ensure long-term survival.

This decision-making problem has been formulated as a Partially Observable Markov Decision Process (POMDP), where the agent must make decisions based on incomplete information about the state of the external environment, particularly unexplored territories and threat distributions.

The goal of the agent is to maintain control over all subregions on the surface while also maintaining resources above defined thresholds that are depleted both by the environment and by taking actions. It must also learn methods to increase resource gains by researching new technology and unlocking new subregions that must also be explored by researching them as well.

Background & Related Work

This project stems from the idea of learning how to properly resource manage in an unforgiving environment. The agent is responsible for both reclaiming a number of subregions with an allocated number of squads. It must also maintain a balance of its available resources in order to upkeep the sanity and comfort of the civilians in which it is protecting.

This experiment has been designed as a harsh environment, where the agent will hopefully learn to aggressively micromanage said resources in order to get higher and higher rewards. We hope that the agent will find a good policy on how to complete its given tasks, perhaps in unique and unexpected ways. This may help us learn how to properly resource manage as well as adapt to constant loss in our environment.

POMDP Formulation

A POMDP is formally defined as a tuple (S, A, O, R, T, Z, γ) where:

- S: state space
- A: action space
- O: observation space
- $R(s, a, s')$: reward function
- $T(s' \mid s, a)$: transition probability function

- $Z(o \mid s', a)$: observation probability function
- $\gamma \in [0, 1]$ is the discount factor

State Space S:

The state space S is a $2N + D + 4$ - dimensional ($S \in \mathbb{R}^{2N+D+4}$) discrete space represented as a tuple:

$$S = (R, T, Q, H)$$

where N is the number of squads allocated to surface reclamation, D is the number of subregions to reclaim and:

- $R = [r_1, r_2, r_3, r_4] \in \{0, 1, 2, \ldots, 100\}^4$ represents resource levels for food, medicine, fuel, and materials respectively, each as an integer percentage (0 - 100%) of maximum capacity
- $T = [t_1, \ldots, t_D] \in \{0, 1\}^D$ represents control status of D subregions where $t_i = 0$ indicates undiscovered (locked) and $t_i = 1$ indicates discovered (unlocked)
- $Q = [q_1, \ldots, q_N] \in \{0, 1\}^N$ represents the availability of each squad where $q_i = 0$ indicates unavailable and $q_i = 1$ indicates available
- $H = [h_1, \ldots, h_N] \in \{0, 1, 2\}^N$ represents squad health where $h_i = 0$ indicates damaged, $h_i = 1$ indicates functional, and $h_i = 2$ indicates optimal condition

Action Space A:

The action space A consists of the following discrete actions for deploying any squad to any region, recalling any squad from their current region, allocation of resources for support or healing and researching of new subregions or technology.

These actions will allow the agent to effectively explore the space through the complex transitions from state to state depending on the action. Our hopes is that the agent will find the best action for a multitude of commonly undesirable states it may end up in.

Observation Space O:

The observation space consists of "blurry" readings of the state space. These blurry readings affect the squad health status and subregion control status. All other state values are perfectly observable.

Reward Function $R\,(s, a, s')$:

The reward function $R(s, a, s')$ quantifies the desirability of transitioning from state s to state s' by taking action a. We wish for the agent to learn how to manage resources and maintain the comfort of its people in this simulated world, thus we should reward the agent for having resources above a certain threshold. We also wish for the agent to maintain control over all subregions, so it will be rewarded for maintaining control over more subregions.

The agent will have a reward for keeping each squad in a functional health state. We decided not to have the agent prioritize a optimal health state to account for real life environmental conditions not modeled as well as out of curiosity on how the agent would behave with less care for squads under its control. Lastly, the agent has a small reward for researching new technology, as this is a must for the advancement of its civilization.

Transition Function $T\,(s' \mid s, a)$:

The transition function $T\,(s' \mid s, a)$ defines the probability of transitioning to state s' given that action a is taken in state s. We define this function in part for each type of action below (a more detailed numerical explanation is provided for each action in the Appendix as well):

Deploy Action Transitions:

When executing a deploy action we check to see if the squad is available, we have enough resources and the squad is a nominal health status. If that is the case, then the squad is deployed to the specified region. Once deployed, we may calculate the probability that the complete a successful, partially successful or failed operation based on base success rates per subregion and the health status of the squad.

Resources are then collected for the former two (a partial success yielding slightly less resources and a failed operation yielding no resources). Resources are collected based on the current tech level τ and the base resource yields per subregion. Similarly, the latter success probabilities two have a 40 and 70% chance of decreasing the squad health by 1.

Recall Action Transitions:

When executing a recall action to bring a specified squad back, we check if the squad is actually deployed and we have enough resources to bring them back. If so, then the squad is made available again and resources are depleted. There is also a 1% chance the squad health depletes by 1.

Healing Action Transitions:

When executing a allocate healing action, we first check if the squad is available and we have enough resources. If so, then we deplete the necessary resources and then increase the squad level by 1 with probability 1 or 0.85 depending on if the squads health level previously was 0 or 1 respectively.

Field Support Action Transitions:

When executing a allocate field support action, we first ensure that the squad is deployed and enough resources are present. If so, we deplete the required resources and increase the squads health with probability 0.6 and overall tech level with probability 0.15.

Exploration Research Transitions:

When executing a research exploration technology action, we check if we have enough resources and if so deplete them and increase the overall tech level with probability 0.9.

Subregion Research Transitions:

When executing a research new subregion action, we ensure we have enough resources and the specified subregion is not yet discovered. If so, then we deplete the required resources and unlock the requested subregion with a probability based on the current tech level.

Environmental Transitions:

Regardless of which action is taken, after each step we deplete all resource values by a set amount and (for each squad) if a squad is available there is a 20% chance their health increases by 1. There is also a 70% chance that a random subregion will have its control lost, meaning the agent will have to redeploy to that region.

Observation Function $Z(o \mid a, s')$:

The observation function $Z(o \mid a, s')$ defines the probability of observing o after transitioning to state s' by taking action a. Since resources and squad availability are assumed to be perfectly observable, the observation function primarily affects squad health and territory control. This results in noisy observations of the regions discovered and the health of each squad.

Health observations have a 70% chance of being accurate while subregion discovery observations are only 80% reliable, reflecting the challenges of maintaining accurate intelligence on distant regions.

Discount Factor γ:

A discount factor of $\gamma = 0.99$ will help the agent properly manage its primary tasks while allowing for goals to be worked towards over longer periods of time.

Terminal State:

The agent will reach a terminal state when all subregions are controlled regardless of the value of any other state. This can only be reached by deploying squads to all regions that are not controlled and ensuring they remain controlled when they are lost due to environmental transitions.

Solution Approach

Looking at our combined state space, we find that for increasing D and N as well as fully discretize integers between 0 - 100, the state space explodes in size. This can however be rectified by more coarsely discretizing the allowable resource values from 0 - 100 at steps of 5. Similarly, the same may be done with the tech level, limiting it from 0 - 0.5 at steps of 0.1. Doing so yields a more computationally feasible state space, which will be used for the analysis of this problem.

The POMDP is coded and solved in Julia, using the POMDPs and other similar libraries. This library allows for the generation of POMDPs in quick and user-friendly manners with access to many efficient solvers such as SARSOP, QMDP and POMCP. POMCP was used for solving and simulating the problem described above. Since the problem has a relatively large state space to explore (even after discretizations), we found it best to use a tree based solver that can easily prioritize exploration of the state space. This will allow the algorithm to explore the full state space and action space effectively and find a good solution policy given such.

POMCP was run with an exploration constant of 100 with 20,000 tree queries and a maximum depth of 15. The simulator used a bootstrap particle filter with 5,000 particles and 100 iterations per simulation. The average training and simulation time is around 6 minutes. These results allow for quick optimization of values and changes in code to ensure proper behavior as well as being high enough to allow the agent to learn a proper policy and reach higher rewards.

Results

Overall, the agent was able to effectively solve the problem and made decisions that aligned with the designers desired outcomes of the problem. The following figures describe the results of the POMCP solver and how they aligned with the previously described hopes for the agents learning.

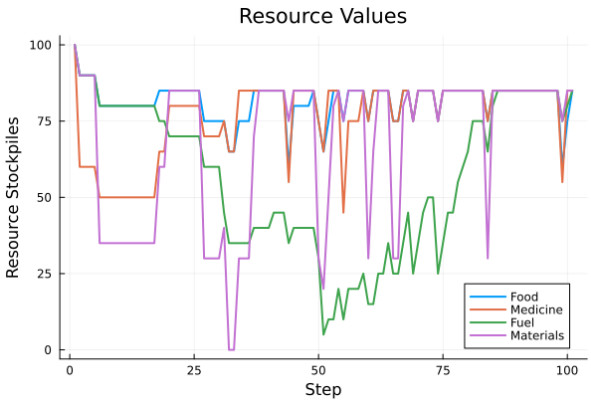

Fig. 1 Resource values over time

Looking back at our reward function, we wished to reward the agent for keeping resources values above defined thresholds, maintaining control over all territories, maintaining squad health and making advancements in technology. By examining the resource values, squad health and tech level over time, we find that the agent managed these values in a way that is desired. The results may not be optimal, however we believe that that is more reflective of reality, where more unforeseen environmental conditions occur.

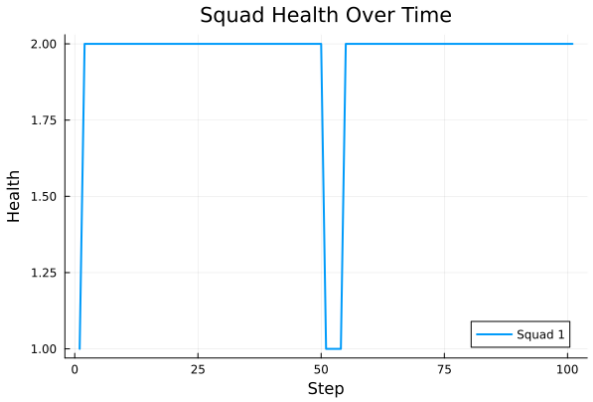

Fig. 2 Squad health over time

Resources values on average stayed about 50% of their maximum levels which we were hoping to see. The only exception to this was the material resource, which often saw the harshest decrease due to its high necessity in many action transitions*. This is one of the more important aspects we hoped the agent would learn, as one of its primary goals was to ensure the safety of its people. We can determine where specific actions such as deployments or researching occurred by examining the values of specific resources. Sharp drops in the "material" resource often correspond to researching new technology. Similarly, harsh drops in "medicine" resources often correspond to filed support or healing actions.

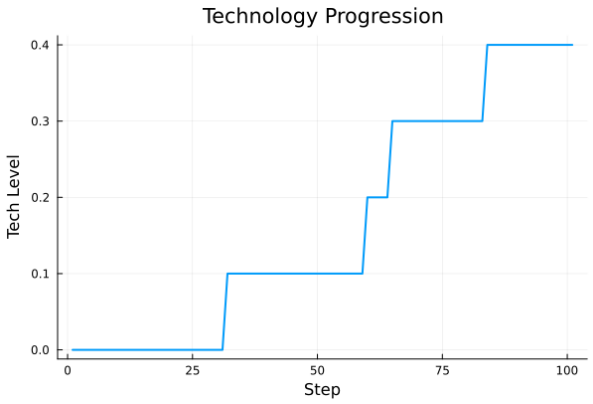

Fig. 3 Tech level over time

*It should also be noted that due to discretizations of the state space that resource values were rounded to the nearest multiple of 5, causing them to seem slightly higher or lower than their actual calculated values.

We also find that squad health values were aggressively maintained in an optimal state, which is somewhat surprising since increasing technology levels commonly corresponds to higher yields per deployment as well (with the exclusion of a failed deployment). It is resource intensive to maintain the health of a squad as aggressively as it did (which can be seen in the "medicine" resource line), so this behavior was unexpected. The authors figured that it might less aggressively balance it between a functional and optimal health status.

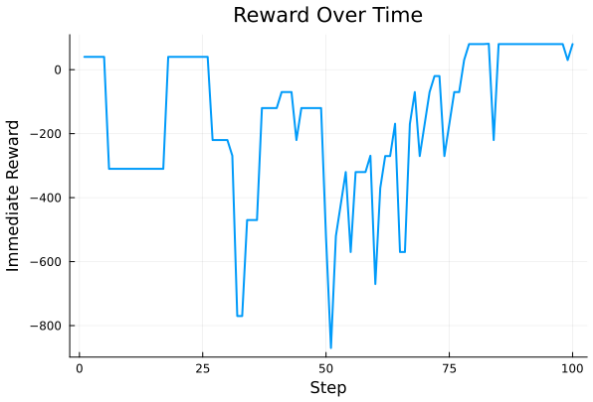

Fig. 4 Reward over time

Similarly, the overall technology level was not a necessarily important behavior we desired the agent to learn (hence the lower reward multiplier) to maximize as we described previously, but we are still happy to see that it did learn to increase it here and there. It also appears the the increases in the overall technology level occurred after researching new subregions. This shows that the agent learned that it would be best to increase its technology level more frequently when it has the ability to earn more resources - which in turn leads to further reward increases.

Conclusion:

Our results found that the agent was able to determine a decent policy for achieving the goals given to it in our defined environment. The agent learned to resource manage in an environment

where resources easily depleted and gain them in more optimal ways by researching technology that would help it yield more resources. The agent discovered that a higher health status of its squads can yield greater returns for resource collection. We can see from the reward over time that the agent eventually stabilizes its reward at a slightly positive number. Given more iterations in the simulation, this value slowly shows a positive growth behavior.

Reflecting on the total behavior of the agent, we can determine it shows satisfying capability to control subregions that probabilistically often slip through its fingers due to aggressive enemy behavior. It learned to research undiscovered subregions such that it can gain access to greater resource yields and achieve its overall goal of controlling all subregions. The agent optimized squad health status's and technology levels to increase its resource yields during deployments, a behavior that we hoped it would flesh out. We found the behavior of the agent is satisfying for such a complex project, it achieves all its goals and showed realistic behavior.

In the future, we plan to expand the scope of the project to model more complex enemy behavior and environmental dynamics. This includes more complex enemy behavior the agent will have to adapt to, more complex transition functions to more accurate reflect reality, and even "special events" that can greatly increase the odds of successful behavior with a very low probability of occurring.

As mentioned previously, built-in toolboxes were used to solve and model this problem. This is unusual for the authors, to which we hope to implement the same process in a custom toolbox with more intuitive and understandable internals. This allows for the authors to better understand the algorithms used to solve such problems.

Contributions and Release:

Contributions:

- Gabriel Agostine: POMDP Formulation, Code and Report

GitHub Project (click here)

The authors do NOT grant permission for this report to be posted publicly.

Acknowledgments:

Special thank you to all involved with this project, technical or otherwise:

- Zachary Sunberg
- Collin Hudson
- Jiho Lee

Citations:

All citations relate to the optimization of agent learning algorithms for war-games and their usages for realistic military operations, officer training and mission optimization. These align with the purpose of this paper, which discusses similar scenarios and come to unique and interesting results.

[1] V. Mittal, J. B. Demarest, K. S. Gilliam, and R. L. Page, "Models of Models: The Symbiotic Relationship Between Models and Wargames," in *Proc. 7th Int. Conf. Simulation Modeling Methodol., Technol. Appl. (SIMULTECH)*, 2017, pp. 215–223.

[2] A. J. Russo, "An evaluation of a management wargame and the factors affecting game performance," M.S. thesis, Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, 1987.

[3] United States Marine Corps, "Simulation training guide," Marine Corps Reference Publication MCRP 7-20A.3, 2023.

[4] K. A. Yost and A. R. Washburn, "The LP/POMDP marriage: Optimization with imperfect information," *Oper. Res.*, vol. 50, no. 4, pp. 607–619, 2002.

[5] K. A. Yost, "Solution of large-scale allocation problems with partially observable outcomes," Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, USA, 1998.

Appendix

The appendix serves to provide more detailed explanations about topics presented in the paper without the limitations on length about the POMDP Formulation.

Action Space A:

$$
\begin{aligned}
A = \{&\text{DEPLOY}(i, j),\\
&\text{RECALL}(i),\\
&\text{ALLOCATE\_HEALING}(i),\\
&\text{ALLOCATE\_FIELD\_SUPPORT}(i),\\
&\text{RESEARCH\_EXPLORATION\_TECH}(),\\
&\text{RESEARCH\_NEW\_SUBREGION}(j)\}
\end{aligned}
$$

where:

- $\text{DEPLOY}(i, j)$: Deploy squad $i \in \{1, \ldots, N\}$ to subregion $j \in \{1, \ldots, D\}$
- $\text{RECALL}(i)$: Recall squad $i \in \{1, \ldots, N\}$ to base
- $\text{ALLOCATE\_HEALING}(i)$: Use medicine to heal squad $i \in \{1, \ldots, N\}$
- $\text{ALLOCATE\_FIELD\_SUPPORT}(i)$: Send resources to deployed squad $i \in \{1, \ldots, N\}$
- $\text{RESEARCH\_EXPLORATION\_TECH}()$: Improve exploration capabilities
- $\text{RESEARCH\_NEW\_SUBREGION}(j)$: Unlock access to subregion $j \in \{2, \ldots, D\}$

Observation Space O:

$$O = (O_R, O_T, O_Q, O_H)$$

where:

- $O_R = [r_1, r_2, r_3, r_4] \in \{0, 1, 2, \ldots, 100\}^4$ represents observed resource levels
- $O_T = [t_1, \ldots, t_D] \in \{0, 1\}^D$ represents observed control status of regions
- $O_Q = [q_1, \ldots, q_N] \in \{0, 1\}^N$ represents observed squad availability
- $O_H = [h_1, \ldots, h_N] \in \{0, 1, 2\}^N$ represents observed squad health levels

Reward Function $R\,(s, a, s')$:

$$R(s, a, s') = \underbrace{R_{\text{territory}}(s, s')}_{\text{Surface control}} + \underbrace{R_{\text{resources}}(s')}_{\text{Resource stability}} + \underbrace{R_{\text{squad}}(s, s')}_{\text{Squad preservation}} + \underbrace{R_{\text{research}}(s, s')}_{\text{Tech advancement}}$$

where each component is defined as follows:

$$R_{\text{territory}}(s, s') = \alpha_t \sum_{j=1}^{D} \max\left(0,\ s'[t_j] - s[t_j]\right) + \beta_t \sum_{j=1}^{D} s'[t_j]$$

$$R_{\text{resources}}(s') = -\gamma_r \sum_{i=1}^{4} \max\left(0, \theta_i - s'[r_i]\right)$$

$$R_{\text{squad}}(s, s') = -\delta_h \sum_{i=1}^{N} \left(0,\ s[h_i] - s'[h_i]\right)$$

$$R_{\text{research}}(s, s') = \epsilon_\tau(\tau' - \tau) + \epsilon_t \sum_{j=1}^{D} \max\left(0,\ s'[t_j] - s[t_j]\right)$$

with coefficients:

$$\begin{aligned}
\alpha_t &= 25.0 && \text{(reward for newly acquired territory)} \\
\beta_t &= 5.0 && \text{(reward for maintained territory)} \\
\gamma_r &= 10.0 && \text{(penalty for resource shortage)} \\
\delta_h &= 5.0 && \text{(penalty for squad damage)} \\
\epsilon_\tau &= 10.0 && \text{(reward for technological advancement)} \\
\epsilon_t &= 10.0 && \text{(reward for discovering new subregions)}
\end{aligned}$$

and minimum acceptable resource thresholds:

$$\theta = [20, 15, 25, 20]$$

Transition Function $T\left(s' \mid s, a\right)$:

The transition function $T\left(s' \mid s, a\right)$ defines the probability of transitioning to state s' given that action a is taken in state s. We define this function in part for each type of action:

Deploy Action Transitions:

When executing a DEPLOY(i, j) action the following occurs if $s[t_j] = 1$ and $s[q_i] = 1$:

$$\begin{aligned}
s'[r_3] &= s[r_3] - \min\left(5j,\ 90\right) \\
s'[q_i] &= 0
\end{aligned}$$

The mission outcome is determined based on two factors: the base success rate of the subregion and the health modifier of the squad.

$$\text{base\_success\_rate}(j) = \begin{cases} 0.8 & \text{if } j = 1 \\ 0.65 & \text{if } j = 2 \\ 0.5 & \text{if } j = 3 \\ 0.4 & \text{if } j = 4 \\ 0.3 & \text{otherwise} \end{cases}$$

$$\text{health\_modifier}(h) = \begin{cases} 0.6 & \text{if } h = 0 \\ 0.9 & \text{if } h = 1 \\ 1.1 & \text{if } h = 2 \end{cases}$$

Three possible mission outcomes can occur:

- **Success:** The squad accomplishes its mission without casualties and returns with maximum possible resources from the region. The probability of this outcome is the product of the base success rate and the health modifier.
- **Partial Success:** The squad accomplishes part of its mission, obtaining some but not all potential resources. There is a 40% chance the squad suffers damage, reducing its health level by 1. The probability of a partial success is 0.05 the base success rate multiplied by the health modifier.
- **Failure:** The mission fails to achieve its objectives. No resources are obtained, and there is a 70% probability the squad suffers damage, reducing its health level by 1. The probability of failure is whatever remains after calculating the success and partial success probabilities.

To model the resources obtained from successful deployments to different subregions, we define a resource collection function reward_distribution(j) that determines the resource yields based on the subregion characteristics:

$$\text{resource\_distribution}(j) = \begin{bmatrix} \text{food\_yield}(j) \\ \text{medicine\_yield}(j) \\ \text{fuel\_yield}(j) \\ \text{materials\_yield}(j) \end{bmatrix}$$

The base yield values for each subregion are defined as follows:

Subregion	Food	Medicine	Fuel	Materials
1	20	30	20	40
2	30	40	30	50
3	40	50	40	60
4	50	60	50	70
5+	60	70	60	80

For each successful deployment, the actual resource yield is determined by:

$$\text{actual\_yield}(j) = \begin{cases} \text{resource\_distribution}(j) & \text{if success} \\ \alpha \cdot \text{resource\_distribution}(j) & \text{if partial success} \\ [0, 0, 0, 0]^T & \text{if failure} \end{cases}$$

With α typically ranging from 0.3 to 0.6 depending on the severity of the partial success. Additionally, if exploration technology has been researched, all resource yields are increased by the current technology modifier τ:

$$\text{final\_yield}(j) = (1 + \tau) \cdot \text{actual\_yield}(j)$$

Where τ starts at 0 and increases by 0.1 (10%) each time exploration technology research is completed.

Recall Action Transitions:

When executing a RECALL(i) action to bring squad i back to base:

$$\begin{aligned} s'[r_3] &= s[r_3] - 5j_i \\ s'[q_i] &= 1 \end{aligned}$$

where j_i is the current region squad i is in. There also exist health consequences:

$$s'[h_i] = s[h_i] - \begin{cases} 1 & P = 0.01 \\ 0 & \text{otherwise} \end{cases}$$

Healing Action Transitions:

When executing ALLOCATE_HEALING(i) the following occurs if $s[q_i] \neq 0$:

$$s'[r_2] = s[r_2] - \begin{cases} 15 & s[h_i] = 0 \\ 25 & s[h_i] = 1 \\ 0 & \text{otherwise} \end{cases}$$

Health improvement probabilities:

$$s'[h_i] = s[h_i] + \begin{cases} 1 & s[h_i] = 0 \\ 1 & s[h_i] = 1,\ P = 0.85 \\ 0 & \text{otherwise} \end{cases}$$

Field Support Action Transitions:

When executing ALLOCATE_FIELD_SUPPORT(i) the following occurs if $s[q_i] \neq 1$:

$$s'[r_1] = s[r_1] - 10$$
$$s'[r_2] = s[r_2] - 15$$

and the following will happen with 85% probability, indicating a successful delivery:

$$s'[h_i] = s[h_i] + \begin{cases} 1 & P = 0.6 \\ 0 & \text{otherwise} \end{cases}$$
$$\tau' = \begin{cases} \tau + 0.1 & P = 0.15 \\ \tau & \text{otherwise} \end{cases}$$

Exploration Research Transitions:

When executing RESEARCH_EXPLORATION_TECH():

$$s'[r_4] = s[r_4] - 40$$
$$\tau' = \tau + \begin{cases} 0.1 & P = 0.9 \\ 0 & P = 0.1 \end{cases}$$

Subregion Research Transitions:

When executing RESEARCH_NEW_SUBREGION(j):

$$s'[r_4] = s[r_4] - \min(20j,\ 90)$$
$$s'[t_j] = \begin{cases} 1 & P = \min\left(0.7 + \frac{\tau}{0.1},\ 1\right) \\ 0 & \text{otherwise} \end{cases}$$

Environmental Transitions:

Regardless of which action is taken, the following occurs:

$$
\begin{aligned}
s'[r_1] &= s[r_1] - 5 \\
s'[r_2] &= s[r_2] - 5 \\
s'[r_3] &= s[r_3] - 5 \\
s'[r_4] &= s[r_4] - 5 \\
s'[h_i] &= s[h_i] + \begin{cases} 1 & s[q_i] = 1,\ P = 0.2 \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in [1:N]
\end{aligned}
$$

Observation Transition Function $Z\left(o \mid a, s'\right)$:

The observation function $Z(o \mid a, s')$ defines the probability of observing o after transitioning to state s' by taking action a. Since resources and squad availability are assumed to be perfectly observable, the observation function primarily affects squad health and territory control:

$$
Z(o \mid a, s') = \prod_{i=1}^{N} Z(o[h_i] \mid s'[h_i]) \cdot \prod_{j=1}^{D} Z(o[t_j] \mid s'[t_j])
$$

where:

$$
\begin{aligned}
Z(o[h_i] \mid s'[h_i]) &= \begin{cases} 0.7 & o[h_i] = s'[h_i] \\ 0.3 & o[h_i] \neq s'[h_i] \end{cases} \\
Z(o[t_j] \mid s'[t_j]) &= \begin{cases} 0.8 & o[t_j] = s'[t_j] \\ 0.2 & o[t_j] \neq s'[t_j] \end{cases}
\end{aligned}
$$

This means health observations have a 70% chance of being accurate, with errors distributed uniformly among the incorrect values. Territory control observations are only 80% reliable, reflecting the challenges of maintaining accurate intelligence on distant regions.