# Agents for the Framework Tribes

Ivan Remus

January 2023

## 1 Abstract

In this paper,we will compare the agents for the Framework Tribes(specifically RHEA,RB,Random) and a new hybrid agent named SRHEA.We will show that SRHEA has potential to have higher statistics than the other agents.

## 2 Introduction

Since DeepBue,a computer program managed to beat Garry Kasparov in 1996,we have managed to make agents that play games better and better,even surpassing the level of the most skilled people in the world.The same happened with AlphaGo and AlphaZero,which managed to outclass Stockfish and Elmo by a big margin.This creates the question,what games do we have that we can still outperform machines?

We will start by talking about one current game called Battle of Polytopia which we will simulate using the framework tribes.

## 3 Agents from the framework Tribes

These agents are relevant to us:

### 3.1 Rule Based

The first agent is a simple rule-based AI that chooses actions based on the current state of the game,without using the FM to reach future game states. It is a hand-crafted heuristic agent modeled with domain knowledge of the game. On each action selection, RB scores each one of the available actions between0and5, to then exe-cute the action with the highest value. In case of a tie, one of the highest-valued actions is selected uniformly at random.Actions are further evaluated by type, prioritizing the most valuable ones to be executed first.Attack

actions are evaluated based on the attacker's at-tack power, the defender's defence and the health of both units. The action is given maximum priority if the target is weaker in defence and HP than the attacker. Attacks receive a retaliation counter-attack if the defender unit is not defeated, hence lower priority is given to attack actions targeting stronger units with high HP.Move actions are penalised if they move the unit in range of stronger enemy units. Higher scores are given to actions that bring units closer to weaker enemy units, villages and enemy cities.Capturing cities and villages,examining ruins and making a unit a veteran are always given the highest possible score. For leveling up, the following bonuses are preferred at each level: extra production; extra resources; city border growth; and spawning a superunit (levels 5 and up).Upgrading and spawning a unit are scored based on the type of unit,resources available and the presence of enemy units within city borders. Similarly,building and resource gathering consider resource and population gain. The Mind Bender's convert action is scored higher for stronger enemy units and its heal action values the number of units affected.Researching a technology is scored based on the tier the target technology is in, giving higher priority to those in lower tiers.Disbanding a unit and destroying a building are never executed5.While this agent incorporates some useful domain knowledge, its main weakness comes from the fact that actions are evaluated independently. Thus, the agent lacks overall planning, unit coordination and an efficient resource man-agement strategy. However, it provides a baseline for comparison with other agents in the game.

### 3.2 Rolling Horizon Evolutionary Algorithm

RHEA (Perez et al. 2013)) evolves sequences of actions(individuals) that are evaluated by executing them from the current state until the end of the individual. The state found at the end of this sequence is evaluated with the same state evaluation function and constitutes the fitness of the individual. Once the given budget is exhausted, the sequence with the highest fitness is selected and its first action returned to the game. Evolutionary operators are applied for selection of individuals, crossover, mutation and elitism. It is common practice in RHEA to use a shift buffer,which moves all the genes one position towards the start of the genome after each action selection, adding a random action at the end to keep the predetermined length.

## 4 Game States

The game state for the framework Tribes is more simple/less complex than the game state of the Battle of Polytopia.

### 4.1 Game State for the Framework Tribes

A game at a current moment has the following components:

1. Game Information

2. Technologies
3. Board
4. Ranking
5. Diplomacy

#### 4.1.1 Game Information

We will keep the information relevant to the game,such as information about who's turn is next,the game mode type etc...

The tribes in the game ,along with their start technology(technology already researched) and start unit(the unit placed in the capital at turn 0) are:

1. XIN_XI starts with technology CLIMBING, starts with unit WARRIOR
2. IMPERIUS starts with technology ORGANIZATION, starts with unit WARRIOR
3. BARDUR starts with technology HUNTING, starts with unit
4. OUMAJI starts with technology RIDING, starts with unit
5. KICKOO starts with technology FISHING, starts with unit
6. HOODRICK starts with technology ARCHERY, starts with unit
7. LUXIDOOR starts with technology -, starts with unit WARRIOR
8. VENGIR starts with technology SMITHERY, starts with unit SWORDMAN
9. ZEBASI starts with technology FARMING, starts with unit WARRIOR
10. AI_MO starts with technology MEDITATION, starts with unit WARRIOR
11. QUETZALI starts with technology SHIELDS, starts with unit DEFENDER
12. YADAKK starts with technology ROADS, starts with unit WARRIOR

The tribe LUXIDOOR doesn't start with a technology,but starts with a city of level 3.

For information about the type of game settings:

- game mode type(either CAPITALS or PERFECTION/SCORE)
- the number of players

- length/number of tiles for the square which will form the board ,which we will note as $N$
- a Boolean value to see if the game is played using Fog of War or not.If False,each player can see the entire board.Otherwise,only the tiles explored
- a Boolean value to see whether the game is over

For each player we will keep

- list of ids for each city a player owns
- the capital id
- the tribe chosen by the player
- current number of stars
- score
- N X N matrix for the observable grid of a player(which areas are explored and which aren't)
- list of ids for the connected cities with the capital
- the status for each monument(available to build,already built or not available)
- list of ids for the tribes this player/tribe has met
- list of units that don't belong to a city
- number of kills
- number of turns without attacking a player,only if Meditation is researched

For information about whose turn it is:

- the current turn
- the id of the player whose turn it is
- a list of ids of the players which is generated at the beginning and specifies the order in which the agents are taking turn
- the current position of the player who is making the move

#### 4.1.2 Technologies

Each player will have a tree of technologies to research,and each tribe starts with a technology already researched.Based on that,the player can acquire other technologies. Each technology has a tier(from 1 to 3) which determines their cost of research(along with the number of cities currently owned) and a prerequisite technology (what technology needs to be researched before being able to research this one)

The technologies for each tribe are as follows:

1. CLIMBING , tier 1, requires -
2. FISHING , tier 1,requires -
3. HUNTING , tier 1,requires -
4. ORGANIZATION , tier 1,requires -
5. RIDING , tier 1,requires -
6. ARCHERY , tier 2,requires HUNTING
7. FARMING , tier 2,requires ORGANIZATION
8. FORESTRY , tier 2,requires HUNTING
9. FREE_SPIRIT , tier 2,requires RIDING
10. MEDITATION , tier 2,requires CLIMBING
11. MINING , tier 2,requires CLIMBING
12. ROADS , tier 2,requires RIDING
13. SAILING , tier 2,requires FISHING
14. SHIELDS , tier 2,requires ORGANIZATION
15. WHALING , tier 2,requires FISHING
16. AQUATISM , tier 3,requires WHALING
17. CHIVALRY , tier 3,requires FREE_SPIRIT
18. CONSTRUCTION , tier 3,requires FARMING
19. MATHEMATICS , tier 3,requires FORESTRY
20. NAVIGATION , tier 3,requires SAILING
21. SMITHERY , tier 3,requires MINING

22. SPIRITUALISM , tier 3,requires ARCHERY
23. TRADE , tier 3,requires ROADS
24. PHILOSOPHY , tier 3,requires MEDITATION

For each player we will have a vector of length equal to the number of technologies that specifies which technology has been researched by the player.

#### 4.1.3 Board

The board will describe a N x N matrix where each tile has a given terrain,resource,building,military units and each tile is either neutral or belonging to a city at some point of the game.

We will have a length of the board $N$ and the board will be represented as a matrix of $NXN$.The board is described as:

1. N X N matrix of terrains to represent each type of terrain
   (a) DEEP_WATER
   (b) SHALLOW_WATER
   (c) PLAIN
   (d) FOREST
   (e) MOUNTAIN
   (f) VILLAGE
   (g) CITY
   (h) FOG
2. N x N matrix of resources to represents each type of resource. Each resource has an id,map character,cost,bonus and a necessary technology to research before being able to exploit the resource.
   (a) FISH
   (b) FRUIT
   (c) ANIMAL
   (d) WHALES
   (e) ORE
   (f) CROPS
   (g) RUINS
3. N x N matrix of buildings to represents each type of building.Each building has an id,cost,bonus,technology requirement and a terrain requirement.

(a) PORT
(b) MINE
(c) FORGE
(d) FARM
(e) WINDMILL
(f) CUSTOMS_HOUSE
(g) LUMBER_HUT
(h) SAWMILL
(i) TEMPLE
(j) WATER_TEMPLE
(k) FOREST_TEMPLE
(l) MOUNTAIN_TEMPLE
(m) ALTAR_OF_PEACE
(n) EMPERORS_TOMB
(o) EYE_OF_GOD
(p) GATE_OF_POWER
(q) GRAND_BAZAAR
(r) PARK_OF_FORTUNE
(s) TOWER_OF_WISDOM

4. N x N matrix of integers to placements of units on the board for each tile.
5. N x N matrix of integers to represents if a tile is under control by a city(-1 means the tile is neutral/not belonging to a city).
6. A map to associate the id of actors(referring to military units,cities and buildings)
7. For each player we will also hold:
   (a) N x N matrix of Boolean values to represent if a tile contains a road. We will assume a tile contains a road if it's a road,port or city(In Battle of Polytopia,a village is considered a road)
   (b) N x N matrix of Boolean values to represents if a tile is navigable

#### 4.1.4 Ranking

Keep information about each player's progress in the game and compare them to create a ranking system.

For each player,we will have associated using the players id the following:

1. score

2. number of technologies researched

3. number of cities controlled

4. number of stars produced this turn/production

5. number of start available(how many starts the player still has)

The ranking will be updated using the following rules: -if a player has won the game,they will be first -else ,the first to win a criteria in the given order will be ahead of another player -if the criterias above are still not enough,then choose at random

#### 4.1.5 Diplomacy

We represent this part of the game as a matrix of M x M ,where M is the number of players.For a cell at position $(i, j)$ we define

$$allegiance[i][j] = \text{ how much trust/favor } player_i \text{ has for } player_j$$

Negative values means distrusts/hate and positive values means the opposite.If the value is 0,then they are neutral. These values will be updated after each action of a player.

#### 4.1.6 Player observation

For a player,an observation will be an incomplete copy of the current game state which will contain the information that the player is allowed to know/see.Thus,for the player whose has the current turn,we will compute the game state observation.

From game information we will get:

- From game settings:
  - game mode
  - number of players
  - length of the square/number of tiles for the side of the square noted as N
- From the information associated with each player
  - the player id
  - the cities the player has
  - the capital
  - tribe chosen

  - number of start left to use
  - current score
  - N x N matrix of the currently observable tiles
  - the connected cities with the capital
  - for each monument,it's status(available to build,already built or not available)
  - the tribes/players met
  - number of units killed
  - number of turns without attacking a player (only changes if the technology MEDITATION is researched)
- From the turn order:
  - the current turn
  - the order of the players to take turns

From technologies,we will only give the player their technology tree.Thus,for each technology the player will get:

- the id of the technology
- the tier of the technology
- the status of the technology(researched,cannot research or possible to research)
- the cost to research in number of stars

From the Board,only the information about the tiles that have been explored according to the observable grid of this player.In the case that Fog of War is false,then we expect the observable grid to be all the board.

Also,we will give incomplete information about other cities,meaning a player can only see the position of an enemy city and what unit it recruited at a turn and for the cities owned by the player,the information about the number of stars per turn etc.

From the ranking , we will get all the information about each player.

From the diplomacy,the allegiance values of the other players to this player.So,the current player doesn't clearly know the relationship between 2 different players.

## 4.2 Action Space

We refer to an action as a move that a player is allowed to make when its their turn for the given game state that changes the game in one way or another.

We can split the actions in 3 categories:

1. City Actions
2. Tribe Actions
3. Unit Actions

### 4.2.1 City Actions

Represent the actions that are possible to be done in the borders of a given city.They are:

- Build a Building
- Burn Forest
- Clear Forest
- Destroy Building
- Grow Forest
- Level Up a City
- Gather Resource
- Spawn Unit

### 4.2.2 Tribe Actions

Represent the actions that a certain player can do at a time as a leader of the respective tribe.They are:

- Build Road
- Research Technology
- Send Stars
- Declare War
- End Turn

The actions of Send Stars and Declare War contribute to the allegiance values from diplomacy.

#### 4.2.3 Unit Actions

Represents the action that a unit in the game is able to do.They are:

- Attack Unit
- Capture City/Village
- Convert Unit(specific to the unit Mind Bender)
- Disband Unit
- Examine Ruins
- Heal Adjacent Units(specific to the unit Mind Bender)
- Make Veteran Unit
- Move Unit
- Recover Health for Unit
- Upgrade Naval Unit

# 5 Implemented Agents and Progress

## 5.1 Reinforcement Learning Agent

### 5.1.1 Environment

Firstly,although the framework Tribes offers a Forward Model as well as a way to play the game,this framework does not behave as an environment.The fact that the framework is the one controlling when an agent acts produces the following obstacles:

- RL Agent can't see the game state after choosing an action to act,due to the game having control of who is called at a given moment(using the advance() method offered by the Forward Model,we could bypass this,but we would have 2x the operations of changing a game state ,which i considered to not be a good choice,since RL Agent already take a long time to train)
- Having multiple RL Agents in the game and wanting to add their experience together to form one RL Agent for better results,we will have to do some bypassing which is not great
- Requires changing the implementation and algorithm of an RL Agent
- How do make the difference between training and playing for the agent?

Based on these reasons,we choose to implement the game as an environment,thus giving the agents the possibility to call the game when they want to give a move.This also only meant chaning the order of the operations in the game,rather than having to rewrite everything from scratch.

Unfortunately,either due to problems/bugs already existing in the framework (some on the animation side which were based in the previous implementation logic ) as well as poor management time by the author,we didn't manage to assure the features that the game had in this environment,at least not enough to have an environment that works 100% of the time.

### 5.1.2 Agent Implementation Ideas

For the Reinforcement Learning Agent,we first started with the description of a game state and action space to get a better grasp of how we should model the neural network of our agent.We tried to implement the Q Learning Algorithm,since it is well known and hoped it would be a good practice in preparation to implementing better RL Algorithms,such as Soft Actor Critic or Policy Gradient.

From their description,we can see that the game state given has a variable length based on the N(length of the square) and P(number of players in game) ,but this cam be given as multiple N X N matrixes to try to make the input size layer to have a fixed size.

For each game state,the number of actions available to a given player is variable,meaning we will have to represent the output layer which has a fixed number of neurons in a different way.

These ideas refer to the implementation of an Q Learning agent.

The first idea,which is to represent the entire action space with the output layer and have each neuron say a probability or Q Vaue for an encoded action is not feasible, Because of some actions like MOVE and ATTACK,which are described by 2 position (line,colomn) for a cell in the square.This means,for a map of length N,we will have to add at least $N^2 * N^2 = N^4$ to represent these actions,and another $N^2 * M$(M is a constant for how many moves we can make on average on a cell) to represent other actions,which have a smaller action space.This means that for $N = 11$ we will have a output layer of size 15000,so it's not feasible.

The second idea for implementation was to give a neural network a pair (Game State,Action) and have it give back a Q Value.We would choose our next action by getting the scores of each pair (Game State,Action) and choosing the one with the highest Q Value.This would have been the idea implemented.

## 5.2 Simple Rolling Horizon Evolutionary Algorithm Agent

### 5.2.1 Idea

The agent we chose to implement is a combination between RHEA Agent and Rule Based/Simple Agent which we will cal Simple RHEA.The idea behind the combination is the following:

- Simple/Rule Based Agent explores the space by choosing what it considers to be the best move at a given moment,thus we can say this algorithm focuses on maximizing short-term rewards
- RHEA uses a population of individuals,each with a sequence of action and a heuristic function to evaluate the fitness of an individual by computing the sequence of actions and seeing the relative improvements relative to the previous game state.Thus,this agent would try to find a sequence of actions that maximizes the long-term rewards.
- Based on the previous paper,we can see that RHEA and Rule Based agents perform the best out of all the agents

Based on the above points,we will try to combine these 2 implementation to attempt to get the best of both.

### 5.2.2 Differences between RHEA and SRHEA

The following points are what the SRHEA agent has which RHEA does not:

1. Rule Base Action Evaluator that gives an action in a given game state a score.This score will then be used to proportionally select an action from a game state.With this , we incorporate 2 variables for probabilites,which will say how much we choose an action randomly or proportionally to the score.The idea would be to try to take from the Rule Based agents some actions it considers good ,and use the RHEA agent to check their long-term rewards.
2. Shift operation that is applied to a percent of the best individuals at a moment,after initializing the population.The idea is to try to retain information previously discovered (doing exploitation) and the rest of the individuals to be created randomly(doing exploration)
3. 2 Selection Operators:Tournament Selection(only the individual with the highest fitness value out of randomly picked participants will be chosen ) and Roulette Wheel Selection(choose an individual proportionally to it's fitness value)
4. 2 Cross Over Operators:Uniform(select uniformly from each parent) and Combinatorial(combine the action from the 2 parents as much as possible)

#### 5.2.3 Rule Base Action Evaluator

We use this Rule Base Action Evaluator to assign a score to each action according to the Rule Based/Simple Agent.Using this ,we define in our model 2 probabilities:

1. INIT_RANDOM_PROBABILITY: the probability to choose randomly an action when creating an individual at the initialization of the population.The 1 - INIT_RANDOM_PROBABILITY is the probability to choose an action proportionally to the score of each action

2. SHIFT_RANDOM_PROBABILITY : the probability to choose randomly an action when creating an individual after shift operation that precedes it.The 1-SHIFT_RANDOM_PROBABILITY is the probability to choose an action proportionally to the score of each action when doing the shift of the population,only for the individuals not shifted

#### 5.2.4 Shift Operation

After we chose the action from the best individual as the action we will play,all the first actions of every other individual will be invalid,so we need to delete the first action from each individual.In this operation,we use the parameter SHIFT_MOVED_SIZE to determine the number of the fittest individuals which we will try to delete the first action and save the following actions(exploitation) and the rest to be generated according to the 2 probabilities(exploration).

#### 5.2.5 Selection Operators

We added the implementation of 2 selection operators(the Tournament Selection operator is found in RHEA as well). The operator that our agent will use is given by the parameter selection_type;

#### 5.2.6 Cross Over Operators

We added the implementation of 2 cross over operators named Uniform and Combinatorial(Uniform is found in RHEA as well).

## 6 Experiments

The following agents are going to be comparared by a Round Robin Tournament:SRHEA,RHEA,RuleBase/Simple and Random.

## 6.1 Experimental Setup

### 6.1.1 Banned Actions

For each agent ,we have excluded/forbidden the following actions to be played(based on the fact that DISBAND and DESTROY are only useful in higher play level and SEND STARS hides how much better at agent is at using it's stars proficiently instead of randomly receiving more stars):

- DISBAND
- DESTROY
- SEND STARS

### 6.1.2 Round Robin Tournament Settings

The game mode chosen is CAPITALS,and each players plays against another player on a map.The starting tribes are Bardur and Imperius and they each player will have the tribes shifted(to see how they perform in the other role).

We also choose 6 level seeds(no preferential choice) and the number of repetitions on each level to be 5,so between 2 players we will get 30 levels. The level seeds are in the order played:

- 1590191438878
- 1590791907337
- 1591330872230
- 1590557911279
- 1590557911279
- 1590597667781

### 6.1.3 Parameters for SRHEA and RHEA

Both SRHEA and RHEA have the following parameters equal(to ensure they have the same computational resources):

- the heuristic method used is DIFF_HEURISTIC_V2
- $epsilon = 1e - 6$ used for noise when choosing between individuals
- $no\_iterations = 5$ to stop after 5 generation of the population
- $POP\_SIZE = 100$
- $MUTATION\_RATE = 0.1$ when we apply the Uniform Cross Over

- $INDIVIDUAL\_LENGTH = 20$ the max number of actions in a sequence
- $TOURNAMENT\_SIZE = 10$ the number of individuals when doing a selection tournament
- $MUTATE\_BEST = 10$ how many individuals to mutate from the best individual

One parameters where SRHEA and RHEA differ is the $ELITE_SIZE$ which is equal to 5 on SRHEA and 1 to RHEA.

Also,the following parameters that are specific to SRHEA:

- $SHIFT\_MOVED\_SIZE = 30$
- $INIT\_RANDOM\_PROBABILITY = 0.5$
- $SHIFT\_RANDOM\_PROBABILITY = 0.5$
- $selection\_type = TOURNAMENT$
- $crossover\_type = COMBINATORIAL$

## 6.2 Results

| Agent vs Agent | SRHEA | RHEA | Rule Based | Random |
|---|---|---|---|---|
| SRHEA | * | 46.67 | 30.00 | 100 |
| RHEA | 53.33 | * | 53.33 | 100 |
| Rule Based | 70.00 | 46.67 | * | 93.33 |
| Random | 0 | 0 | 6.67 | * |

Table 1: Agents winning rate

| Agent vs Agent | SRHEA | RHEA | Rule Based | Random |
|---|---|---|---|---|
| SRHEA | * | 12833.83 | 9587.00 | 13580.50 |
| RHEA | 13302.67 | * | 11960.50 | 13858.50 |
| Rule Based | 10530.50 | 9822.83 | * | 8988.17 |
| Random | 6270.67 | 5202.23 | 4952.67 | * |

Table 2: Agents average score

| Agent vs Agent | SRHEA | RHEA | Rule Based | Random |
|---|---|---|---|---|
| SRHEA | * | 21.83 | 9.97 | 23.30 |
| RHEA | 19.33 | * | 13.17 | 20.91 |
| Rule Based | 26.63 | 23.90 | * | 22.97 |
| Random | 6.93 | 7.23 | 4.33 | * |

Table 3: Agents average star production

### 6.3 Comparison

From the table 1,we can see that the agent with a winning rate above 50% for all agents is RHEA,which seems to do a little better than SHREA and Rule based in this category.Also,Rule Based would be placed in second place,since it clearly dominates SRHEA ,but it also lost a couple of rounds against the Random Agent.

From table 2,it seems that RHEA is the agent with the highest average score,followed by SRHEA and Rule Based.SRHEA is able to compete against Rule Based in terms of score,even trough it loses against Rule Based,suggesting that SRHEA could be better in a match for points rather than capitals.

From table 3,the agent with the most star production in general is Rule Based by a big margin,followed by RHEA and SRHEA.

From these tables,we can observe that RHEA seems to be the most balanced one,since it focuses on both economy and military in a balanced way,where as SRHEA focuses more on economy and less on military,leading to a lower winning rate and star production but with an average score better than RB. RB seems to be the one that focuses on military,taking a decent win rate as well as a high average production star by winning the cities of their enemies,but not so much on score as RHEA ,but close enough to Rule Based.

### 6.4 Conclusion

although SRHEA doesn't win any category chosen,it still shows potential by placing second place at the score average table,showing that it does work as a hybrid between RHEA and Rule Based.With fine tuning of the parameters or a grid search of the parameters,it might be possible to find better parameters to manage to increase it's rating.

## 7 References

1. Tribes: A New Turn-Based Strategy Game for AI Research

2. Framework Tribes GitHub

3. Rolling Horizon Evolutionary Algorithms in General Video Game Playing

4. Population seeding techniques for Rolling Horizon Evolution in General Video Game Playing

5. Rolling horizon evolution enhancements in general video game playing

6. Acquisition of chess knowledge in AlphaZero