Bolster-MAS: AASMA 2021 Project Report

Group 49

João Fonseca  
 Instituto Superior Técnico  
 89476  
 Lisbon, Portugal  
 joaomffonseca@tecnico.ulisboa.pt

Tomás Lopes  
 Instituto Superior Técnico  
89552  
 Lisbon, Portugal  
 tomas.lopes@tecnico.ulisboa.pt

Carlos Marques  
Instituto Superior Técnico  
 98639  
 Lisbon, Portugal  
 carlos.a.marques@tecnico.ulisboa.pt

ABSTRACT

Turn-based strategy (TBS) games have dramatically increased in popularity over the past decades, and as such, have become a popular field for the development of Artificial Intelligence. As the scope of video games starts going beyond mere entertainment into areas such as corporate management and military use, using intelligent systems to truly understand how to make the most out of the complex interactions these games allow becomes increasingly important.

In this paper, we showcase our simple medieval-style TBS game, which we call “Bolster”. We also discuss the key characteristics of the competition-based multi-agent system associated with it, “Bolster-MAS”. In this system, each agent, implemented using stateful extensions to the classic subsumption architecture, represents a player controlling a village, competing in a free-for-all scenario. Finally, we analyze the performance of the agents based on a set of metrics. [TODO – summary of analysis]

1 Introduction

As turn-based strategy (TBS) games have increased remarkably in fame since their inception in the late 1970s, they have also become an increasingly popular focus area in the AI field. Intelligent agents often have the ability to navigate significantly complex environments such as these with great effectiveness; hence, as video games start showing uses in many other relevant fields, devising adequate strategies for these games becomes pertinent.

With this project, our goal was to explore the subject of competition-based multiagent systems in the context of TBS games, taking inspiration from already existing games such as the Civilization series[[1]](#footnote-2) and other similar games such as Tribal Wars[[2]](#footnote-3) to create an environment which corresponds to a simpler, more basic version of these games where each agent acts as a player, competing in a free-for-all scenario until one of them is crowned the winner. The end result of this was the game “Bolster”, and the corresponding multi-agent system, “Bolster-MAS”.

In short, in Bolster, each player manages a village in a medieval setting where they must take decisions regarding resource management and troop mobilization to defeat other players, while also staying alive and maintaining the prosperity of their village. In Bolster-MAS, agents follow a simple priority-based approach, which, when used in combination with each agent’s state and memory, determines the decisions taken by each agent throughout the game.

2 Environment: Bolster

The environment in which the agents operate consists of the game Bolster. This game comprises a group of villages, each of them belonging to its own player. Each player can build new infrastructure for their village, upgrade it, and take advantage of it to increase its productivity and grow its military force.

There are seven available buildings, which are all upgradeable and all serve different purposes:

* The mine provides iron to the village. The more it is upgraded, the more iron is extracted. It starts at level 1, producing 20 iron per turn, and can be upgraded to level 5, reaching a production of 320 per turn.
* The quarry provides stone to the village. The more it is upgraded, the more stone is extracted. It starts at level 1, producing 20 stone per turn, and can be upgraded to level 5, reaching a production of 320 per turn.
* The sawmill provides wood to the village. The more it is upgraded, the more wood is gathered. It starts at level 1, producing 20 iron per turn, and can be upgraded to level 5, reaching a production of 320 per turn.
* The warehouse is where all the village’s resources are stored. Upgrading it will increase the maximum storage capacity of each resource. It starts at level 1, storing up to 200 of each resource, and can be upgraded to level 5, reaching a maximum storage of 3200.
* The farm determines how big the population of the village can be. The higher it is upgraded, the more population the village will be able to sustain, increasing the maximum number of peasants. It starts at level 1, sustaining up to 10 peasants, and can be upgraded to level 5, reaching a maximum population of 810.
* The barracks are used to recruit troops. Upgrading the barracks unlocks the recruitment of different, more specialized types of troops. The barracks start at level 0 and can be upgraded to level 3. More on this below.
* The wall provides a defensive bonus to the troops defending the village. The higher it is upgraded, the higher the defensive bonus is. It starts at level 0, providing no defense bonus, and can be upgraded to level 3, providing a multiplicative defense bonus to the village of 1.5.

Building, upgrading, and recruiting all have their own specific resource costs, which are divided among the three different resources already mentioned: stone, iron, and wood. Upgrade costs get exponentially higher with each building level. Recruiting troops also consumes a peasant per unit alongside the other three resources, and demoting said unit returns that peasant.

There are four classic types of units available, each with a different purpose, excelling at a specific task:

* Warriors are the cheapest units to recruit but are also the weakest. They are the first unit available in the barracks (level 1), with equal offensive and defensive power (2 ATK, 2 DEF), and have a relatively cheap cost.
* Archers are the second available unit (level 2). They are more expensive than Warriors, and they excel at defense (2 ATK, 7 DEF).
* Catapults are made available at the same time as Archers (level 2) and, in opposition, excel at offense (7 ATK, 2 DEF).
* Cavalrymen are the last unit available in the barracks. They are the strongest unit in the game, with both high defense and offense (8 ATK, 8 DEF).

Units can be used to attack other villages. The outcome of an attack is determined by which side has higher power, calculated based on the attack or defense power of its unit types (depending on the side), the number of units of each type and a random luck factor. The losing army is wiped out, whereas the casualties of the winning army are based on the magnitude of the win and by a separate luck factor.

In the case of a victorious attack, the attacking army steals resources from the defending village and depletes its health points, based on the offensive power of its surviving troops. Once a village’s health points drop to zero, the player controlling that village loses. The last surviving player is crowned the winner.

After an attack takes place, both the attacker and the defender obtain a detailed report with information that includes the turn in which the attack took place, starting/ending troops on both sides, plundered resources, damage dealt, and each army’s total power. The exception to this is for the case of failed attacks: if a player sends out an attack and loses, that player will not get any information about the opponent, only receiving information about the overall outcome of the fight and casualties.

There is also a fifth unit in the game, which functions differently from other units. Spies are the most expensive unit in the game; even so, they can be recruited with level 1 Barracks. They cannot be sent on normal attacks, cannot die during village defenses and they do not have offensive or defensive power; instead, they can only be used for spying other villages. Spying a village consumes a spy and returns a static copy of the state of the spied village at the time of spying. This copy is called an “Espionage” .

Every turn, each player may take, sequentially, up to one upgrade decision (erecting a new building or upgrading an existing one), one recruitment decision (recruiting or demoting any number of units of a specific unit type), one spying decision (sending off a spy to obtain an espionage of another village) and one attack decision (sending off a specified number of troops of specific unit types to attack a single village).

Each player possesses complete information on the state of its village, including buildings and troops, as well as the name of the villages of the players that have not lost. Additionally, the player also has access to a complete attack report and espionage history.

3 Multi-Agent System: Bolster-MAS

The Bolster-MAS (Bolster Multi-Agent System) consists of a set of agents, each competing against one another in an attempt to win games of Bolster. Each agent represents a player, taking the aforementioned decisions for its own village, with the ultimate goal of being the last survivor.

Each agent possesses a set of sensors and actuators. The sensors provide the agents with the information that each player has access to, mentioned at the end of section 2. The actuators allow the agents to create new buildings, upgrade existing ones, recruit and demote units, spy other villages, and send attacks.

The agents in Bolster-MAS were implemented based on the subsumption architecture, extended to include relevant state, personality, and “short-term” memory. This combination of reactive architectural principles and stateful extensions makes for agents with hybrid behavior.

A purely reactive agent would be unsuitable for this kind of environment: in turn-based strategy games such as Bolster, selecting the next move implies careful consideration of recent events, which may lead to changes in strategy and personality required to obtain the best results and to keep up with the other players. With that said, however, the crux of the decision-making process can be done based on simple “principles”, evaluating each one in descending order of priority. In the case of Bolster-MAS, this priority order may change depending on the personality (henceforth named “Stance”) of each agent and/or recent events.

3.1 Memory

Each agent stores, in its memory, different kinds of data. This memory is “short-term” in the sense that although in some cases, all the events regarding a player’s village are stored, only the most recent ones are typically considered during state/stance changes and decision-making. This data includes:

* A log of all the attack reports associated with its village.
* A log of all the espionages sent by the agent.
* The total attack power values of its village over the last ten turns.
* Turns elapsed since last attack.
* Turns elapsed since last defense.
* Turns elapsed since last failed attack.
* Turns elapsed since last failed defense.

The reason why only the most recent information is considered, particularly in the case of attack reports and espionages, is that as the game progresses, the information in these reports becomes outdated, since these do not keep track of the growth of the enemy villages, acting as a simple “snapshot” of the state of the opponents’ villages at the time of the event.

Overall, agents are quite conservative regarding their decision-making. For instance, agents will opt not to send out attacks blindly, only sending out attacks if the agent has access to any kind of information that shows that in a recent turn, the opponent was weaker than itself. More on this in a later section.

3.2 Stance

Each agent starts with one of three possible stances:

* Offensive – favors the recruitment of offensive troops and takes higher risks when sending out attacks.
* Defensive – favors the recruitment of defensive troops and is more conservative when sending out troops, always making sure to stay well defended, even when sending out attacks. Also prioritizes defense-oriented buildings, such as the Wall.
* Neutral – a middle-ground between the two previous stances, favoring the recruitment of hybrid troops that can attack and defend with equal effectiveness.

Each stance is associated with a set of parameters that influence decision-making. These parameters include:

* Recovery turns – determines the number of turns to wait after a defeat before continuing to send attacks to villages it has attacked recently. The more offensive an agent is, the lower this number is, implying less cautious behavior.
* Last turns considered – determines the number of recent turns that an agent can consider when looking at attack reports and espionages. The more offensive an agent is, the higher this number is, making it more prone to making decisions based on outdated information.
* Win magnitude – determines the ratio between its own attack power and an enemy village’s defensive power (from an espionage) required in order to make said village eligible for an attack. The more offensive an agent is, the lower this number is, implying a riskier behavior.
* Own attack power – determines how to compute its own attack power, namely whether to include the offensive stats of its defensive troops or not.
* Turns without fighting – determines how many turns without attacking or defending must elapse before its stance’s offensiveness is increased.
* Warrior/archer/catapult/cavalrymen send ratio – determines the percentage of each troop to send off in attacks. The more offensive an agent is, the more likely they are to send a higher percentage of troops, leaving its own village less defended.

This stance is dynamic: depending on the circumstances of the game, an agent can change its own stance at the beginning of each turn. This can occur in the following situations:

* If an agent’s village health drops below 25%, that agent becomes defensive.
* If a non-offensive agent does not attack and is not attacked for a specified number of turns, its stance becomes more offensive (defensive to neutral or neutral to offensive).
* If a non-defensive agent suffers a large reduction in its attack power, its stance becomes more defensive, depending on how significant that reduction is (offensive to neutral, offensive to defensive, or neutral to defensive).

Additionally, each agent is initialized with a “troop focus” parameter, randomly generated between 0.3 and 0.7, which determines the fraction of troops that each agent is willing to spend on troops in a single recruiting decision. This value is independent of the stance.

3.3 Decision-making

Given what we have examined so far regarding agent memory and stance, we can now explain each agent’s decision-making process.

For each of the four decision types (upgrading, recruiting, spying, and attacking), the same overall structure is followed. Firstly, an option generation function is used to determine all the possible decisions that can be taken based on the current resource count and available peasants. Then, a filter function is called to prioritize the generated options. Finally, the highest-priority decision is executed – in some situations, this may be simply doing nothing.

3.3.1 Upgrade decision

Upgrade decisions are based on the following priority system, with the agent executing the first decision that it can take based on the generated options:

1. If the farm cannot hold more peasants, upgrade it.
2. If resource production is above half of the warehouse capacity for any resource, upgrade the warehouse.
3. Upgrade the barracks.
4. If the agent is defensive, upgrade the wall.
5. Upgrade the resource camp of most the lacking resource.
6. If the agent is not defensive, upgrade the wall.
7. If the warehouse is full of at least one resource, upgrade it.
8. If the warehouse is full, upgrade the farm.
9. Upgrade nothing.

3.3.2 Recruit decision

Recruit decisions are based on the following priority system:

1. If the village has no available peasants, the farm is at max level and there are non-cavalrymen, non-spy units in the village, demote up to 20 units, prioritizing the demotion of warriors, archers (if offensive), catapults (if defensive), archers (if not offensive), and catapults (if not defensive).
2. If the village has 5 or less spies, recruit up to 3 spies.
3. Recruit cavalrymen.
4. If the agent is offensive, recruit catapults.
5. If the agent is defensive, recruit archers.
6. Recruit warriors.
7. Recruit nothing.

3.3.3 Spying decision

Spying decisions are based on the following priority system:

1. If the agent has available spies, spy a random village that it has not spied over the last ten turns.
2. Spy nothing.

3.3.4 Attack decision

Attack decisions are based on the following priority system:

1. If the agent has recent victorious attacks against villages and is not in recovery mode, re-attack one of those villages at random.
2. If the agent has a recent espionage against villages and its own attack power is greater than the defense power of the spied villages by a certain magnitude, attack the one with the lowest defense power.
3. Attack nothing.

4 Analysis/Evaluation

We had twelve agents – four starting with a defensive stance, four with a neutral stance, and four with an offensive stance – play 2500 games of Bolster. The results, as well as agent histories and state information for all 2500 games, were persistently stored. Unless explicitly mentioned otherwise, all the analyses and evaluations done in this report will be based on the data of these simulations.

4.1 Metrics

A wide array of metrics was used to analyze the performance of Bolster-MAS. These metrics include:

* Number of turns per game
* Number of turns each agent stays alive per game
* Decisions taken per agent per game
* Successful and failed attacks per agent per game
* Successful and failed defenses per agent per game
* Total troop casualties per unit per agent per game
* Troop casualty history per agent per game
* Village health history per agent per game
* Prosperity rating history per agent per game
* Attack power history per agent per game
* Defense power history per agent per game

4.2 Results analysis and empirical evaluation

[TODO – analysis]Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00

1. To find out more: <https://en.wikipedia.org/wiki/Civilization_(series)> [↑](#footnote-ref-2)
2. To find outmore: <https://en.wikipedia.org/wiki/Tribal_Wars> [↑](#footnote-ref-3)