MATH 381 Summer 2021

Efficient Armies in StarCraft II

Connor Wise, Murtaza Jafry

Background

In the real-time strategy game StarCraft II, one must gather and use resources to build up their base and army in an attempt to destroy the other player's base and army. The resources used to create one's base and army are limited, so one needs to be careful about how much they are spending. The two main resources, minerals and gas, are gathered over time at one's base and are used to generate new structures, research upgrades, and produce new units. Though not explicitly a resource spent in the game, time is also valuable as each unit takes time to produce. When building an army, a typical player wants to minimize both the amount of resources and the time spent producing their units.

Army composition is a huge aspect of the game. Each army is composed of a variety of different units, each with their own health, damage, and resource cost. There is a population cap of 200 that players cannot exceed, and different units can contribute different amounts to this cap. In game, this is called supply, and as an example one Marine contributes 1 supply whereas a Siege Tank contributes 3 supply. Each unit behaves differently, for example some units can only attack flying units, some units deal bonus damage against organic units, some units are ranged. Additionally, there are three races that one can play as, Terran, Protoss, and Zerg, each with different units and play styles. How a player builds their army is one of the biggest determining factors for whether or not they might win a battle. For example, a line of Siege Tanks might do well against a group of Marines, but will get absolutely destroyed by just a single Liberator air unit.

One key aspect of the game is scouting out an enemy's army and building one's own army to counter that. The faster one can decide on a counter strategy, the better that player has a chance of winning. Our goal is to create a few different models that can determine optimal army composition with minimal resource cost and build time given some enemy army composition.

Research Questions

1. Given an enemy army composition, for each race what is the viable counter army with the least amount of build time?

2. Given an enemy army composition, for each race what is the cheapest viable army one can build to counter that enemy army?

Community Impact

This project will provide a new analytical tool that players may use to analyze games. For example, a player might look at past games to find times when they countered with a sub-optimal army composition in the middle of the game and found that they could have saved up more resources for a better army later in the game. By seeing exactly when and how a certain play went wrong, the player can learn from their past games and be better prepared for the future. This tool can also be used for players to develop their playbooks and keep a small reference sheet of good counters for common army compositions. By knowing ahead of time what sort of army one wants to build, a player can focus more of their attention on other aspects of the game and hopefully gain a competitive edge. Though it remains to be seen, if the tool is fast enough one might even be able to find optimal army compositions in the middle of a game.

Community Contacts: Our main contact within the StarCraft II community is a player who goes by the gamertag MrCurler, with additional input from the player Cairnes. Our community contacts were consulted to ensure that the output of our models will be useful and provided input as to what sort of information our models should include. For example, an early draft of our model included constraints for requiring the proper buildings to be built before being able to produce their respective units (i.e. one must build a Barracks before producing Marines), but MrCurler brought up the fact that in game, no one is going to factor in building costs when thinking about the total cost of their army.

Methods

There are two key components to our models, the combat simulator and the linear program. The combat simulator will be used to determine if a given counter army is viable or not. We will be implementing a simulator similar to the third approximation model found in the paper *Approximation Models of Combat in StarCraft 2*[1]. Even though it simplifies the gameplay significantly, this model is not very computationally intensive and was shown to be fairly accurate for simulating simple battles between two armies. We will create a function that takes in the enemy and counter army compositions, runs the combat simulation a bunch of times, and tells us what percentage our counter army was victorious and the average units remaining. We will use some cutoff to determine what we want as a "viable" counter army.

It might be that we want at least 10% of our army to survive 80% of the time, or we might require an average of at least 30% of our damage per second to survive 90% of the time.

The linear programs will be used to find our counter army compositions with one optimizing for resources used and another optimizing for production time. During development, we might come across some other constraints we wish to test, such as resources used over time needing to be a certain ratio of our total resource production (and resource production can be tweaked to approximate different stages of the game, so less production for early game and more production for late game), but for now our two main constraints are keeping the army within the supply cap and making sure our army is viable.

Outline of Linear Program (LP)

Let $Army$ be the list of the names of all of the units that can be produced for a given race and let $name \in Army$ be the name of a unit. Let $cost[name]$ be an integer variable representing the cost of that unit, $count[name]$ be an integer variable representing the number of units with that name, $time[name]$ be an integer variable representing the number of seconds it takes to produce that unit, and let $supply[name]$ be an integer variable representing the amount of supply required for that unit. Let $viable$ be a binary variable representing if our army is a viable counter or not ($viable$'s output will be determined by our combat simulator).

Our first research question is to minimize the total build time for our army, and so our objective function is to minimize

$$\sum_{name} time[name] \cdot count[name].$$

For our second research question, we wish to minimize the resource cost of our army and so our objective function is to minimize

$$\sum_{name} cost[name] \cdot count[name].$$

For both of these linear programs, our constraints are to keep our total supply below the cap of 200 and to make sure that our army is a viable counter to the given enemy army. These constraints are written as

$$\sum_{name} count[name] \cdot supply[name] \leq 200$$

$$viable \geq 1$$

$$viable \in \{0, 1\}$$

Output/Applications

After the execution of this linear program, we hope to provide optimal configurations for individual players to increase their probability of winning a game. Specifically, we will seek to optimize two distinct linear objectives. These are minimizing the total build time it would take a player to construct an army to fight in a game. This has practical application by allowing a player to construct as many teams as possible in the shortest period of time. This would allow for more battles and further allow for an increase number of battles. Thus, if a player is seeking to maximize the total number of battles to either win by repeated army combat, then such an output would provide the appropriate resource distribution such that the number of battles can be maximized.

However, if a player is constrained based on the number of resources, our second linear program will provide a means of creating an army cost efficiently. Specifically, this approach will assist the player in partitioning their resources most optimally for different battles.

With a combination of these two linear programs, a player will have the discretion to maximize the total build time for their armies in times of large conflict, and in other times produce the most cost-efficient army. An interplay between these two army constructions will provide a player with many opportunities to not only be successful in their battles, but also learn to most optimally preserve their resources. The output of both of these linear programs will be the exact resource use of all available resources to optimize their respective function.

Other References

We will be referencing the StarCraft II page on liquidpedia for all of the data on the stats and attributes for each unit. We will also be implementing the combat simulation models described in the paper *Approximation Models of Combat in StarCraft 2*, and all of our analysis will be made with reference to the limitiations of the combat models described in that paper.

Bibliography

[1] I. Helmke, D. Kreymer and K. Wiegand, *Approximation models of combat in starcraft 2*, 2014.