Generalizable Battle Simulator

*Background:*

A war is often seen as a game of choices (battles) made by 2 (or more) sides, although usually with far reaching and devastating consequences. Choices can be as simple as where to move or as complicated as how to synchronize multiple groups of troops converging on the same location. Either way, stronger tactics are key to triumphing in battle. We decided to create a model that describes and emulates the ability of effectively moving units on a battlefield.

*Model:*

In general, you can break down any battlefield into m number of zones, split based on proximity, potential tactical advantages, accessibility, ease of movement, etc. Similarly, you can break down any troop composition into n number of different troop types. Troop classifications can be based on style of combat (ranged, melee), mobility, armaments, armor, etc. Certain troops are going to have clear advantages over other troops. For example, a ranged unit has a clear advantage over a melee unit when the melee unit cannot reach the ranged unit.

Knowing all this, over some time step, every unit type () in zone () has some likelihood to kill another unit of type () in a zone (). This means unit damage can be described as a 4-dimensional parameter matrix (D) that maps the damage from units in zones of one side of a battle onto the units in zones on the other side of the battle. What is left is describing movement between zones for each unit type. This flow will be capped for each unit type () going from starting zone () to ending zone (). Therefore, a 3-dimensional parameter matrix (B) maps the maximum flow rate of different troops between different zones. This gives leaders on each side the ability to choose a normalized (-1 to 1) flow rate (a parameter matrix C with dimensions of B) from each zone to another for each of their unit types.

Moving zones is costly because it usually puts the unit moving in harm’s way. The traveling units have a survival rate (parameter matrix S with dimensions of B) capturing how likely they are to survive the trip from one zone to the next. Using this information, we created a model that would track troop numbers in each zone over discrete time steps using the following discrete differential equations:

Where X is the matrix (size n by m) for one side’s troops. B1, C1, and S1 are the B, C, and S matrices for the first side, and B2, C2, S2 are the B, C, and S matrices for the second side. Also, B.\*C is limited if there are not enough troops to move from the starting zone to the finishing zone. To generalize the model to more than 2 sides, additional equations can be added and the D matrix effect for the new variable should be added to all the other existing equations.

*Testing the Model:*

To thoroughly test this model, one would need tons of time series data for battles with enough troops that probabilities can accurately be described by estimated values. If such data existed, one could attempt to estimate/regress the parameters of the model using part of the data and run the simulation out through the rest of the time series data and compare the results to the data to check for consistency in the model. Since we could not find such extensive data, we decided to use our model to simulate a large battle from the Lord of the Rings Trilogy.

*Simulation:*

Helm’s Deep is a large valley in the north-western Ered Nimrais of Rohan, also known as the White Mountains. In the second Lord of the Rings movie (*The Two Towers*), the fortress of Helm’s Deep has become home to the main army of Rohirrim. The fortress has a very clear tactical advantage in that it can only be attacked from one direction, so any attackers are at a clear disadvantage. During the battle of Helm’s Deep, the Uruk-hai (the attacking army) have many strategies to siege the fortress. The protagonists of the film fend off waves of Uruk-hai until they breach the barricades, then the heroes retreat to the next level up and continue to fight from there. For our simulation, since each level acts the same after the retreat, we broke up the battlefield into 3 zones. The first zone is where the heroes fight from, the second zone is where the Uruk-hai are trying to scale walls/run up stairs/run down alleys to get to the first zone to attack the heroes, and the third zone is outside the fortress where the Uruk-hai wait before there is room to advance into the second zone. In the movie, Uruk-hai army is described as a monotonous force of around 10,000 melee fighters, while the protagonist force is a mixture of roughly “300 Men and a large group of Elves.” The Elves are archers and the Men are a mix of archers and melee soldiers. The protagonists’ goal is just to hold out until dawn, so Gandalf the White can show up and wipe out the remaining Uruk-hai.

*Results:*

*Model Limitations:* There are several limitations currently in our model. The biggest limitation is that all the effects are assumed to be first order effects. The model does not currently consider things like diminishing or increasing returns to scale that could result from certain combat phenomena (ex. Phalanx formation, limited area to shoot from, support units etc.). The model also currently does not consider the interaction effects of having friendly troops interact between zones (ex. Information sharing, covering fire, distraction, baiting, etc.). In addition, the model caps the flow of each unit crossing zones, when possibly a more accurate model might limit the number of overall units crossing each boundary, regardless of what type they are.

*Conclusion:*

*Further Development:* When developing this model further, the most important item to add would be nonlinear/cross zonal interaction terms. One could also consider creating stochastic implementations for the kill rate, so the model would be more realistic for low troop counts since the relative variance increases when the troop number shrinks. Another possible idea is to add a “wounded” effect where there is some chance that a troop is only wounded, but not killed, so they are less effective, but still alive and can fight.