FIT3139 – Final Project

Vignesh Guruparan

# Section 1 – Specification Table

|  |  |
| --- | --- |
| **Base model** | Guerilla Warfare |
| **Extension assumptions** | Stochastic Guerilla Warfare  A stochastic guerilla warfare model was created to reflect the non-deterministic nature of real-world conflicts, and to determine how it will affect the outcomes.  Assumptions made with this model were that the conflicts would occur in discrete time, and that for each time interval, exactly one fighter would be removed from the conflict. On top of this, all time increments were considered to be equal. |
| **Techniques showcased** | Markov chains: Stochastic outcomes in discrete time  Montecarlo simulation: Simulate several conflicts with similar parameters  Game Theory: Comparing multiple warfare strategies between each other to determine optimal tactics |
| **Modelling question 1** | What is the effect of stochasticity on conflict outcomes? |
| **Modelling question 2** | How can allocation of troops affect conflict over several connected battles? |

# Section 2 – Introduction

In Guerilla warfare, it is likely that no two conflicts will be the same, with several factors, including variance between fighters’ ability on the same team and environmental impacts. Therefore, it is worthwhile to consider a level of stochasticity in each individual conflict, the effects of which will be explored using Markov Chains and Monte Carlo Simulation.

On top of this, Guerilla warfare conflicts mostly occur in a set of connected events, rather than a singular “battle”. Therefore, the way that both the defending militia and guerrilla troops allocate their numbers across several battles of a “war” can have major impacts on the winning outcomes of either side. A “game theory” application will be used to determine what strategy will be the most beneficial for each side, given a certain set of initial conditions and parameters.

# Section 3 – Model Description

The base model of Guerilla warfare, as shown in Equations 1and 2, depict the manner of which the defending team (), and the offensive team () decrease over a single conflict. The base model is numerical, non-linear, deterministic and **in discrete time** (to match that of the stochastic model).

An assumption that the base model makes is that there are no external factors that affect the parameters of the conflict after it begins, such as reinforcements or changes in environment. On top of this, the values at each time interval will not be rounded, due to the fact that rounding will likely result in troops either being over or under-estimated, affecting the reliability of results. This choice does come with drawbacks, the main one being that the accuracy of the model will somewhat decrease, as in real-world scenarios, it is not possible for there to be a decimal amount of people fighting for either team. This effect could be explained by decimal numbers representing a fraction of a soldier’s effectiveness, but for simplicity’s sake, this will not be the case in this model. To ensure that conflicts do not last forever as a result of this, if a value of x or y goes below 0.1, it will be set to 0, effectively ending the conflict.

An example of the base model can be shown in Figure 1 below, with initial conditions and parameters as stated in the title.

A graph of a number of different colored lines

Description automatically generated

Figure 1: Plot of Guerilla warfare base model

The updated stochastic model uses a Markov chain, with its transition matrix structure as shown in Figure 2 below:

A paper with writing on it

Description automatically generated

Figure 2: Schematic of Markov Chain transition matrix

The transition matrix is a square matrix, where each row represents a possible combination of defending and offensive troops, spanning from to . In this model, exactly one troop from either side will be removed from the conflict after each time increment. The rows of the matrix represent the current states, and the vertical states represent the possible states that the system can move to, with entries being the probabilities of moving to the “column” state from the respective “row” state. As stated previously, from any state, only one defending or offensive troop can be removed from the conflict, with probabilities shown below in Equations 3 and 4:

In the equations above, the probabilities are based on the two equations depicting the rates of change of x and y (Equations 1 and 2). The values of and are that of the current state of the system. States where either the value of or are zero are considered absorbing states, meaning that the system will perpetually remain at that state, signalling the end of the conflict.

This model results in the most simplistic depiction of Guerilla warfare but will result in some differences between this model and the base model, the most visible one of which is that the conflict time will generally be longer the stochastic model, in comparison to the base model. This could’ve been avoided by including transitions that either removed multiple troops of one side or removed troops from both sides in a single interval. This strategy would be difficult to implement, as it is likely that new parameters would have to be introduced to define the probabilities of either outcome, and they would have to interact with the existing parameters, greatly complicating the model.

Another difference between the two models is that the graph of the stochastic simulations do not “flatten out” towards the end of the conflict, as there is no possibility to remain in a state that has an ongoing conflict. The likely solution to this shortcoming is to have a probability to remain at the same state, potentially increasing it as time progresses. Once again, this was not included into the model due to its complexity, and the fact that the stochastic models already had longer conflict times than the base model.

# Section 4 – Results

What is the effect of stochasticity on conflict outcomes?

To compare the base model with the new, stochastic model, two Markov chain simulations were run with the same parameters and initial conditions as those of the graph in Figure 1. As shown in Figure 3 below, the stochastic element of the Markov chain introduces a high level of variance for this conflict, where in the base model, it appears as though the conflict is a straight-forward victory for the defending team.

A graph of a number of people

Description automatically generated with medium confidence

Figure 3: Comparison of base model and stochastic model simulations

The first main observation to be made between the two models is that it is possible for either team to win with these parameters, with the offensive team winning approximately ~50% of the time (over 5000 simulations), contrary to what is described by the deterministic model. The second observation is that the average conflict length for the stochastic model (12 days) is far greater than that of the deterministic model (~75 days). The likely cause of this, as stated earlier, is the fact that only one agent is removed from the conflict each day, extending conflict lengths depending on the initial number of agents. A graph with the conflict lengths of 100 simulations can is shown in Figure 4, below.

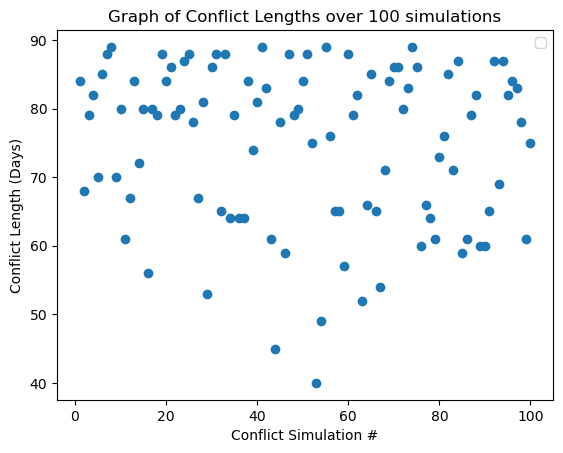


Figure 4: Plot of conflict times over 100 stochastic simulations

The average conflict length for the defending team’s wins was ~65 days, while the average length for offensive team wins was ~85 days. The main cause of this deviation is the fact that the offensive team has to remove more troops from the conflict than the defending team, where both teams can only remove one opponent at a time (although the probabilities at which this occurs will likely not be the same for both teams).

This “single removal” approach is also the reason why the model may not be appropriate for describing conflicts with a large number of troops in both sides, as the size of the transition matrix representing the conflict is bounded by , where an average computer is not able to allocate such space for the transition matrix. A potential workaround to allow simulation of larger conflicts is to use a more space-efficient structure, as the one used currently is a numpy array, with each element being another numpy array, to represent a 2D matrix.

How can allocation of troops affect conflict over several connected battles?

For the purposes of analysis, a “war” will be considered a set of connected conflicts, where both sides have a number of troops that they can allocate to the battles however they wish. Due to constraints surrounding processing time, there were two strategies explore, which each side had the option of implementing. The first strategy was to “evenly load” the troops, meaning that each battle would have the same number of starting troops. The second strategy was to “back load” the troops, which involved allocating only one troop to each of the first five battles, and evenly distributing the rest of the troops over the remaining battles. This strategy aims to “tank” the battle with the hopes of gaining and advantage for the rest of the war. In this investigation, it will be assumed that neither side is aware of the strategy the other is electing to implement.

To determine the effectiveness of these strategies in conjunction with each other, 1000 wars consisting of 40 battles each were conducted for each combination of strategies. The parameters of the simulation were and , with initial conditions of and , where represent the total number of troops for the defending side and offensive side respectively, at the beginning of the war. In this simulation, to “win” a war, the side must win most of the battles, and in the case of a draw (20 wins each), the win will be awarded to the offensive team. A payout table can be created using the overall war win percentages for each side, corresponding to the different strategy combinations, as shown below in Table 1.

Note: As the percentages present in Table 1 are derived as a result of 1000 simulations each, there may be slight uncertainties in the values, but they provide a relatively accurate representation of war outcomes.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Offensive Team | |
|  |  | Evenly Loaded | Back Loaded |
| Defending Team | Evenly Loaded | (32.6,67.4) | (21.1,78.9) |
| Back Loaded | (65.7,34.3) | (49.8,50.2) |

Table 1: Payout table for all strategy combinations

Using the “nashpy” library in python, the normal form game defined by Table 1 can be solved for, revealing that the optimal strategy for both teams is to “back load” their troops. By observing the payout table, it is understood that back loading is a dominant strategy for the defending team, as its payouts when the offensive team either evenly loads or back loads is strictly better than if they were to evenly load. Therefore, as the defending team is guaranteed to back load, to ensure the most optimal payout the offensive team must also back load (back loading is also a dominant strategy for the offensive team).

This combination of strategies results in an extreme likelihood that the offensive team wins the first battle, as both sides only have one troop each, and the value of is significantly higher than that of (see Equations 3 and 4). In this scenario, there are now defending troops and offensive troops to be evenly dispersed across battles, with an average of and of . Even though this is a relatively small increase relative to the initial conditions from evenly loading ( and ), it evidently has great impacts on battle, and consequently, war outcomes. This is compounded on top of the fact that the defending team is at a great advantage in the rare occurrences when they are able to win the first battle. Both of these impacts result in a massive improvement in war outcomes for the defending team, jumping from a win percentage when both teams adopt the standard strategy, to a win percentage when both teams play to their optimal strategy of the two options.

Seen below, in Figure 5, is two graphs of the remaining soldiers after each battle in 2 separate wars, the first where both teams backload, and the second where both teams evenly load.

A graph of war and war

Description automatically generated

Figure 5: Graph of Remaining Troops after each Battle over 2 Wars

As the results present in Figure 5 are only over one war each, there is a relatively large level of noise in the values recorded. However, simply observing trends in the two graphs, when both teams backload, the defending team tends to lose the first battle, then is able to make up for it by increasing their win percentage over the rest of the war. When both teams evenly load, there isn’t an observable trend in teams winning rate and the remaining troops after each conflict, resulting in the offensive team having a tangible advantage over the course of the war, due to the initial conditions and parameters.

Note: This nash equilibria only strictly applies for the specific set of initial conditions and parameters, it may vary if those are changed.

In a real-world scenario, it is extremely likely that an army would employ the backloading strategy, as it isn’t feasible to have a battle where they send only one troop, with the means to do otherwise. To expand this investigation, more realistic strategies can be applied, such as linearly increasing the allocation of troops between a minimum and maximum value, or randomly allocating a number within a certain range. This will greatly increase the processing time to accurately determine the effectiveness of each strategy in combination of each other, as the time complexity is bound by O(), where and are the number of strategies the defending offensive teams have available to them respectively.

# Section 5 – List of Algorithms and Concepts

* Guerilla Warfare model
  + Base model
* Markov chain simulation
  + To represent/simulate Guerilla Warfare stochastically
* Monte Carlo simulation
  + To determine effect of stochasticity on Guerilla Warfare model
  + To determine effectiveness of opposing Guerilla Warfare strategies
* Dominance Theory to find Nash Equilibria
  + To find optimal strategy for either team to employ