

# Mathematical Model for Item Build Maximization of League of Legends

Jon-Michael Rutter, Max Highsmith

December 8, 2016

**Abstract**

## 1 Introduction

League of Legends is the massively popular game developed by Riot Games. It is a Multiplayer Online Battle Arena (MOBA) style game where two teams of five players each face off against each other in a computer generated environment. Although the goal of the game is simple, the wealth of characters, items, abilities, strategies, and sheer player-base make each game a drastically different experience.

League of Legends currently boasts 100 million active players. Last year, over the five weeks of World Championship play, over 334 million viewers tuned in, and 36 million watched the final match between SK Telecom Team 1 and the Koo Tigers.

In League of Legends, there are five positions on each team: Top Lane, Jungle, Mid Lane, Attack Damage Carry, and Support. While compositions and roles vary, there is one role dedicated to a specific class: the Attack Damage Carry, who is a ranged Marksman type character. Typically, these characters rely on their auto-attacks to inflict massive damage to the enemy team. Players have a choice between 21 different characters with substantially different play styles to fill this role. Through a game of league of legends players accumulate gold and purchase items which contribute to the players ability to complete objectives during the game. Different items allow for different advantages and disadvantages. This project aims to construct a model for selecting the optimal combination of items for a champion filling the role of Attack Damadge Carry.

## 2 Background Terms

There are a variety of terms used to describe elements of the game which are essential to understanding this model.

Term	Definition
Player	The human being who is playing League of Legends
Champion	An in game character controlled by the player.
ADC	Attack Damage Carry, sometimes AD Carry. This champion is responsible for inflicting physical damage to enemy champions.
Item	an entity which a champion can be equipped with to provide some sort of improvement
Shop	Location where gold is exchanged for items
Build	a combination of multiple items
Caitlyn	An ADC champion, this is the champion whose build this model aims to optimize
Sivir	An ADC champion, an opponent champion used in 1v1 metrics
Quinn	An ADC champion, an opponent champion used in 1v1 metrics

### 3 Variables and equations

#### 3.1 Primary variables quick reference

L	Level
G	Gold
B	Build
N	Natural Stats
A	Adjusted Stats
R	Attack Array
DPS	Damadge per Second
IFS	individual fight score
NDPS	Normalized DPS
S	1v1 score vs Sivr
Q	1v1 score vs Quin
Z	Build Score
I	item
i	index of shop
c	champion type
$A_Q$	adjusted stat matrix for champion Quinn
$A_S$	adjusted stat matrix for champion Sivr

#### 3.2 Equations quick reference

$$B = f(H, i) = (I_1, I_2, I_3, I_4, I_5)$$

$$N = f(L, c)$$

$$A = f(B, N)$$

$$R = f(B, A)$$

$$DPS = f(A)$$

$$NDPS = f(DPS, A, G)$$

$$IFS = f(A_1, A_2)$$

$$S = IFS(A, A_S)$$

$$Q = IFS(A, A_Q)$$

$$Z = f(D, S, Q, B)$$

## 4 Variable in Depth

### 4.1 Gold

Gold is the currency used by players in league of legends. Each player has a reserve of Gold which can be paid to the shop in exchange for equipping items to a players champion. Throughout the game gold grows at a non linear rate dependent upon completion of in game objectives. Gold is a positive integer value.

### 4.2 Level

Champions in league of legends grow levels as time progresses in the game. Each time a champion gains a level they become more powerful. A champions level,  $L$ , is a discrete integer s.t.  $1 \leq L \leq 18$

### 4.3 Natural Stat

We define Natural Stat as a vector whose values are baseline stats which a champion has prior to augmentation with item purchases. These values are determined based on the champion type and level.

$$NaturalStat = \begin{pmatrix} Health(Level) \\ HealthRegen \\ MagicResist \\ AttackDamadge \\ AttackSpeed \\ CritStrike \\ LifeSteal \\ AttackSpeedOffset \\ Level \end{pmatrix} = f(Level, ChampionType) \quad (1)$$

This describes the internal mathematical mechanism for generating the components of the Natural Stat vector. These details may be skipped the first read through. FIX: internal mechanism of generating values

### 4.4 Item

Item is a variable which represents an item in the game shop. Each item variable provides information on a bonus which a character will recieve if they posses said item. The nature of the bonuses can be baseline addition, percent increase, or otherwise specified alteration.

Examples:

Ruby Crystal	+150 baseline health
Dagger	+12% attack speed

## 4.5 Build

We define a Build,  $B$ , as a combination of items which a character has selected.  $B$  is dependent upon gold and item index. There are a finite number of combinations of items which a player may purchase based on the amount of gold that they currently possess. The index variable is used to identify which of the potential combinations of items  $B$  represents. The  $item_i$  fields can be 0 to signify a lack of an item in that slot

$$Build = \begin{vmatrix} item1 \\ item2 \\ item3 \\ item4 \\ item5 \end{vmatrix} = f(G, i) \quad (2)$$

Each  $item_i$  is an item variable. It is worth noting that in league of legends there are 7 item slots, but one is reserved for wards and one is typically used for mobility, neither of which factor into our model.

## 4.6 Adjusted Stats

The Adjusted Stat matrix,  $A$ , is a vector whose values represent the statistics of a character after it has been equipped with a combination of items. The Adjusted Stat matrix contains the same fundamental fields as a natural stat matrix (with the exception of `AttackSpeedOffset`) but alters the fields according to the bonuses assigned to each  $item_i$  within build  $B$ .

$$AdjustedStats = \begin{vmatrix} Health(Level) \\ HealthRegen \\ MagicResist \\ AttackDamage \\ AttackSpeed \\ CritStrike \\ LifeSteal \\ Level \end{vmatrix} = f(B, NaturalStats) \quad (3)$$

## 4.7 Attack Array

The Attack Array is a 2 dimensional array which represents the ordered sequence of maneuvers a champion would optimally used in battle. Attack Array is generated by an Adjusted Stats matrix variable and Build variable. The Adjusted Stats matrix variable is needed to evaluate the strength of auto attacks as well as the scaling bonus applied to champion spells while the Build variable is used to account for any items with active abilities. To understand the structure of the Attack Array think of each column as representing information about an individual maneuver which a character might take during a given battle. The row elements represent information about the columns' maneuver and breaks down into 4 fundamental categories. The construction is as follows.

0	Move1	Move2	...	Move 15
Time				
Attacks Medium				
Reductions				
Type of Attack				

### The Categories of Rows

**Time** 1 row:

the amount of time it takes to perform maneuver

**Attack Medium** 3 rows:

Physical amount of physical damage dealt by maneuver

Magic amount of magic damage dealt by maneuver

True amount of true damage dealt by maneuver

**Reductions** 8 rows:

Flat Armor reduction

Percent armor reduction

**Types of Attack:** 1 row:

2 possible values: auto, spell

The remaining variables  $D, S, Q, Z$  are variables which are used as metrics for builds, so their explanation is delayed until the metrics sections

## 5 Metrics

The goal of a player is to ultimately assist his team in destroying the enemy Nexus. There are a variety of smaller objectives which are essential to achieving this goal. For the sake of simplicity this model considers battle based objectives and strives to measure the effectiveness of a given character from a battle standpoint. We do this by providing some variables which represent a champions ability to succeed in various battle environments.

### 5.1 Instantaneous Danger

Our Metrics for evaluating the efficiency of a given build B on a the champion include

- DPS= DamadgePerSecond = The amount of damadge a champion can distribute per second
- IFS= IndividualFightScore = the likelihood of the champion equipped with build B of defeating an opponent O

The IndividualFightScore is applied to two opponent champions, Quinn and Sivr. The Builds provided to Quin and Sivr are based on the league of legends community meta rather than mathematical optimization. This allows the generic IndividualFightScore metric to provide two scores, the likelihood of beating a Quinn with comparable gold, and the likelihood of defeating a Sivr with comparable gold. We

To provide a comprehensive metric for Caityln builds we will combine the three metrics (IFS(Quinn), IFS(Sivr), DPS) into one index meant to represent a champion's danger to the enemy team at a given time.

$$Danger = f(DPS, IFS(Quinn), IFS(Sivr)) \quad (4)$$

We now describe our damage metrics in greater detail

### 5.2 DamadgePerSecond

The DamadgePerSecond metric is intended as a metric which is independent of the composition of an opposing teams roster. As its name implies, DPS represents the rate at which caitlyn can dispel damadge when equipped with a particular build. To ensure that this metric factors in item induced bonuses like magic and armor penetration we construct a dummy target with standard armor and magic resistance for her to target. Using the damadge calculation formula specified by riot We compute the total amount of damadge she would dispel over 15 maneuvers and divide it by the amount of time required to complete all such maneuvers.

$$DPS = \frac{TotalDamadgedistributedoverT}{Totaltime} \quad (5)$$

For the purpose of combining our metrics into a composite damage metric we want the output of the DPS function to fall within the interval  $[-1, 1]$ , where positive values indicate a good score and negative values a bad score. To do this we introduce a variable  $DPS_{avg}$  which is the average DPS of all AttackArrays generated from Builds of cost  $g$ .

$$NDPS = \frac{NDPS - DPS_{avg}}{DPS_{avg}} \quad (6)$$

### 5.3 IndividualFightScore

To evaluate the effectiveness of a build, this metric models a champions ability to win fights against a variety of potential opponents. Before rigidly defining the internal mechanism of the *individualFightScore* variable it is useful to describe the goal behind this metric as well as the intuitive interpretations of the variables inputs and outputs.

#### Input

We require a mechanism for numerically evaluating a champions ability to defeat an enemy champion in 1v1 combat. Therefore the primary inputs we would expect are the CharacterStats matrix for the two fightingl champions.

$$individualBattle = F(ChampionAStats, ChampionBStats) \quad (7)$$

#### Output

In a league of legends battle between characters A and B there are fundamentally 3 potential outcomes from the perspective of champion A.

##### 1. Champion B dies and champion A survives.

This is considered a victory by Champion A and should contribute to a build being categorized as effectiveness

##### 2. Champion A dies and Champion B survives

This is considered a loss by Champion A and should contribute to a build being categorized as ineffective

##### 3. Champion A dies and Champion B dies

This outcome will be referred to as a draw. Within this model a draw is considered neutral and should not contribute to a build being viewed as more or less effective.

We also note that while there are only 3 fundamental outcomes to a 1v1 battle, not all victories and defeats are equal. Because of this we wish to construct the individualBattle variable so that we are given a numerical representation for how severely champion A won or lost. Note that this metric must be standardized so that it can be compared with other metrics effectively, thus we must place consideration on the scale and bound of the value. Keeping these factors in mind we construct the potential outputs as follows



**1 . Champion A Victory:**

a positive number which is bounded by 1. The higher the number the more severe the victory

$$\textit{individualBattle} = (\textit{RemHealthChampionA} / \textit{StartHealthChampionA}),$$

$$0 < \textit{individualBattle} \leq 1,$$

**2. Champion A Defeat:**

a negative number which is bounded by -1. The lower the number the more severe the loss:

$$\textit{individualBattle} = (\textit{RemHealthChampionB} / \textit{StartHealthChampionB}) * -1,$$

$$-1 \leq \textit{individualBattle} < 0$$

**3. Champion A Draw:**

$$\textit{individualBattle} = 0$$

*BattleSequence*

## 6 Code Summary

## 7 Results

Our Model tested for

## 8 Potential Areas for Expansion

Some of our potential areas for expansion include incorporating items which buff character utility when factoring instantaneous danger and overall effectiveness. Potential variables include: warding buffs, jungle damage, healing allies, minion buffs, tower buffs, and dragon buffs.

## 9 Works Cited