Improved Balance in Multiplayer Online Battle Arena Games

by

Chailong Huang

A thesis submitted to the
Department of Computer Science
in conformity with the requirements for
the degree of Master of Science

Bishop's University
Sherbrooke, Quebec, Canada
March 2018

Abstract

The Multiplayer Online Battle Arena (MOBA) game is a popular type for its competition between players. Due to the high complexity, balance is the most important factor to secure a fair competitive environment. The common way to achieve dynamic data balance is by constant updates. The traditional method of finding unbalanced factors in a MOBA game like DOTA2 is mostly based on professional tournaments, which is a small minority of all the games and not real-time. We develop in this thesis an evaluation system for the DOTA2 based on big data with clustering analysis, neural networks, and a small-scale data collection as a sample. We then provide an ideal matching system based on the Elo rating system and an evaluation system to encourage players to try more different heroes for a diversified game environment and more data supply, which makes for a virtuous circle within the evaluation system.

Directory

Chapter 1

Introduction

The explosive development of the Internet has brought great convenience to people, and also made leisure and entertainment style much more diversified. Electronic games [1], especially e-sports games, have become an important part of people's lives. With the increasingly diversified social life and the gradual emancipation of people's minds, electronic games are becoming more and more mainstream, especially among young people. The game industry has also flourished, more and more excellent games are coming out. As a common way of entertainment in daily life, the electronic game is a good help for people to eliminate fatigue and reduce the mental pressure of studying and working. Some healthy games can even help children grow up happily in an educational way [2]. Research shows that games have a good effect on the development of the brain, especially for minors.

An electronic game is software, which needs hardware equipment with certain performance to provide the operating environment. With the substantial increase in hardware performance and people’s desire to pursue better entertainment, various types of high-quality games are coming to people’s daily life. Multiplayer Online Battle Arena (MOBA) games [3] are very popular among young people because of their interesting and playful features. The reason why there are so many MOBA game lovers is the competition in all kinds of areas such as operation, strategy, teamwork. The basis of this competitive pleasure lies in its balance.

DOTA (Defense of the Ancient) is the first independent MOBA game. As a classic game with a 15-year history, it’s still very popular as the second generation DOTA2. Witnessing its enduring popularity, more and more companies have seen the business opportunities, so a batch of similar MOBA games came out, *“League Of Legends”*
from Riot Games, *“Heroes of the Storm”* from the famous Blizzard Entertainment, *“Heroes of Newerth”* from S2 Games are all excellent MOBA games. Even though they are not as balanced as DOTA, they still have a large number of players.

Success and failure always exist simultaneously, so not all games are as excellent as the ones mentioned above. In China there are even more MOBA games, but most of them are commonly criticized for their low quality, unbalanced setting, even plagiarism. Game design is actually a process of producing artwork: beside creativity and inspiration, exquisite workmanship is also required. Balance is the workmanship of games. It is even more important in a competitive MOBA game. An imbalanced game cannot guarantee the loyalty of the fans; players will get bored easily if they only have a few options to win a game. Thus a deep understanding of the balance of the game as well as the way to achieve it are both necessary for a MOBA game. With the analysis of MOBA game and data balance, this thesis develops a new method to achieve a better implementation of dynamic balance in DOTA2, and then an ideal matching system is designed as an improvement over the original one.

In this thesis Chapter 2 focused on the overview of an emerging game type called MOBA (Multiplayer Online Battle Arena), with the representative work DOTA2 (by Valve Corporation). In Chapter 3 we show that because of MOBA game's complexity, balance is the most important purpose for designers from the design process (static balance) to an on-going concern through new patches (dynamic balance). Most companies like Valve are trying to evaluate the imbalanced factors based on players reports and game situations in professional tournaments. The former only reflects the most prominent problems, the later only include professional players as sample, ignoring thousands million ordinary players around the world. In Chapter 4 preliminaries about DOTA2 rank/ matching system, K-Means cluster analysis and BP-Neural Network are introduced for further research. The main contribution of this thesis is in Chapter 5, where a new method is developed to achieve a better implementation of dynamic balance, with small-scale data collection as a sample. As a multiplayer online game, a fair matching system also plays an important role in game balance. Based on DOTA2 original rank/matching system (which uses the same rank system based on Elo ratings as *"League of Legends"*), we design an ideal algorithm for the matching system. Finally we come up with an improvement on the rank/matching system based on the analysis of dynamic data balance. Finally potential problem and further research are discussed in Chapter 6.

Chapter 2

MOBA Games Overview

MOBA is the abbreviation of Mutiplayer Online Battle Arena games, also called Action Real-time Strategy Games (Action RTS, ARTS), or DOTA-like Games, which is a subclass of Real-time Strategy Games.

In a MOBA game, there are no buildings, population, resources, arms training, technology tree, or unit organization, which are all common in traditional RTS games. Players in MOBA games are usually divided into two camps, five versus five by fighting for more gold to buy items, and experience to level up, which can all strengthen the players' units. The ultimate goal is to destroy a certain building of the other side. Unlike most traditional hard-core RTS games such as *"Warcraft III: The Frozen Throne"* and *"Star Craft"*, MOBA game player usually controls one character only called "hero", which has specific abilities and slots to equip with items.

Compared with traditional RTS game, MOBA games are more entertaining, so the difficulty of the operation is designed to be acceptable for most of the average players. Not deliberately requiring any high level of operation and mental strength makes MOBA games more universal, and easier to start. Beside, most MOBA games are free games, which is another low threshold for a large number of players.

2.1 Development Path of MOBA Games

Origin

In 1998, Blizzard Entertainment released the classic RTS game *"Star Craft"*, which is the first time that Blizzard bound a map editor with a game, which allows all the players to edit their own map. Among all custom maps, the one named "Aeon of Strife" from an unknown player became originator of all MOBA games.

Development

In 2002, "the conscience of gaming industry" Blizzard Entertainment again released the sequel of *"Warcraft"* series *"Warcraft III: Reign of Chaos"*. This legendary game soon became the hottest topic for every player around the world. As before, a map editor was released with the game itself, which threw players once again in the madness of designing custom maps. In a short time, a group of map designer got together and planted the seed of DOTA, the milestone of MOBA games.

The warcraft-version *"Aeon of Strife"* first came out, followed by *"Valley of Dissent"* and *"Defense of the Ancient"*. Finally, *"DOTA All-Star"* became the ultimate DOTA map, with continuous updates to fix bugs and adjust balance. Since then, MOBA evolved into a real game type.

Prosperity

The fantastic gameplay attracted countless crazy players in a very short time. Tens of unique heroes, various items, fine operability, imaginative tactics, and of course the constantly improved balance, all these advantages allowed DOTA to sweep the world in a few months. Asia, Europe, South and North America were suddenly full of DOTA players and lovers. While at that time "*Warcraft III"* including DOTA could only be played on local area network, LAN Battle platform became another business opportunity. With the heat of DOTA, many battle platforms appeared, such as GG (Good Game) and VS.

More and more players gave up on all kind of custom maps such as 3C, RPG (Role-playing game) maps and even Warcraft III itself, to became DOTA fans. It was also at that time that the huge number of players attracted other game companies, which tried to share this MOBA cake. As a result, some excellent MOBA game like *"Smite"* and *"League Of Legends"* came out, with great success and huge profit. Based on these success precedents, even more MOBA games kept coming out.

2.2 Representative of MOBA Games

DOTA2

Since DOTA came out, countless imitations followed the trend. Of course, most of them got their own features, in one way or another. Though DOTA is classic and balanced, it is just a custom map based on the *Warcarft III* game engine [5] after all, which is a game from 2002. With the rapid development of information technology, old game engines are destined to be eliminated because of

their shortcoming and limitations: There is a 8MB limit for *Warcarft III* custom maps, which means that one day inevitably no more new elements could be added into DOTA, including new heroes, abilities and items. Low image quality was also intolerable to players after 10 years. Last but no least, some bugs based on the game engine itself can never be fixed in DOTA. All these problems have restricted the further development of DOTA. At the same time, new MOBA games such as *"Smite"* and *"League Of Legends"* competed for DOTA players based on their own engines and good publicity, and so kept more and more potential players away from DOTA. This phenomenon seemed to suggest that DOTA was meant to gradually exit the stage of history. In 2008 Icefrog, "the father of DOTA" who kept updating DOTA without any benefits for years, finally decided to take a new step in DOTA's life, that is, make DOTA2.

In 2009, Icefrog and Valve hit an agreement, starting the development and research work on DOTA2. After two years, DOTA2 came out, and immediately won the praise of players around the world. Based on Source Engine [14], DOTA2 fixed most bugs that would never be fixed in DOTA, with brand new features like skins, watch, matching, voice chat, community, and reports. Since then, DOTA2 literally became a individual game, detached from Warcarft III.

To this day, DOTA2 still has a large number of players, with a 1.2 million highest number of online players on Steam (the platform of Valve). Also DOTA2 is the leader on professional e-sports game. In 2017, the International 7 was held in Seattle, U.S. with a more than 24 million dollar prize pool. Players who left DOTA returned to DOTA2, which again shows the classic nature and balance of this game. As a high-quality MOBA game, with remarkable gameplay and balance, this thesis will take DOTA2 as an analysis template.

Other MOBA games

"League Of Legends" or LOL was developed by Riot Games, which has been acquired by the Chinese Internet giant Tencent. Besides the classic MOBA elements from DOTA, LOL also has its own features, such as runes, talent tree, and of course, brand new heroes.

Looking around the world, LOL has much more players than DOTA2, mainly because of Riot's wise development strategy that released the game with vigorous publicity in 2010-2011, during the period that players lost by DOTA have not joined the DOTA2 camp. Also, LOL is even more entertaining than DOTA2, with lower difficulty to get started. The bright color cartoon style attracted more young players as well. Objectively speaking, in game balance, LOL didn't do as good a job as

DOTA2 did. For example, in professional games the heroes played rate (the number of heroes that have been picked in the whole tournament versus the total number of heroes) of LOL is much lower than DOTA2, that is, so many heroes in LOL are too weak to be picked and played in professional games, which means that the designs of these heroes is not balanced.

"Heroes of Newerth" (HON) was developed by S2 Games. It inherited DOTA's traditional play style, and has a great number of fans in Europe and America with excellent reputation.

"Smite" was developed by Hi-Rez Studios. This 3D MOBA game is based on the Unreal 3 game engine. Under the epic background of the war of gods, *"Smite"* combines for the first time MOBA with a FPS (First Person Shooter) game. Therefore, it has a higher requirement for players' personal operating skills and teamwork.

Drawing on *"Smite"*'s success, Blizzard Entertainment developed and published *"Overwatch"* in 2016. With better game engine and images, *"Overwatch"* caused a huge response, and won the "Game of the Year" at the 20th annual DICE Awards (by the Academy Of Interactive Arts & Sciences)

There are so many successful MOBA games. The above are just some representative cases. In general, these games all came from the idea of DOTA, combined with their own creativity such as FPS, and history or mythology background. All in all, the basis of MOBA games is the balance, which of course, got the most successful implementation in the DOTA series.

2.3 Related Properties of DOTA2

DOTA2 is a very complicated game. This thesis will only introduce the elements that are related to our research, while omitting irrelevant factors.

Introduction to the map and the way to win

In general, every DOTA2 game has 10 players, divided into two camps. Every player needs to pick a hero to control at the beginning of the game, doing a 5 versus 5 confrontation. The two camps are respectively called Radiant and Dire with one ancient structure in each of their bases. The bases are located on each side's high ground, with three lanes to the other side. For each side, there are two barracks (melee and range) and tree defense towers on each lane. Barracks produce three types of "creeps" (melee, range and siege)every 30 seconds who automatically attack all the enemy units

along the lane. Defense towers automatically attack enemies whenever within their range. The strength and number of creeps grow over time. Each side also has two jungles, with some neutral creeps in them. Players control their heroes to kill creeps and enemy heroes to gain gold and experience. Gold can be used to build items strengthening heroes. Experience is for leveling up, heroes can get one spell point each level for one of their four abilities (three basic abilities and one ultimate ability which can only be gained on certain levels). The only way to win is to destroy enemy's ancient structure right at the very center of their base. Figure 2.1 shows the basic structure of the DOTA2 map.

Figure 2.1 DOTA2 mini map

Units and items

113 different *heroes* (Figure 2.2) can be picked by players. Each hero is one of the following three types: Strength, Agility and Intelligence based on their major attributes. Beside these three types, there are more attributes combined on every hero (Figure 2.3) including: Armor, Move speed, Attack speed, Magic resistance, Health/Mana points, Damage, and four abilities with different cool down and damage types, etc.

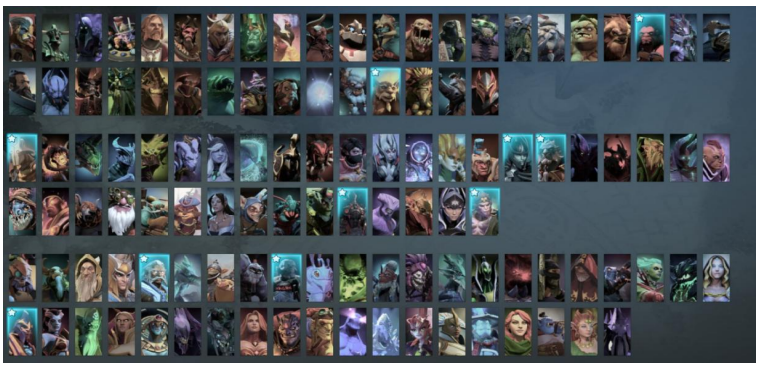

Figure 2.2 Hero Profiles

Figure 2.3 Attributes and Abilities interface

Each side has three types of *creeps*: Melee, Range and Siege. Once barracks are pulled down, the enemy's corresponding lane will produce super creeps. If all the six barracks are down, enemy's creeps become mega creeps. Radiant and Dire also have 9 jungle camps each side with neutral creeps (2 of them are ancient neutral creeps), which will refresh every minute if they are killed by players. Some heroes have abilities to summon specific creeps to fight for them.

Items can be bought and built in each side bases' Spring (the resurrection place which can also provide recovery for heroes), the Neutral Shop, and the Side lane shop. Items can be divided into Recovery products (used to restore heroes' health point and mana points), Wards (to provide some vision for a certain time) and other items which can provide some addition on their attributes and specific abilities.

Most heroes have 4 *abilities*: 3 ordinary and 1 ultimate ability. Each ordinary ability can be leveled up with a maximum level 4. The ultimate ability can be leveled up only at hero's level 6, 12 and 18. Every ability has a different effect such as dealing damage, stun, slow the enemy units, or provide beneficial status for the hero and its allies. The Talent tree was also added into DOTA2 several patches ago. Each hero can choose their talent at level 10, 15, 20 and 25 to give themselves a unique status or ability bonus.

Chapter 3

Game Balance Overview

Definition of Game Balance

"A game is a series of meaningful choices." ----Sid Meier

This quote reveals the nature of game balance, that is, every player is supposed to have multiple choices to achieve their goals. Back to MOBA games, the goal is to win the game in the competition. Since there are more than a hundred heroes and items in the game, countless choices like team composition, item choice and combat strategy are made by every player every minute everywhere. Some of them are wise and good, which can lead players to a final victory, others are not. But if only one choice is correct at every crossroad, the game will soon become a meaningless repeat, and so winning the competition becomes gradually boring. That is imbalance. A balanced MOBA game is supposed to have bad choices, but there are always several good choices every time, and different sets of options will eventually produce different results. In other words, in a balanced MOBA game you can try everything to win, unlike a robot who follows the same rules all the time.

We then define the game balance like this: Under certain conditions, there are plenty of options for the player to choose from. All game imbalance can be summed up as too few options to choose.

The Importance of Game Balance

The fundamental purpose for a player to play a game is to gain pleasure, so the playability of a game decides how long its life will be. For MOBA games, balance determines the playability, and it is even more important than for other types of games, for indeed nobody would stare at ability effects or beautiful scenery, or heroes' panel with many numbers for a long time. Competition is the soul of MOBA games; the intelligence and courage contest determines how much pleasure can every player gain from a game. All these are based on balance. Without balance, it doesn't even matter if you have more experience or better skill, the one who grabs the "perfect choice" always win the game. This

choice could be an unbalanced hero, or a bug in the matching system. So for every game designer and developer, balance is always the most important factor from beginning to the end.

Basic Idea to Achieve Balance in MOBA Games

"The Battle Arena" reveals the competitive attribute, which means that a large amount of data support is required to achieve the fairness of competition. Since it is mutiplayer, different levels of skills and experience need to be supported as well. A matching system should always balance the strength of both teams to make sure that each side's win rate is as close as possible to 50%. To achieve balance in MOBA games, the following two problems must be solved at the same time.

Game Data Balance

Game data balance means that when both sides have almost the same leveled players, the internal parameters of the game itself are able to make the two sides have almost the same win rate. Of course 50% is always the idealized model which can never be achieved. What designers can do is to make it as close as possible to 50%, for there are thousands million choices in every game, from hero pick and item build to ability cast and even every move.

Game data balance is subdivided into static data balance and dynamic data balance. Static data balance means that after the design and development, but before the release of the game, all the parameters in the game are in balance. Dynamic data balance means after the feedback during internal test and public beta, the production team adjusts data and add new elements through maintenance and update to achieve a better balance. Chapter 5 will elaborate in the process of achieving data balance in a MOBA game like DOTA2.

3.1 Static Data Balance

Static data balance is the balance that exists in the completed game without any feedback or updates. Once a MOBA game's development stage is finished, assume there were two teams with the same level players, and without any updates for the game itself both sides have the same win rate. In this idealized condition, the MOBA game is considered in *static data balance*. Some simple games in real life satisfy this model, such as *"Scissors, paper, rock"*, where each player has a 50% win rate.

Firstly consider *"Scissors, paper, rock"*, which follows the ideal static data balance because of its simple rules: Scissor>paper, paper>rock, rock>scissor. These three rules made up this game simple but fair. When it comes to a MOBA game, thousands million unpredictable factors influence the result every second, even a right click on the map will cause a butterfly effect. So the ideal 50%-50% can never be achieved provided that the designers are human beings. In order to get closer to this model, designers and developers are allowed to make a series of tests and adjustments in internal versions. The basic static data balance should thus be accomplished to a certain extent.

An excellent game design blueprint is always highly balanced even before the game's development, which can also make it much easier for developer to make adjustments after the release to achieve even more satisfactory a balance.

With more and more complicated games coming out, one tries to treat the development of a game as the development of a system because of their similar complexity. A balanced good game should always follow the following three rules in all stages of design and development.

The modularity of game elements

The modularity of game elements can be summarized as follows: Every element in the game only exist for one or a few necessary purposes. If this rule can be followed from the very beginning, designers will not have much trouble fixing or adjusting any problem later on, for any adjustment won't affect other parts of the game.

For example, there used to be a "mixed" type of damage in DOTA, which was really a nightmare for the designer Icefrog. This type deals full damage on creeps, but deals both magic and physical damage to hero units, which means hero's magic resistance and armor must both be considered to calculate the damage reduction (usually normal damage only concerns one of theses two). Beside, mixed damage ignores the "magic-proof", a status that can protect units from magic damage. For all the above reasons, Icefrog needed to do implement many more calculations when he tried to do some adjustments related to the "mixed" type of damage to achieve the expected result. This problem is caused by the *Warcraft III* game engine. When it came to DOTA2 with the new Source engine, Icefrog immediately replaced the "mixed" damage with "pure" damage, with the new setting: ignore "magic-proof", armor and magic resistance, all damage reduces by 25%. Since then, lots of adjustments became much more convenient.

Excellent modularity is the premise to design a balanced game without many extra, unnecessary adjustments. Later updates will be also much easier for the designers with this habit.

Coherent design principle

The coherent design principle [7] is the most important rule for every game designer in the initial design stage. It means all the elements should be designed based on the overall concept of the game. An element that is independent of the concept of the game will bring about a great deal of imbalance.

One of the most important concept of DOTA is that "every player controls a unique hero unit with corresponding abilities to compete". As an example that does not fit this concept consider Arc Warden. This hero's ultimate ability is "Tempest Double". Once activated, there will be two Arc Wardens with individual health/mana points, status, ability cool down and mobility. So we actually have two heroes at the same time. Compared with other heroes, this is too strong. Due to its imbalance, players decided by default not to pick this hero to play, until many times modification which made this hero almost the same level with others.

Complexity Suppression

The essence of Complexity Suppression can be summarized as follows "keep things simple and easy to understand". The whole design must be simple and clean, basic concepts should be always followed through all stages to make sure every part is necessary. The more complicated the originally design is, the more difficult later adjustments will be. Making the game concept easy to understand for players is also every designer's duty.

Beside the three rules above, *micro-regulation* is the next important step to achieve basic static data balance. All types of data require reasonable ranges. The biggest challenge for micro-regulation is how to find problems. During this procedure, the modularity of game elements and the coherent design principle will be essential. Without them, finishing the basic balance in reasonable time would be impossible.

The most effective way for micro-regulation is to do lots of game tests: The testers and the game designers discuss certain conditions, predict the consequences, then verify if the conjecture is correct through actual operation; if not, analyze what is reasonable and what needs adjustments.

3.2 Dynamic Data Balance

Dynamic data balance is the interaction between players and game designers through feedback and adjustments to keep the game in a healthy balance all the time.

It actually means that designers fix bugs and do adjustments according to players' reaction and feedback after the game is released. It is still a test in some way, but the group of testers is much large (every single player in the world), which will definitely make problems more concrete and more worth of an improvement.

The basic process of dynamic data balance is: New update released→ Gameplay changes→ Players found some imbalance→ Feedback problem→ Summarize and adjust imbalanced data→ Release next update (Figure 3.1)

Patch Version	Release Date	Highlights
7.08	2018-02-01	▪ Balance Changes
7.07d	2017-12-19	▪ Balance Changes
7.07c	2017-11-17	▪ Balance Changes
7.07b	2017-11-05	▪ Balance Changes
7.07	2017-10-31	▪ Two New Heroes: Dark Willow and Pangolier ▪ Five New Items: Aeon Disk, Kaya, Meteor Hammer, Nullifier, and Spirit Vessel ▪ Removed Two Items: Iron Talon and Poor Man's Shield ▪ Hero Attribute Reworks ▪ Talent Tree Reworks
7.06f	2017-08-20	▪ Balance Changes
7.06e	2017-07-02	▪ Balance Changes
7.06d	2017-06-11	▪ Balance Changes
7.06c	2017-05-29	▪ Balance Changes
7.06b	2017-05-21	▪ Balance Changes

Figure 3.1 DOTA2 updates

The way to achieve dynamic data balance in MOBA games is similar with micro-regulation at a test stage of static data balance. The difference is that much more data is available from players all over the world. Again we focus on DOTA2 as an example. Its company Valve has a dedicated team to accept reports from players around the world in the DOTA2 community. If too many players report a single imbalanced problem (or a bug), analysis and adjustment will be considered for next update. Another important reference is Professional Tournaments. There are around 100 games in a DOTA2

professional tournament [9]. From the behaviour of professional players, most based on the popularity of the heroes and items, Valve analyzes which heroes/ items are picked the most and which the least. With the principle "balance every single hero", they will strength as the most unpopular heroes and nerf (weaken) the hottest. For example, in the International 2013 tournament, almost every hero on "Most Ban/Pick" list from figure 3.2 got a nerf in the next update to weaken their popularity. This idea is simple but effective: in most conditions, the more powerful a hero is, the more popular it will be, especially in professional games. While every DOTA2 tournament consists of around 100 games, with this limited data collection (Figure 3.2), only significant imbalance can be reflected, while some more imbalanced factors may not be observed. Also professional players with superb skills are only a very small part of the DOTA2 community. Ignoring thousand millions ordinary players around the world makes this method unable to reflect the imbalanced factors comprehensively enough.

The above makes the evaluation one-sided and not real-time (there is usually a big professional tournament every 2 to 3 months). This is the main problem we try to address here. In Chapter 5, a new method based on data analysis for all players is developed.

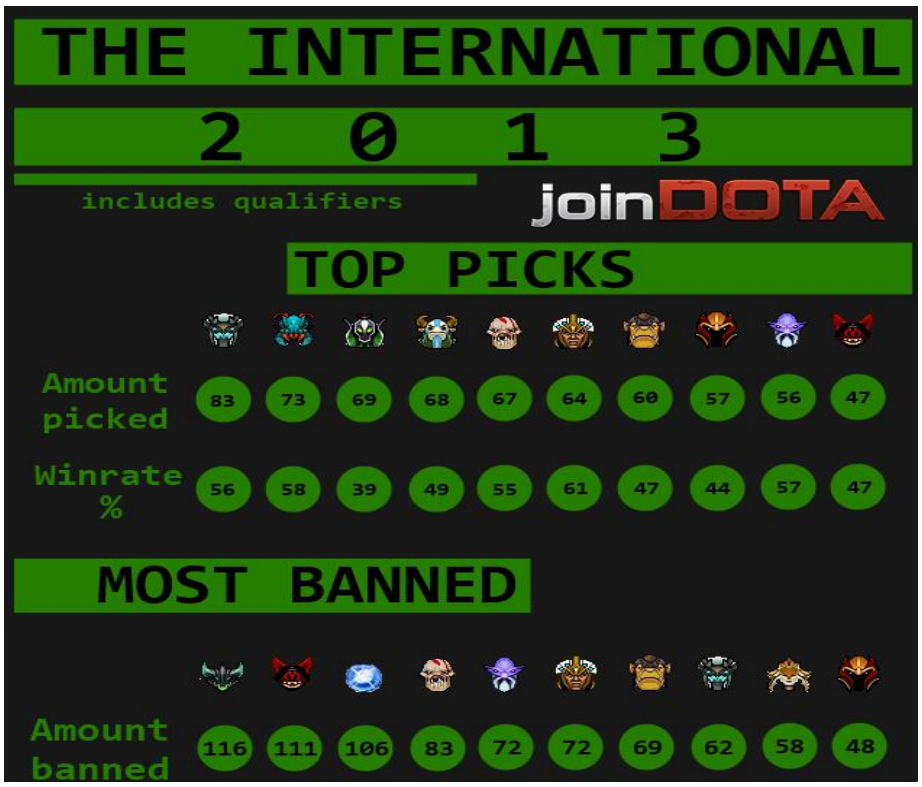

Figure 3.2 Most Ban/Pick Heroes in the International 2013

3.3 Matching System Balance

A player's level depends on experience, personal nerve reaction time, teamwork awareness and physical/mental status. Without a reasonable matching system that can give every player an appropriate evaluation on their level, it would be impossible to set up both team to have a similar overall level before the game starts. Fortunately, even if most MOBA games' matching system are not able to judge every player's level every movement, they still have a general judgement for every player according to their performance in a large number of past games. Maybe a player has a abnormal performance in one or two games, but the evaluation will have a dynamic nature based on an overall judgement. After awhile, the evaluation system can objectively reflect the player's level. In this way, it is easy to see that the most convenient and intuitive method is to construct a numerical evaluation system according to player's performance game by game.

DOTA2 has an excellent matching system with MMR (Match Making Rating) for every single player. In Chapter 5, through a detailed analysis of DOTA2 matching system, an ideal matching system with improvements will be developed.

Chapter 4

Preliminaries

4.1 DOTA2 Matching System

An idealized matching model can be established based on a rank system: Select 10 players with the same rank scores to start a game; once the game is completed, each winner can earn certain scores, while each loser loses the same scores.

It is easy to see that this ideal model is difficult to achieve in real life, with limited time and players: 5 to 10 minutes would be the maximum matching time for a normal player to tolerate, and finding 10 players with exactly the same scores within this time might be an impossible mission.

Based on this, DOTA2 did some effective adjustments on its matching system to provide a better experience. While under this system the changes in scores may surprise players after a game, these changes are all reasonable from a statistical point of view.

For example, the most important principle in DOTA2 matching system is that both sides have a similar average score, which has the largest weight in the final matching. If one side has a significantly higher average score than the other side, they will gain less if winning the game, or lose more if they lose the game. The opposite is true for the other side. In this system, the higher-score team also has a higher win rate; if the result goes the other way, the system will reward the lower-score team more and also punish the higher-score team more, as a compensation for the matching system. Language is also an important factor in MOBA games. DOTA2 matching system tries to match all the players who have the same first language based on their setting, or at least the 5 players on same side have the same language.

Until now, the matching system only cares about matching 10 solo players, but in most cases, 2 to 5 party players start a game as a group. Considering that most people would have different behaviour and game styles with their friends in real life rather than strangers in the game, DOTA2 has two individual rank systems as “solo” and “party” scores, respectively are used in these two modes.

The Elo Rating System:

The Elo rating system [20] is a method for calculating the relative skill levels of players in zero-sum games such as chess. It is named after its creator Arpad Elo, a Hungarian-born American Physics professor.

The Elo model was originally based on a normal distribution. However, practice shows that the performance of the players is not normally distributed, so the current hierarchical scoring system usually uses a logistic distribution.

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}} \tag{4.1}$$

where:

x_0 = the x-value of the sigmoid's midpoint,

L = the curve's maximum value, and

k = the steepness of the curve.

If Player A has a rating of R_A and Player B a rating of R_B, the exact formula (using the logistic curve) for the expected score of Player A is:

$$E_A = \frac{1}{1 + 10^{(R_B - R_A)/400}} \tag{4.2}$$

Similarly the expected score for Player B is:

$$E_B = \frac{1}{1 + 10^{(R_A - R_B)/400}} \tag{4.3}$$

Supposing Player A was expected to score E_A points but actually scored S_A points. The formula for updating their rating is:

$$R'_A = R_A + K(S_A - E_A) \tag{4.4}$$

The K factor is based on the scoring rules and depends on what is the score unit for each game (10, 50, or even 100). S_A= 1 if player won the game, else S_A=0.

In the Elo rating system the new updated rating for a player is only related to his original rating, the outcome of the game (win/lose) and the opponent's rating before the game, which satisfies the basic ranking/matching system of DOTA2. Also players' behaviour is similar in competitive games (as shown in Figure 4.2). Finally, official description of another MOBA game *"League of Legends"*

confirms the fact that its rank/match system is based on Elo ratings [16]. Considering their high level of similarity, we assume that DOTA2 follows the Elo rating system in the same way.

4.2 K-Means Cluster Analysis

K-Means cluster analysis will be used in this thesis to cluster all the heroes into different types according to their data, the same type of heroes can be assessed with the same standard afterwards.

Given a set of observations (x_1,x_2,....x_n), where each observation is a d-dimensional real vector, K-means clustering [19] aims to partition the n observations into k (≤ n) sets S = {S_1, S_2, …, S_k} so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). That is:

1. Randomly select k cluster centroids as μ_1,μ_2,...μ_k.
2. Repeat the process below until convergence

For every sample i, determine its cluster according to calculation to find its nearest centroids as the cluster result for this iteration:

$$c_i := \arg\min_j \| x_i - \mu_j \|^2 \tag{4.5}$$

with ‖ ‖ as modulus of N-dimensional vector. For every cluster, recalculate the corresponding centroid by calculating average coordinate among all the vectors in this cluster.

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c_i = j\} x_i}{\sum_{i=1}^{m} 1\{c_i = j\}} \tag{4.6}$$

with k the number of clusters set manually, c_i the nearest cluster among k to sample i (one of 1 to k). and μ_i the center of a cluster.

4.3 Artificial Neural Network

BP-Neural Network will then be used to determine the specific weights of each type of data for every type of heroes respectively, for the final complete evaluation system.

An Artificial Neural Network (ANN) [21] consists of numerous simple units connect as a network. It was firstly developed by Meculloch and Pitts in 1943. ANN as a neural model reflects the function of human's neural system, which is parallel processing of nonlinear system.

There are several kinds of ANN algorithms. Based on our data processing requirement, this thesis will use BP-Neural Network [18].

BP-Neural Network

The BP-Neural Network was introduced by Rumelhart in 1985, based on the backpropagation algorithm. It is a multi-level learning network with supervised learning. The network structure is shown in Figure 4.2.

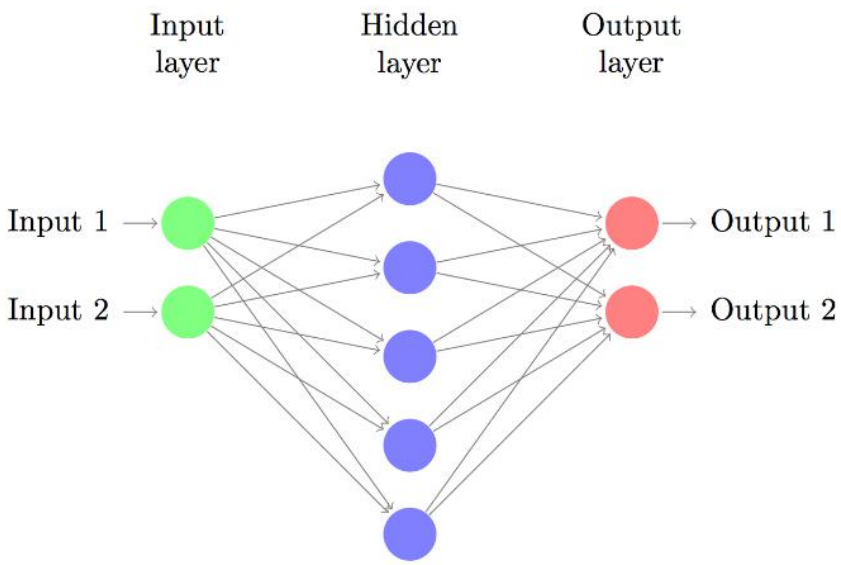

Figure 4.2 BP-Neural Network Structure

The learning algorithm of a BP-Neural Network is as follows:

1. Set initial parameters ω and θ, (ω is the initial wight, and θ the threshold)
2. Train the samples on the network, calculate y_i with the formula below:

$$y_j = \left[1 + e^{-(\sum_i w_{ij} x_i - \theta_j)} \right]^{-1} \tag{4.7}$$

with x_i as input (i=1,...m); ω_{ij} as connection weight from i to j (i=1,...,m, j=1,...n), initial weights set as small number between [0,1]. θ_j as threshold, y_j as output.

3. Adjust weight coefficient ω according to the difference between the known output and the calculated output from above (d_j-y_j) to:

$$\Delta\omega_{ij} = \eta\delta x_j \tag{4.8}$$

The scale factor η, also called learning rate, is set as a small number between [0,1] according to the following principle: in order not to cause shock and to ensure better accuracy, increase the value of η gradually until it reaches a satisfactory training degree. x_j is the input for the hidden neurons (j=1,...n), d_j is the known output (training data), and δ is a value related to the output deviation:

$$\delta = \eta_j(1 - y_j)(d_j - y_j) \tag{4.9}$$

The outputs of neurons in hidden layers cannot be compared directly, So after the reverse projections:

$$\delta = x_j(1 - x_j)\sum_k \delta_k \omega_{jk} \tag{4.10}$$

where k ranges over the upper nodes, so the deviation δ is calculated inversely from the output layer by layer, and the weights of neurons in all layers are adjusted as follows:

$$\omega_{ij}(t) = \omega_{ij}(t-1) + \Delta\omega_{ij} \tag{4.11}$$

with t as the number of learning times.

This algorithm is an iterative process, adjusting the values of ω in each round. The iterations continue until the output error is less than a certain allowable threshold. Then the network training is finished.

The role of a BP-Neural Network is to convert the input-output problem of a group of samples into a nonlinear optimization problem, which uses a gradient descent algorithm most commonly used in optimization techniques.

Chapter 5

Balance Implementation in DOTA2

5.1 Evaluation System for the Dynamic Data Balance of Heroes

Static data balance is only a concern in the research and design stages of a game. Especially important for a MOBA game like DOTA2 is to keep a guarantee of its dynamic balance by patches and updates. In order to accurately evaluate the balance/ imbalance factors, this thesis will establish an evaluation system for Heroes.

Since the purpose is to find potential imbalanced factor, win rate (Figure 5.1) and damage dealt (Figure 5.2) would be good choices to judge if a hero is much too strong [4].

Hero	Win Rate	Matches Played
Lycan	58.34%	143,108
Zeus	54.76%	413,685
Vengeful Spirit	54.74%	188,403
Chaos Knight	54.68%	202,533
Underlord	54.55%	159,830
Omniknight	54.46%	253,389
Huskar	54.24%	166,247
Shadow Shaman	54.10%	574,525
Outworld Devourer	53.99%	292,948
Centaur Warrunner	53.84%	126,990

Figure 5.1 DOTA2 Top 10 Win Rate Heroes in Very High Skill Games

January, 2018

Hero	Damage per Minute	Healing per Minute	Building Damage per Minute
Tinker	998.60	4.18	4.52
Zeus	977.91	1.33	11.21
Spectre	762.40	3.01	51.00
Bristleback	728.42	2.52	47.22
Sniper	722.93	0.34	95.24
Gyrocopter	687.39	1.03	91.83
Arc Warden	682.03	0.84	84.65
Huskar	671.66	1.95	150.33
Ember Spirit	666.30	1.69	32.43
Outworld Devourer	657.18	0.89	58.38

Figure 5.2 DOTA2 Top 10 Damage Dealt Heroes in Very High Skill Games

January, 2018

There are 17 different heroes in the figures above. Zeus, Huskar and Outworld Devourer appear in both figures. Considering more types of data is needed and will be collected through an API later, but we will only focus on these heroes in this thesis to simplify data processing.

Then, a 10-player sample (random selected from writer's friends list) with the behaviour of these 17 heroes is collected as shown below (Figure 5.3) in all games they have played (range from 2 to 23 games individually) in January (in Figure 5.3, damage dealt is used as an example):

	A	B	C	D	E	F	G	H	I	J	K
1	GAME ID	Sccc	AI(EASY)	Rapier Rapier	Xiyan	Xphotograph	报复社会MO	Angelina	K_ASS	LuciferShana	NineG
2	Lycan	13850	10582	16854	11258	9825	11367	7845	18451	14526	10486
3	Zeus	26482	19853	22684	13698	19874	23658	12548	19269	21256	17987
4	Vegenful Spirit	7654	9845	4856	16504	5482	12574	8416	4019	14253	6874
5	Chaos Knight	11263	8764	12541	4216	9841	9331	8949	12577	10471	9096
6	Underlord	8945	5698	9874	11025	5476	9841	13024	9006	10104	4169
7	Omniknight	5612	4875	3641	4028	6214	3210	3987	4699	4251	5966
8	Huskar	10582	7685	18752	8481	7985	11374	10244	6934	17772	9424
9	Shadow Shaman	4012	4862	3684	5024	4687	4024	3940	3169	4127	5874
10	Outworld Devourer	14588	15862	12485	11487	12368	13674	13024	10147	15487	9871
11	Centaur Warrunner	9625	7685	11254	8947	7587	8714	5922	10478	6988	8744
12	Tinker	24960	11253	9874	20147	9897	14782	8770	18973	24873	14876
13	Spectre	26630	24563	17845	14793	18972	11258	20481	18633	10474	22103
14	Bristleback	18542	12485	15846	9877	15693	14870	11254	17451	12544	10024
15	Sniper	12368	11257	16485	19832	10166	11985	14120	9046	16870	12205
16	Gyrocopter	18963	9632	8746	20460	14885	7966	10441	7012	18969	17543
17	Arc Warden	8624	5762	20148	4980	7691	9822	22687	19890	10146	23662
18	Ember Spirit	14632	16875	11684	12486	13633	19890	9923	10207	13362	17439

Figure 5.3 Average Damage Dealt per Game Using 17 Heroes by 10 Players

January, 2018

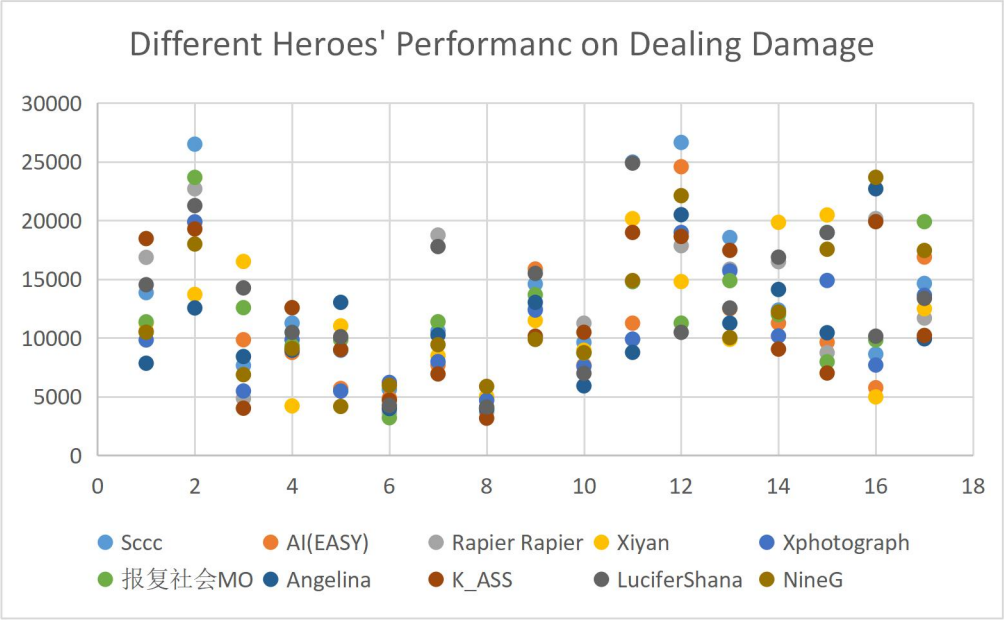

Figure 5.4 Different Heroes' Performanc on Dealing Damage

According to the distribution in Figure 5.4, it is easy to see that different heroes' ability to deal damage differ substantially. Thus damage dealt seems to be an important standard to evaluate if a

specific hero is too strong or not. However there are more factors that must be considered for a complete picture.

Different heroes have different positions, attributes and abilities. All these features of various heroes have different effects in every DOTA2 game. For example, some heroes are meant to deal a huge amount of damage; some on the other hand are good at limiting enemy heroes' actions with stuns, slow, silence, etc; some others are supposed to help teammates by healing and take damage from the enemy. Focusing on one or a few specific data is not a wise way to evaluate heroes.

Figure 5.5 show three different types of heroes: Zeus, has 4 damage-dealing abilities, so that even a bad player can deal a lot of damage with it. Centaur Warrunner, a representative tank, is supposed to take the most damages from enemy in every team fight, and also deals some damage at the same time. Shadow Shaman, usually played as a support, helps cores (damage dealers) have a better environment to farm and stun enemy heroes to let allies have easy kills.

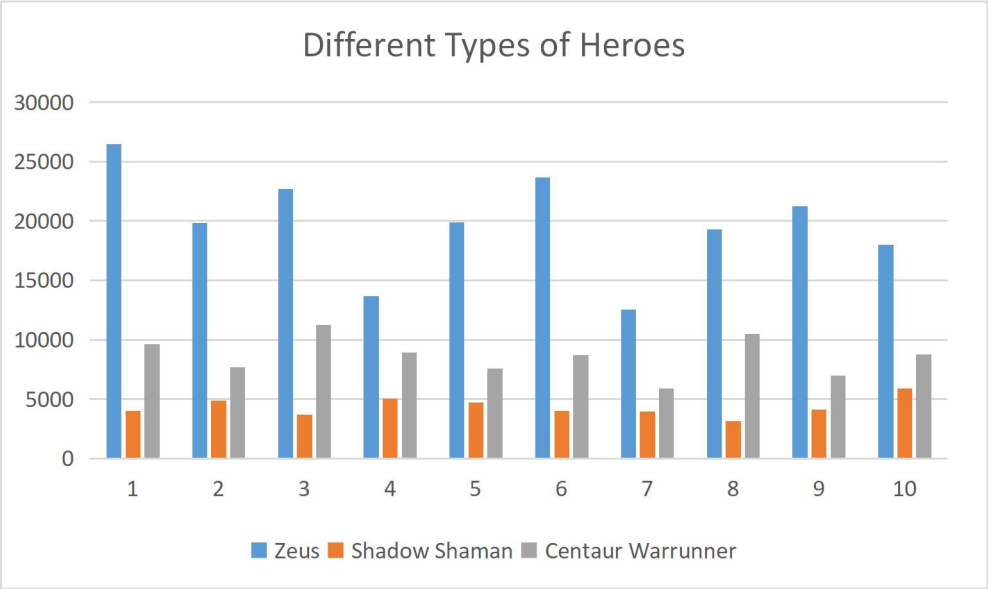

Figure 5.5 Different Types of Heroes

These samples show that the evaluation should be done in a comprehensive way considering all the abilities of a hero. In this aspect, even the data analysis website dotamax.com still have many deficiencies in that it features too few types of data. Fortunately, DOTA2 itself has an application programming interface (API) for its database including every single game's detailed data [6,10], as shown for example in Figure 5.7. From these schemes, many more types can be obtained directly

such as “damage taken” or “stun time” for every single game. With all these data, comprehensive analysis becomes possible.

```
COMBAT SUMMARY
--- npc_dota_hero_disruptor ---
  Total Damage Done: 5133
    to npc_dota_hero_tiny: 810
    to npc_dota_hero_legion_commander: 1632
    to npc_dota_hero_tinker: 560
    to npc_dota_hero_tusk: 1562
    to npc_dota_hero_crystal_maiden: 569
    - ability damage -
    with disruptor_thunder_strike(0): 1970
    with disruptor_static_storm(5): 1432
    other damage done: 1731
  Total Stuns: 33.50 seconds
--- npc_dota_hero_zuus ---
  Total Damage Done: 56111
    to npc_dota_hero_tiny: 12793
    to npc_dota_hero_legion_commander: 12126
    to npc_dota_hero_tinker: 12201
    to npc_dota_hero_tusk: 12186
    to npc_dota_hero_crystal_maiden: 6805
    - ability damage -
    with zuus_arc_lightning(0): 11479
    with zuus_lightning_bolt(1): 15111
    with zuus_static_field(2): 8975
    with zuus_thundergods_wrath(5): 19377
    other damage done: 1169
  Total Stuns: 82.73 seconds
  Total Slows: 505.46 seconds
--- npc_dota_hero_batrider ---
  Total Damage Done: 17973
    to npc_dota_hero_tiny: 7656
    to npc_dota_hero_legion_commander: 2478
```

Figure 5.6 Data Summary For a Single DOTA2 Game

```
CREATE EXTENSION IF NOT EXISTS tsm_system_rows;

CREATE TABLE IF NOT EXISTS matches (
  match_id bigint PRIMARY KEY,
  match_seq_num bigint,
  radiant_win boolean,
  start_time integer,
  duration integer,
  tower_status_radiant integer,
  tower_status_dire integer,
  barracks_status_radiant integer,
  barracks_status_dire integer,
  cluster integer,
  first_blood_time integer,
  lobby_type integer,
  human_players integer,
  leagueid integer,
  positive_votes integer,
  negative_votes integer,
  game_mode integer,
  engine integer,
  radiant_score integer,
  dire_score integer,
  picks_bans json[],
  radiant_team_id integer,
  dire_team_id integer,
  radiant_team_name varchar(255),
```

Figure 5.7 Types of Data That Can Be Called Directly From DOTA2

The following are the factors that are significant in the assessment of a hero:

Win rate: the most important factor to evaluate, weight S is given.

Damage dealt: Can be divided into two subclasses as Building damage dealt, and Hero damage dealt. Since the only way to win a DOTA2 game is to destroy the enemy's Ancient base, and killing enemy's heroes would lead to an easy push, these two factors are both important for every single game. Weight A is given.

Time of Stun and Hex: Stun and Hex can totally restrict any of enemy heroes' actions, which will lead to an easy kill. However Stun and Hex themselves cannot make a killing happen like damage dealt, so they have a lighter weight B.

Time of Debuff: Including silence, root, slow and mute. These debuffs can only restrict one of the abilities such as move, attack, using ability or items, so they are even weaker than Stun and Hex. Weight C is given.

Buff and Heal: Buff provides a beneficial effect for allies including speed up, and extra damage. Healing helps allies regenerate their health/mana points. These two factors have the same weight as debuff: C.

Damage Taken: Taking damage from enemy may help, but it is not necessary all the time. Sometimes having a high capacity to take damage may even promote mistakes. In all this is weaker than all the factors above; weight D is given.

Support Ability: Mainly reflected in purchasing of supportive items for the whole team, such as wards to provide vision and dust/sentry for detection, supportive behaviour is very important in MOBA games including DOTA2; we give the weight B for it.

Obtaining accurate weights requires massive computation and iterative verification based on big data. Our aim is to provide a method rather than complete the calculation.

We then construct the formula for the assessment of heroes as follows:

$$M = \sum_{i=1}^{n} W_i * F_i \tag{5.1}$$

F_i, represent the normalized average values for the factors listed above, based on the data from all DOTA2 games over a period of time, and W_i are the corresponding weights listed above. M is the Balance Value.

In a certain period of time (for example a month after the last update is released), if a specific hero has a significantly abnormal balance value (too high or low), it is very likely that the last update has

a very imbalanced effect on this hero. A fix may be needed after more tests to decide whether a new update is necessary.

Based on the general idea outlined above, it is easy to see that the “damage dealt” (to heroes or towers) type heroes always have the best assessment since these two data have the highest weights all the time. In practice things are not that simple. Many imbalanced heroes are so because of their different functional abilities, such as long-time stuns and huge amounts of hero healing. The assessment above is therefore not suitable for every unique hero. A better classification is required.

Besides the Damage Dealt data above, two more data are now considered: Stun Time and Hero Healing (as a representative of Buff). There is no big data directly available for these two on the statistics website, Instead we obtained the other 2 types of game data from the combat summary of the 10 players mentioned earlier using 17 heroes in January (Figure 5.8). Below is the average statistics for Stun Time, Hero Healing, and Damage Dealing.

	A	B	C	D
1		Average Damage	Average Stun(s)	Average Healing
2	Lycan	12504.4	22.4	3425.5
3	Zeus	19730.9	42.8	80
4	Vegenful Spirit	9047.7	87.6	194
5	Chaos Knight	9704.9	99	864.2
6	Underlord	8716.2	63.5	1109.6
7	Omniknight	4648.3	18.3	5804.8
8	Huskar	10923.3	8.6	1095.2
9	Shadow Shaman	4340.3	156.4	92
10	Outworld Devourer	12899.3	18.9	0
11	Centaur Warrunner	8594.4	105.2	889.6
12	Tinker	15840.5	39.5	162.5
13	Spectre	18575.2	10.4	620.6
14	Bristleback	13858.6	12.3	1824.5
15	Sniper	13433.4	8	0
16	Gyrocopter	13461.7	48.8	0
17	Arc Warden	13341.2	22.9	135
18	Ember Spirit	14013.1	9.5	0

Figure 5.8 Average Statistics of 17 Heroes

We are now ready to perform clustering analysis [15] using Weka. First, we need to determine the number of clusters.

According to the rules of DOTA2, Heroes are divided into two primary roles, known as the Carry and the Support. Carries, which are also called "cores", begin each match as weak and vulnerable, but are able to become more powerful later in the game, thus becoming able to "carry" their team to victory. Supports generally lack abilities that deal heavy damage, instead having ones with more functionality and utility that provide assistance for their carries. So basically a Carry is responsible

for dealing huge amount of damage, while Supports create better chances for their Carries, eg, by controlling the enemy (Stun) or by healing allies. However in our opinion, two classifications are not enough. Some heroes in DOTA2 can deal some damage (maybe not that much though), while offering Stuns and Healing at the same time. Most players called this kind of heroes Functional Cores.

We therefore submit that three types called “Carry” “Support” and “Functional Core” are needed.

```
Final cluster centroids:
                                Cluster#
Attribute          Full Data          0          1          2
                      (17.0)      (5.0)      (1.0)     (11.0)
============================================================
AverageDamage     11978.4353     8142.3     4340.3 14416.5091
AverageStun          45.5353      74.72      156.4    22.1909
AverageHealing      735.1471    1612.44         92   394.8455
```

Figure 5.9 Result of Clustering Analysis on 17 DOTA2 Heroes Data

In Figure 5.9, Cluster 0, 1, 2 represent Functional Core, Support, and Carry, respectively. Hence Cluster 1 mainly contributes Stun, Cluster 2 mainly contributes damage, and Cluster 0 does pretty well on Damage, Stun, and Healing at the same time. Concretely the only element in Cluster 1 is Shadow Shaman, which is indeed an excellent support position hero in DOTA2. Also, Vegenful Spirit, Chaos Knight, Underlord, Omiknight and Centaur Warrunner are indeed able to deal some amount of damage and offer healing and stun at the same time. Others are purely damage dealers. The reason there is only 1 support hero is that the sample was picked as 10 most damage dealing heroes and 7 highest win rate heroes, so the data is supposed to have most elements in Cluster 2 and least elements in Cluster 1, which reflects the real world situation. Notice that determining the type of a hero should be based on not only on the three types of data considered here, but also on the other factors discussed earlier. We limit our cluster analysis only on these three data points due to the limited resources allocated to this work. Extending our analysis to more data is however immediate.

Based on the cluster analysis, the rules of weight for heroes’ abilities in formula 5.1 are now improved as follow:

“*Carry heroes*”: Damage dealt> Healing>= Time of Stun and Hex

"*Support heroes*":Time of Stun and Hex> Damage dealt>= Healing

"*Functional cores*": Weights of Damage dealt, Healing and Time of Stun and Hex are almost at the same level.

We used the following principle: Each type of heroes has its specific duty, so the most important capability for a single hero is based on what it is supposed to do (deal damage or support team or limit enemy). The other abilities are not that important compared with its main purpose. In this way, the balance of every single hero can be quantified in the same evaluation system. Afterward, a better balanced update can be completed based on big data on thousands million games rather than only professional tournaments.

With formula 5.1, suppose the weight of 11 "Carry" heroes" are respectively 0.5, 0.3, 0.2 for Damage, Stun, and Heal; and 0.35, 0.35, 0.3 for 5 "Functional Cores". Since only 1 sample in "Support", this type will be ignored here.

We normalize our values using a linear function:

$$y = (x - MinValue)/(MaxValue - MinValue) \tag{5.2}$$

The Balance Value for these two types are then shown in Figure 5.10:

Carry	BalanceValue	FunctionalCores	BalanceValue
Lycan	0.305882353	Vegenful Spirit	0.603650768
Zeus	0.760553204	Chaos Knight	0.675028769
Huskar	0.178206232	Underlord	0.478515032
Outworld Devourer	0.107470106	Omniknight	0.3
Tinker	0.471929363	Centaur Warrunner	0.624677433
Spectre	0.473918548		
Bristleback	0.231839052		
Sniper	0.202656888		
Gyrocopter	0.366235384		
Arc Warden	0.175338899		
Ember Spirit	0.188945456		

Figure 5.10 Rough Balance Value for Carry and Functional Cores

The Balance Value ranges from 0.107 to 0.76. Based on these values all types of heroes can be evaluated at the same level. When there are 113 heroes with all the 7 attributes listed above considered, this range will be smaller and more precise. The ideal model will have a range from 0.4 to 0.6.

Based on the rough estimate above, we now use an Artificial Neural Network (BP) to determine the weight for each type. We only consider the "Carry Heroes" in this thesis as the type has 11 samples, but the extension for other types is immediate.

The inputs are the three types of data namely Damage, Stun and Heal. Each input datum has a neuron (1, 2, 3 respectively) and the output is the evaluation for hero's balance (balanced or not) based on the Balance Value M. The Sigmoid activation function is:

$$O = \begin{cases} 0 & 0.4 <= M <= 0.6 \\ 1 & otherwise \end{cases} \tag{5.3}$$

where O stands for balanced.

To get a complete training set, output must be determined at first that is, which heroes are balanced and which are not. The basic method would be one of the following two: (A) refer to the previous update; if some heroes got modifications, then the data before was unbalanced and the data after modifications is balanced. (B) wait for the next update; heroes with modifications are assumed to have transitioned from unbalanced to balanced.

The logs for DOTA2 replays are only kept for 30 days, which makes it hard to collect data from last update (it is also the reason why only the data from January is collected). We will therefore determine which heroes are balanced and which are not using the following assumption for a rough training of the neural network: we suppose that the heroes who just got modifications in the latest update are balanced, while the others are not.

The balanced heroes are: Lycan, Ember Spirit, Gyrocopter and Tinker. The rest 7 are not.

With the three types of data as input, 3 neurons in the hidden layer and initial weight as 0.5, 0.3, 0.2, after the training with 11 samples, the weights for every neuron are as shown in Figure 5.11:

Hidden layer neurons	Input neurons			Output Neuron
	1	2	3	
1	0.205612	0.102755	0.768041	0.5719874
2	0.672780	0.207483	0.344029	-0.263680
3	0.980589	0.462451	0.016968	0.585587

Figure 5.11 Weight Coefficients Between every Two Neurons

So far, the relations between every 2 neurons are collected, using the method of Wang and Sun [17]:

$$r_{ij} = \sum_{k=1}^{P} W_{ki}(1-e^{-x})/(1+e^{-x}), \quad x = \omega_{jk} \tag{5.4}$$

$$R_{ij} = |(1-e^{-y})/(1+e^{-y})|, \quad y = r_{ij} \tag{5.5}$$

The absolute influence coefficient is:

$$S_{ij} = R_{ij} / \sum_{i=1}^{m} R_{ij} \tag{5.6}$$

with i=1,...,m as input neurons of the neural network, j=1,...n as output neurons, k=1,...P as neurons in hidden layers, W_{ki} as weight coefficients from the hidden layer neuron i to the input neuron k, and W_{jk} as weight coefficients between the output neuron j and the hidden layer neuron k. The absolute influence coefficient S is the required weight.

Based on this method and the data collected, the weights for every type of data are calculated as shown in Figure 5.12:

Data Type	Weight (S)
Damage	0.446598302
Stun	0.240285127
Heal	0.31311657

Figure 5.12 Weight Coefficients of Every Type of Data for Carry Heroes

Based on the result, a relatively complete evaluation system for “Carry Heroes” can be established as:

$$M = \sum_{i=1}^{n} W_i * F_i \tag{5.1}$$

With W_i=[0.446598302, 0.240285127, 0.31311657], and the normalized F_i=[Average Damage, Average Stun Time, Average Heal]^T. For every single hero we have:

$$CarryHero = \begin{cases} balanced & 0.4 <= M <= 0.6 \\ imbalanced & otherwise \end{cases}$$

It should be emphasized again that to simplify the calculation only three types of data are considered. A complete evaluation system will require all the types of data listed earlier.

The process above assumes that the latest updated heroes are balanced”, which is actually an inaccurate method. The ideal accurate method is to always keep all the data for every hero for the latest month. After the new update is released (with the traditional methods mentioned above), we

can pick the heroes that were modified and separate them into two sets: a. the old version set and b. the new version set. With these two sets as training sets for BP-Neural Network the old version set is unbalanced, the new version is balanced. Once the data collection is finished for the new version, we can retrain the system, and the accuracy will increase. After several training sessions with different updates, the final evaluation system can be established.

In other words the sample experiment above is based on an inaccurate assumption (because as mentioned, logs are only kept for 30 days, which makes it impossible to collect data from other updates), which makes it of reduced utility for validating more heroes. The ideal model requires more and persistent data collection for a long time, concerning next several updates. We believe that our system is both feasible and accurate if it is fed with accurate and complete big data. Our system has the following advantages: 1. updates no longer rely on the limited data from professional tournaments, instead all player around the world can be part of it; 2. the new changes in game (like new heroes and new strategies) can be balanced through reevaluation of the related heroes in real-time, the new training can be finished in a short time with assessment readily available for the next update; 3. with more and more training, this evaluation system will become more and more accurate and stable.

Since the third point requires lots of more data for training and test in the future, here we only elaborate the first and second advantages:

Case 1: 113 unique heroes seem to be enough, but this is not actually true for an energetic MOBA game. In fact, Valve Corporation is still designing and releasing new heroes every year to keep the freshness and also some existing heroes are remade with brand new abilities, which is another kind of new heroes. New (remade) heroes can be analyzed and clustered in a short time, in respect to their balance for a quick update, rather than waiting for feedback or data from professional tournaments.

Case 2: With some modifications on the values of its abilities, the position of a certain hero may change. For example, the Hero Monkey King used to be known as a Support, and Naga Siren as Carry. After an update on their ability values (mostly damage and time of stun), Monkey King became Carry and Naga Siren Support. In this case, new clustering analysis and balance evaluation are necessary, which can also be finished as soon as the data has changed with our evaluation system, in a very short time.

Case 3: When new strategies appear, there may be some certain heroes who can take full advantage of them in an important role. For example, there used to be a popular strategy in mid 2014 named the Pushing Strategy. This strategy requires all five heroes as a group and destroy enemy's towers early in the game. Only a few heroes are suitable for this strategy such as Pugna, Nature Prophet and Undying. With increasing popularity and higher win rate than other strategies, all types of data of pushing heroes rose rapidly. If this strategy is too strong in most conditions, then the new Balance Value of this hero will go beyond the balanced range (>0.6). At this time, a nerf about this hero or this strategy is necessary to be considered in the next update rather than waiting for the results of the tournaments.

Example:

Suppose two new heroes are released in an update (since there were no new heroes released during the last 4 months). Here we use Sven and Troll Lord as examples, since they are pure damage dealer carry according to experience.

	Average Damage	Average Stun(s)	Average Healing
Sven	16187.5	29.8	0
Troll Lord	14982.3	13.2	1195.2

Figure 5.13 Average Statistics of 2 Heroes

Figure 5.13 shows the data in the last 15 days from the 10 players sample. Now assume that this is the first 2-weeks worth of data for the two new heroes. With our classification, they are classified as "Carry Hero". To compare them with others, we use formula (5.1) to calculate their Balance Value with weight coefficients W_i=[0.446598302, 0.240285127, 0.31311657]. Outcome is: Sven=0.356003529, Troll Lord=0.293009308.

If the weight coefficients are accurate enough (with enough training), then we can say that these 2 new heroes are a bit weaker than the average level (<0.4). Some positive modifications are required to be considered in the next update. All this analysis can be finished in a short time (2 weeks after the new heroes were released).

This evaluation system based on big data from daily games all over the world can be more accurate and comprehensive, and a real-time reflection of the dynamic data balance of every single hero. Compared with the relatively mature MOBA game DOTA2, it can work even more effectively

on a new one, for which keeping to the balance in an ideal range in a short time would be important to seize the players and the market.

5.2 Ideal Matching System Design Based on the Elo Rating System

Inspired by the principle of the DOTA2 matching and rank system and the evaluation system described in the previous chapter, this part introduces an ideal matching system based on the Elo rating system together with some other improvements to achieve a more comprehensive system, that provides a better balance in support of the MOBA game players' experience. Notice that the DOTA2 matching system is not open-sourced, and so not available. We start from an idal matching system based on matching rules.

An Ideal Matching System

The ideal matching system follows the Elo rating system as follows:

Let K=50 (standard points a player may earn or lose after a DOTA2 game)and S_A=1 (player wins this game) or 0 (player loses this game). Based on DOTA2 rank/matching system, there are two Teams A and B with different average rank scores (R_A and R_B) in the same game, 5 players on each side.

Let D=R_B-R_A, then:

$$E_A = P(D) = \frac{1}{1+10^{D/400}}$$

$$E_B = P(-D) = \frac{1}{1+10^{-D/400}}$$

For convenience, use the Percentage Expectancy Table [22] in the appendix to simplify the calculation process. Notice that sores differences are shown in Table 1 as the median of the difference range shown in Table 2.

Now suppose Team A has an average rank score of 3890, and Team B an average rank of 3700.

With D=190, and with the above formula, we obtain E_A=0.75 and E_B=0.25.

If Team A wins the game then every player in Team A will gain 12.5 points and every player in Team B will lose 12.5 points. On the other hand, if Team A loses the game then every player in Team A will lose 47.5 points and every player in Team B will gain 47.5 points.

Based on the Elo rating system, matching rules can be set up as follows:

1. Both sides have a similar average score.

2. The gap between the players with the highest and the lowest score is small.

3. Both sides have a similar experience that is, a similar number of played games for players on both sides.

4. Scores of highest players on both sides are similar.

5. Both sides have similar number of solo/party players.

6. Complete the matching as fast as possible.

With the above rules, we propose the following process for the matching system.

First, every match making team is a node, and every match is a queue with maximum 10 players. A team can be a solo player with his solo MMR, or a 2-5 party players with their average party MMR.

1. Once a new node comes in, detect if there is a eligible queue for it (with the requirements above, normally the MMR is in range, there is space available for the node, etc.),
2. If yes, then add the node to the queue with minimal score difference.
3. Otherwise, add this node into a empty queue, with the new matching parameters based on the Elo rating system then wait.
4. Once a new node is added to a queue, check if this queue is full.
5. If yes, then find a sever to start this game and empty the queue;
6. Otherwise, keep waiting.
7. Every 30 seconds, check if there are new nodes added into the waiting queue
8. If yes, then keep waiting.
9. Otherwise, expand the condition to a larger acceptable score difference based on the Elo rating system.
10. Repeat from 7.
11. If a queue has waited for 5 minutes, remove all the nodes, repeat from 1 to 6, and empty the queue.

The corresponding flow chart is shown in Figure 5.13.

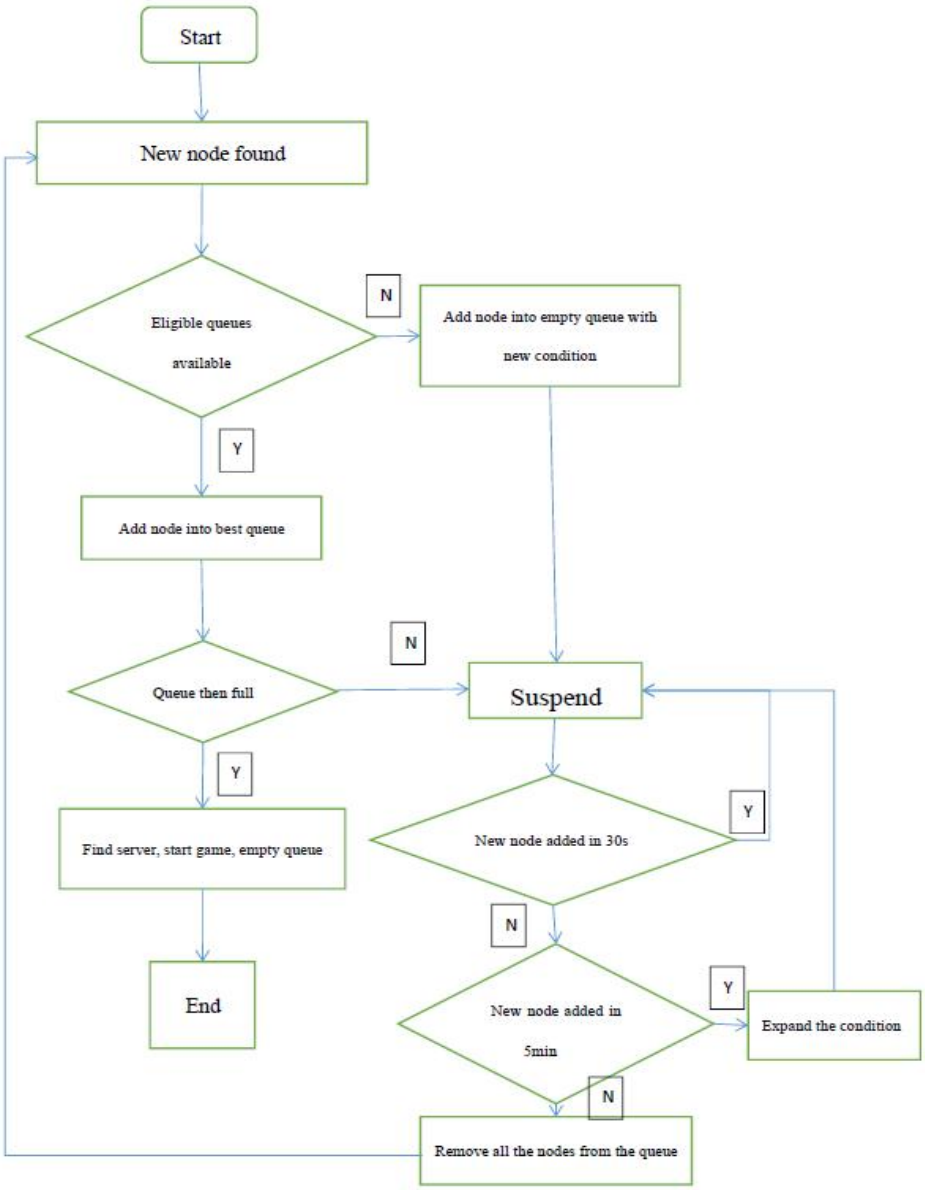

Figure 5.13 Flow Chart for the Ideal Matching System

An Improvement on Scoring Method for the ideal Matching System

The matching system above is based on the DOTA2 rank/matching. The rank system is based on the Elo rating system, which is indeed suitable for most competitive games, in real life as well as in e-sports. However for MOBA games like DOTA2, balance is the most important factor. With 113 different heroes, everyone is supposed to have a seat in this game. The Elo rating system only focuses on the outcome of a game, with no concern about what heroes players use. Also, since the perfect balance in this complicated game can never be achieved, there must be some heroes that are stronger than others, which is also why dynamic data balance is required.

What if within a certain patch, all the players only pick several certain strong heroes to play all the time? In this situation the balance of the game will be destroyed, together with the experience of players. From another point of view, some heroes are meant to be harder to get started than most others; if the designer doesn't encourage players to start with them, the game will lack many different elements. In order to encourage players to play different heroes for a diverse environment, we propose an improvement on the scoring method discussed earlier.

First, we introduce a new concept called Hero Rank. This rank reflects the players' level while playing a certain hero, and is similar to the assessment of heroes in formula 5.1. While the former focused on an individual player's level compared to others', the latter concerns an individual hero's balance among all.

We construct formula as follows:

$$H = \frac{\sum_{i=1}^{n} K_i * E_i}{n} \tag{5.7}$$

H is the rank score for a certain hero of a player; E_i is the data from all the games with this player using the given hero, including Win rate, Damage dealt, Stun/Slow Time, K/D/A, Damage Taken, etc; K_i is the weight for each type of data, which is the same as the weights in formula 5.1 (differences between heroes' types, can truly reflect how important this ability is for a certain hero) and n is the games the player has played with this certain hero. n is effective only when it is greater than a certain number, 10 for example, to make sure the player is familiar enough to this hero for a precise and stable evaluation. This follows the same rule on calibration of DOTA2 rank system.

We then compute T, the Hero-Pool Value

$$T = \frac{\sum_{i=1}^{n} H_i * U_i}{\sum_{i=1}^{n} U_i} \tag{5.8}$$

to evaluate a single player's ability to use different heroes. H_i is the Hero rank for a certain hero; U_i is the number of games this player has played with this certain hero; and n is the number of heroes this player has used.

Notice that even if the total pool is 113 heroes, it is not realistic to suppose that every single player can operate every single hero, so n will be set between 30 to 50, which means that the Hero-Pool

Value only considers a player's highest top 30 to 50 unique heroes' rank, which encourages players to play as many heroes as they reasonably can.

With this value, together with the old formula, we compute the improved final Rank as follows:

$$O = T * k + S \tag{5.9}$$

O is the final rank score, S is the old-version rank score based on the Elo rating system, T is the Hero-Pool value, and k is a constant coefficient given by statistical calculations.

With this improvement, the outcome of the game is no longer the only element that affects a players' rank score in the DOTA2 rank system. Players will be encouraged to try more heroes with more combinations in a team. Therefore more heroes will be used in every play, more data will become available, problems and imbalance are easier found, for an overall better balanced environment. The entertaining value of the game will also increase, with fresh elements for every single player.

The improved matching system is actually the bonus of the evaluation system (the real important contribution), for indeed the evaluation system makes it possible to assess players' ability using different heroes based on Formula (5.1). At the same time, according to statistics, some most popular heroes are played 20 times more than the least popular ones, and this gap has a bad influence both on the game environment and on the data analysis of our evaluation system. Therefore the improvement (5.9) on the matching/rank system will directly encourage players to use more heroes for a balanced environment and more data supply for an accurate clustering analysis and evaluation to all the heroes. This is all a virtuous circle.

Chapter 6

Conclusions

This thesis discusses the MOBA type of games, with its development history and game mechanics. We analyzed the importance of game balance and the way to achieve it in MOBA. As an outstanding representative of such games, we focus on DOTA2 and study ways of improving its balance, both on data and rank/matching system. The balance directly determines the diversity, playability and even life of a game, and the traditional method of determining what data needs to be modified in the next patch in DOTA2 basically relies on players' reports and the behaviour in professional tournaments. Thousands millions players' game data are not considered in this traditional method, which in turn makes it unsuitable for real-time analysis and also not accurate or comprehensive enough. A new method which consists of data collection, cluster analysis and neural network classification has been developed to quantify 113 unique heroes in DOTA2, and to measure the balance. With original data collected from the DOTA2 API, heroes are clustered into three types based on their features and data. A neural network is then used to determine the weights for every piece of data. A Balance Value is then computed as a standard to measure if a certain hero is balanced or not. Based on this, further updates can target these unbalanced factors. Although applied on limited data (30 days worth of logs, only representative types of data being collected to simplify computing, some inaccurate assumptions), this evaluation is still feasible and effective, especially with more time and data support. Also, this method would play a better and more important role in newborn MOBA games than in a game that has become more balanced such as DOTA2. Based on the Elo rating system, an ideal matching system for DOTA2 is designed, together with an improvement based on the hero evaluation system for a better balanced environment and more data supply.

An immediate continuation of this work would be to investigate how to accurately calculate all the W values in Formula 5.1 for each type of data. This thesis used the strategy to assume that the latest updated heroes are balanced, which diminishes the accuracy of the results of training the neural network. An ideal and accurate method is to always keep all the data for every hero for the last

month. After the new update is released (with the traditional methods mentioned above), one can pick the heroes with modifications, and separate them into two sets (old version set and new version set). We then train the BP-neural network with the old version set as “Unbalanced” and new version set as “Balanced”. It should be noted that this method requires more time and data collection. With enough training and our adjustment of the weight coefficients (ideally thought 3 to 4 patches), we believe that this system can become a standard.

This thesis only focused on the data of heroes. Data of items can and should be considered as a factor for data balance as well. Items have the same types of data to heroes’ abilities such as damage, stun, buff, plus price as one more factor to be considered.

In all, the purpose of balance in MOBA is to let every player at similar skill level have a similar chance to win a game with multiple choices, regardless of the variety of heroes and items, all of them at the same level but with different effects. Only following this rule in the future, new MOBA games as well as the relatively developed existed ones can offer a better experience.

References

[1] Regan L.d Mandryk, Diego S Maranan, *False Prophets: Exploring Hybrid Board/Video Games*. Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, USA, January 2002

[2] Dr. Mark Griffiths, *The Educational Benefits of Video games*, Education and Health, P47, Vol. 20 No.3 2002.

[3] John Funk. *MOBA, DOTA, ARTS: A Brief Introduction to Gaming's Biggest, Most Impenetrable Genre*. Polygon. Retrieved 23 December 2015. Available: https://www.polygon.com/2013/9/2/4672920/moba-dota-arts-a-brief-introduction-to-gamings-biggest-most

[4] Conley, K., and Perry, D. *How Does He Saw Me? A Recommendation Engine for Picking Heroes in Dota* 2. Stanford University, 2013.

[5] D. Polancec, *Developing MOBA Games Using the Unity Game Engine*, University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia,2017

[6] Johansson, F., and Wikstrom, J. *Result Prediction by Mining Replays in DOTA2*. Master's thesis, Faculty of Computing, Blekinge Institute of Technology, 2015

[7] Andrew Rollings, Dave Morris, *Game Architecture and Design*, by New Riders Publishing, 2004, P105

[8] Hui-Chun Hsiao *A Brief Review of Digital Games and Learning* ,Pennsylvania State University,2007

[9] Valve Corporation. *The International DOTA2 Championships Official Website.* http://www.dota2.com/international/overview/

[10] B. G. Weber and M. Mateas. *A Data Mining Approach to Strategy Prediction,* in Proc. Computational Intelligence and Games, P140-147, 2009.

[11] M. Stanescu, S. P. Hernandez, G. Erickson, R. Greiner, and M. Buro. *Predicting Army Combat Outcomes in Starcraft*, Conference on Artificial Intelligence and Interactive Digital Entertainment, 2013, Department of Computing Science University of Alberta Edmonton, Alberta, Canada, P86.

[12] Schubert, M.; Kriegel, H.-P. & Züfle, *A. Managing and Mining Multiplayer Online Games*. Institute for Informatics, Ludwig-Maximilians-Universität München, Munich Germany, pp. 441-444, 2011.

[13] N. Pobiedina, J. Neidhardt, M. d. C. Calatrava Moreno, and H. Werthner. *Ranking Factors of Team Success*. In Proc. of the 22nd International Conference on World Wide Web Companion, pp. 1185-1194, 2013.

[14] Valve Corporation. *The Source Engine Documentation,* 2014. Available: https://developer.valvesoftware.com/wiki/SDK_Docs,

[15] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik. *Cluster: Cluster Analysis Basics and Extensions*. R package version 1.15.2, 2014.

[16] Roit Games *League of Legends Matchmaking Explained.* http://forums.na.leagueoflegends.com/board/showthread.php?t=12029

[17] Sun Huijun, Wang Xinhua, *Determination of the Wetion Indexes with Artificial Neural Network Method*. Journal of Shandong University of Science and Technology, Vol. 20 No.3, 2001. P84

[18] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. *Learning Representations by Back-propagating Errors*. Nature. 323(1986), 533-536

[19] MacQueen, J. B. *Some Methods for classification and Analysis of Multivariate Observations*. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, P93. 2008

[20] Wood, Ben. *Enemydown uses Elo in its Counterstrike:Source Multilplayer Ladders*. TWNA Group. Archived from the original on 2009-06-12. Available: https://developer.valvesoftware.com/wiki/Working_Ladders

[21] Marcel van Gerven. *Computational Foundations of Natural Intelligence*. Department of Artificial Intelligence, Radboud University, Nijmegen, Netherlands, [22] Pradu Kannan, Elo Percentage Expectancy Table, Available: http://www.pradu.us/old/Nov27_2008/Buzz/elotable.html

APPENDIX:Elo Percentage Expectancy Table

Based on the Elo rating system (4.2) and (4.3), let $D=R_B-R_A$, E_A and E_B can be easily completed from the Percentage Expectancy Table [22] (Table 1). Further score differences can be obtained from Table 2 with the results in Table 1 as the median.

P(D) Table											
P(D)	D	P(D)	D	P(D)	D	P(D)	D	P(D)	D	P(D)	D
1.00	*	0.83	273	0.66	117	0.49	-7	0.32	-133	0.15	-296
0.99	677	0.82	262	0.65	110	0.48	-14	0.31	-141	0.14	-309
0.98	589	0.81	251	0.64	102	0.47	-21	0.30	-149	0.13	-322
0.97	538	0.80	240	0.63	95	0.46	-29	0.29	-158	0.12	-336
0.96	501	0.79	230	0.62	87	0.45	-36	0.28	-166	0.11	-351
0.95	470	0.78	220	0.61	80	0.44	-43	0.27	-175	0.10	-366
0.94	444	0.77	211	0.60	72	0.43	-50	0.26	-184	0.09	-383
0.93	422	0.76	202	0.59	65	0.42	-57	0.25	-193	0.08	-401
0.92	401	0.75	193	0.58	57	0.41	-65	0.24	-202	0.07	-422
0.91	383	0.74	184	0.57	50	0.40	-72	0.23	-211	0.06	-444
0.90	366	0.73	175	0.56	43	0.39	-80	0.22	-220	0.05	-470
0.89	351	0.72	166	0.55	36	0.38	-87	0.21	-230	0.04	-501
0.88	336	0.71	158	0.54	29	0.37	-95	0.20	-240	0.03	-538
0.87	322	0.70	149	0.53	21	0.36	-102	0.19	-251	0.02	-589
0.86	309	0.69	141	0.52	14	0.35	-110	0.18	-262	0.01	-677
0.85	296	0.68	133	0.51	7	0.34	-117	0.17	-273	0.00	*
0.84	284	0.67	125	0.50	0	0.33	-125	0.16	-284		

Table 1. P(D) Table according to the Elo Percentage Expectancy Table

Corresponding expected score rate based on Scores Difference											
Scores Difference	Expected Score Rate for High scorers	Expected Score Rate for Low scorers	Scores Difference	Expected Score Rate for High scorers	Expected Score Rate for Low scorers	Scores Difference	Expected Score Rate for High scorers	Expected Score Rate for Low scorers	Scores Difference	Expected Score Rate for High scorers	Expected Score Rate for Low scorers
0-3	0.50	0.50	92-98	0.63	0.37	198-206	0.76	0.24	345-357	0.89	0.11
4-10	0.51	0.49	99-106	0.64	0.36	207-215	0.77	0.23	358-374	0.90	0.10
11-17	0.52	0.48	107-113	0.65	0.35	216-225	0.78	0.22	375-391	0.91	0.09
18-25	0.53	0.47	114-121	0.66	0.34	226-235	0.79	0.21	392-411	0.92	0.08
26-32	0.54	0.46	122-129	0.67	0.33	236-245	0.80	0.20	412-432	0.93	0.07
33-39	0.55	0.45	139-137	0.68	0.32	246-256	0.81	0.19	433-456	0.94	0.06
40-46	0.56	0.44	138-145	0.69	0.31	257-267	0.82	0.18	457-484	0.95	0.05
47-53	0.57	0.43	146-153	0.70	0.30	268-278	0.83	0.17	485-517	0.96	0.04
54-61	0.58	0.42	154-162	0.71	0.29	279-290	0.84	0.16	518-559	0.97	0.03
62-68	0.59	0.41	163-170	0.72	0.28	291-302	0.85	0.15	560-619	0.98	0.02
69-76	0.60	0.40	171-179	0.73	0.27	303-315	0.86	0.14	620-735	0.99	0.01
77-83	0.61	0.39	180-188	0.74	0.26	316-328	0.87	0.13	超过 735	1.00	0.00
84-91	0.62	0.38	189-197	0.75	0.25	329-344	0.88	0.12			

Table 2. Corresponding expected score rate based on Scores Difference