 Acta Univ. Sapientiae Informatica **12,** 2 (2020) 183–204

DOI: 10.2478/ausi-2020-0011

Improved balance in multiplayer online battle arena games

Chailong HUANG
Department of Computer Science
Bishop's University
Sherbrooke, Quebec, Canada
email: huang@cs.ubishops.ca

Stefan D. BRUDA
Department of Computer Science
Bishop's University
Sherbrooke, Quebec, Canada
email: stefan@bruda.ca

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 is mostly based on professional tournaments, a small minority of all the games and not real time. We develop 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 in the evaluation system.

1 Introduction

Electronic games [10], especially e-sports games, have become an important part of people's lives. Multiplayer Online Battle Arena (MOBA) games [2] in particular are very popular among young people because of their interesting

Computing Classification System 1998: H.3.3
Mathematics Subject Classification 2010: 62H30, 62M45
Key words and phrases: mupliplayer online battle arena game, game balance, matching system, clustering, neural network

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, and teamwork. The basis of this competitive pleasure lies in its balance.

DOTA (Defense of the Ancient) is the first independent MOBA game. A classic game with a 15-year history, it is still very popular as the second generation DOTA2. Witnessing its enduring popularity, more companies have seen the business opportunities, so a batch of similar MOBA games came out. 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 good. There are even more MOBA games in China, but most 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, and 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.

This paper develops a new method for achieving a better implementation of dynamic balance in DOTA2, and then an ideal matching system is designed as an improvement over the original. That is, the main contribution of this paper is a new method 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 also 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.

2 MOBA games overview

Mutiplayer Online Battle Arena (MOBA) games are also called Action Real-time Strategy Games (Action RTS, ARTS), or DOTA-like Games. This in turn is all a subclass of Real-time Strategy (RTS) Games. In a MOBA game players are usually divided into two camps five versus five, and fight for more gold to buy items and for experience to level up. The ultimate goal is to destroy a certain building of the other side. A MOBA game player usually controls one character only called “hero”, which has specific abilities and slots to equip with

items. Our work focuses on DOTA2, which is a complex game. This section will only introduce (briefly) the elements that are related to our research.

Every DOTA2 game has generally 10 players, divided into two camps. Every player needs to pick a hero to control at the beginning of the game. The two camps have 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 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 at the center of their base.

There are 113 different *heroes*. Based on their major attributes, each hero is one of the following three types: Strength, Agility and Intelligence. There are more attributes combined in every hero beside these three types, including Armor, Move speed, Attack speed, Magic resistance, Health/Mana points, Damage, four abilities with different cool down and damage types, etc.

Most heroes have four *abilities*, three ordinary abilities and one ultimate. 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.

3 Game balance overview

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

The quote above reveals the nature of game balance, namely that every player is supposed to have multiple choices to achieve their goals. 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, others are not. But if only one choice is correct at every crossroad the game will soon become a meaningless repetition, and so becomes gradually boring; this is imbalance.

By contrast, a balanced MOBA game has bad choices, but it also has several good choices every time. In a balanced MOBA game you can try everything to win, unlike a robot who follows the same rules all the time.

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. In MOBA games balance is even more important for playability than in other types of games. 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 situation could be caused by an unbalanced hero or by a bug in the matching system. In all, balance is always the most important factor for every game designer and developer, from beginning to the end.

Game data balance Data balance is subdivided into static and dynamic 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. For dynamic data balance, after feedback during internal test and public beta, the production team adjusts data and adds new elements through updates to achieve a better 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. Section 5 will elaborate on the process of achieving data balance in a MOBA game like DOTA2.

Focusing again on DOTA2 as an example, 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 [14]. From the behaviour of professional players, most based on the popularity of the heroes and items, Valve analyzes which heroes or items are picked the most and which the least. With the principle "balance every single hero", they will strengthen the most unpopular heroes and nerf (weaken) the hottest. However, with such a limited data collection only significant imbalance can be identified. Professional players are a very small part of the DOTA2 community, so this method cannot reflect the imbalanced factors comprehensively.

The above methodology makes the evaluation one-sided and not real time (big professional tournaments only happen every 2 to 3 months). This is the main problem we try to address in this paper; a new method based on data analysis for all players is developed in Section 5.

Matching system balance A player's level depends on experience, personal 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, most MOBA games' matching systems have a general judgement for every player according to their performance in a large number of past games. After awhile, the evaluation system can objectively reflect the player's level.

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

4 Further preliminaries

K-means cluster analysis will be used in this paper to cluster all the heroes into different types according to their data, so that the same type of heroes can be assessed with the same standard. Given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, K-means clustering [8] aims to partition the n observations into $k \leq n$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares or WCSS (i.e., variance). This is accomplished by randomly selecting k cluster centroids μ_j, $1 \leq j \leq k$, determine cluster membership based on the distance from centroids, calculate new centroids by averaging the coordinates of the vectors in the current clusters, and repeat this process until the centroid selection converges.

An artificial neural network [3] consists of numerous simple units connect as a network. There are several kinds of neural networks, but based on our data processing requirement this paper will use a *BP-neural network* [11]. Such a network will be used to determine the specific weights of each type of data for every type of heroes respectively, for the final complete evaluation system.

The BP-neural network is based on the back-propagation algorithm. It is a multi-level learning network with supervised learning, featuring input and output neurons but also a "hidden" layer of neurons. The learning algorithm adjusts the parameters of the neurons based on the training data. Thus, a BP-neural network converts 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.

The Elo rating system [7] is a method for calculating the relative skill levels of players in zero-sum games such as chess. The current hierarchical scoring

system usually uses the logistic distribution $f(x) = L/(1 + e^{-k(x-x_0)})$, where x_0 is the x-value of the sigmoid's midpoint, L is 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, then the expected score of Player A is $E_A = 1/(1 + 10^{(R_B - R_A)/400})$. Similarly the expected score for Player B is $E_A = 1/(1 + 10^{(R_A - R_B)/400})$. Supposing Player A was expected to score of E_A points but actually scored S_A points, the formula for updating their rating is $R'_A = R_A + K(S_A - E_A)$. The factor K is based on the scoring rules and depends on what is the score unit for each game (10, 50, 100, etc.). $S_A = 1$ 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. Players' behaviour is similar in competitive games. Finally, official description of another MOBA game League of Legends confirms the fact that its rank/match system is based on Elo ratings [13]. Considering their high level of similarity, we thus assume that DOTA2 follows the Elo rating system in the same way.

5 Balance implementation in DOTA2

In order to accurately evaluate the balance/ imbalance factors, we will establish an evaluation system for Heroes. Inspired by this evaluation system we then introduce an ideal matching system based on the Elo rating system together with some other improvements to achieve a better balance.

Win rate and damage dealt would be good choices to judge if a hero is too strong [1]. We consider 17 different heroes. Zeus, Huskar and Outworld Devourer appear in the statistics collected (using an API as described later) for both win rate and damage dealt, so we primarily focus on them to simplify data processing. We then collected a 10-player sample (randomly selected from one of the aurhor's friends list) playing with these 17 heroes as shown in Figure 1 in all the games played in January 2018 (ranging from 2 to 23).

The distribution shown in Figure 2 shows that different heroes' ability to deal damage differ substantially, so that damage dealt appears to be important in evaluating whether a specific hero is too strong or not. However there are obviously more factors that must be considered for a complete picture. Different heroes have different positions, attributes and abilities, which in turn have different effects in every DOTA2 game. For example, some heroes are meant

	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 1: Average damage dealt per game using 17 heroes and 10 players (January 2018).

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.

Figure 3 show three different types of heroes: Zeus has a damage-dealing abilities of 4, 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 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 stuns enemy heroes to let allies have easy kills.

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 has 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 [5, 12]. This API allows access to many more types such as "damage taken" or "stun time" for every single game. With all these data, comprehensive analysis becomes possible. The following factors are significant in the assessment of a hero:

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

2. *Damage dealt* can be divided into Building damage dealt, and Hero damage dealt. 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, so these two factors are both important. Weight A is given.

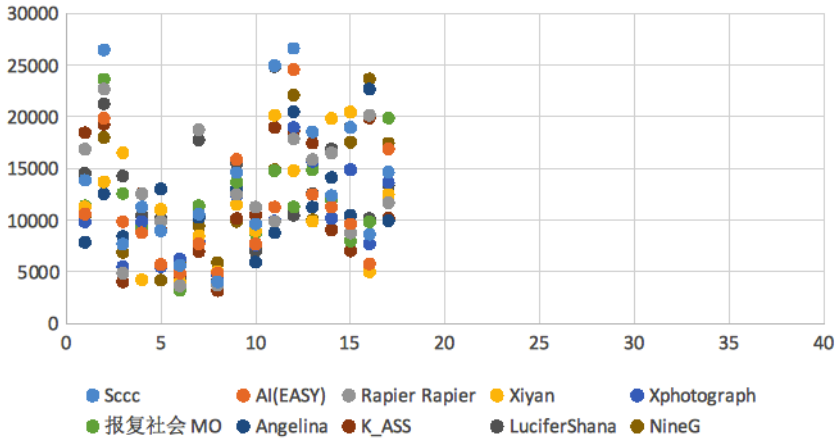

Figure 2: Different heroes' performance on dealing damage.

3. *Time of stun and hex:* Stun and hex can totally restrict any of enemy heroes' actions, which lead to an easy kill. However stun and hex themselves cannot make a killing happen, so they have a lighter weight B.

4. *Time of debuff* includes silence, root, slow and mute. These debuffs can only restrict one of the abilities such as move, attack, using ability of items, so they are even weaker than stun and hex. Weight C is given.

5. *Buff and heal* provide a beneficial effect for allies including speed up and extra damage (buff), and help allies regenerate their health/mana points (heal). These two factors have the same weight C as debuff.

6. *Damage taken* is helpful 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.

7. *Support ability* is mainly reflected in purchasing 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. Once the weights are determined we construct the usual

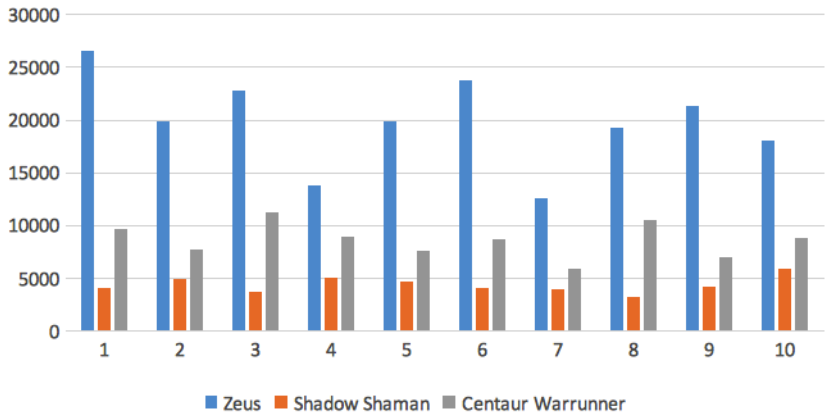

Figure 3: Different types of heroes.

formula for the assessment of heroes:

$$M = \sum_{i-1}^{n} W_i \times F_i \tag{1}$$

where 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 thus becomes the *Balance value*. In a certain period of time, if a specific hero has a significantly abnormal balance value (too high or low), it is very likely that the last update had a very imbalanced effect on this hero. A fix may be needed after more tests to decide whether a new update is necessary.

However, 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 healing. The assessment above is therefore not suitable for every unique hero. A better classification is required.

Besides damage dealt, 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 game data from the combat summary of the 10 players mentioned earlier, using 17 heroes in January 2018. Figure 4 shows the average statistics for stun time, hero healing, and damage dealing. We then performed clustering analysis [9] in Weka.

	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 4: Average statistics of 17 heroes.

The first step is 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 (or "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 functionality and utility that provide assistance for their carries. Basically a Carry is responsible for dealing huge amount of damage, while Supports create better chances for their Carries. However in our opinion, two classifications are not enough. Some heroes in DOTA2 can deal some lower amount of damage, while offering stun and healing at the same time. Most players call this kind of heroes Functional Cores. We thus submit that three types ("Carry" "Support" and "Functional Core") are needed.

Figure 5 shows the summary of our clustering analysis. Clusters 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. The only element in Cluster 1 is Shadow Shaman, which is indeed an excellent support hero. 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

```
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: Result of clustering analysis on 17 DOTA2 heroes data.

is only one 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. Notice that determining the type of a hero should be based 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, we refine the weights for our heroes' abilities in Equation (1) as follows:

1. *Carry heroes*: damage dealt $>$ healing $\geq$ time of stun and hex
2. *Support heroes*: time of stun and hex $>$ damage dealt $\geq$ healing
3. *Functional cores*: the 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 hero is based on what it is supposed to do (deal damage or support or limit enemy). The other abilities are not that important compared with its main purpose. This way, the balance of every single hero can be quantified in the same evaluation system. A better balanced update can be completed afterward based on big data on thousands million games rather than only professional tournaments.

Suppose the weights for damage, stun, and heal, respectively are 0.5, 0.3, 0.2 for the 11 Carry heroes, and 0.35, 0.35, 0.3 for the 5 Functional Cores. Since the Support cluster contains a single sample, this type will be ignored here. We then normalize our values using a linear function ($y = (x - min)(max - min)$). The Balance values for these two types are then shown in Figure 6.

Carry	BalanceValue	FunctionalCores	BalanceValue
Lycan	0.305882353	Vengeful 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 6: Rough balance values 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 then used a BP-neural network to determine the weight for each type. We only consider the Carry heroes in this paper 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 activation function is $\mathsf{O} = 0$ for $0.4 \leq \mathsf{M} \leq 0.6$ and $\mathsf{O} = 1$ otherwise.

To obtain a complete training set some method of determining the output value O is needed. We propose one of the following two methods:

1. Refer to the previous update; if some heroes were modified, then the data before was unbalanced and the data after modifications is balanced.

2. 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 difficult to collect data from last update (it is also the reason why only the data from January 2018 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. According to this

Hidden neurons	Input neurons			Output neurons
	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 7: Weight coefficients between neurons.

Data type:	Damage	Stun	Heal
Weight (S):	0.446598302	0.240285127	0.31311657

Figure 8: Weight coefficients of every type of data for carry heroes.

estimate the balanced heroes are Lycan, Ember Spirit, Gyrocopter and Tinker, while the other seven are not balanced.

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 7.

We refer to the neurons using the indices i for input neurons, j for output neurons, and k for neurons in the hidden layer, with $1 \leq i \leq m$, $1 \leq j \leq n$, and $1 \leq k \leq P$. The relations between every 2 neurons are given by the method of Wang and Sun [4]: $r_{ij} = \sum_{k-1}^{P} W_{ki}(1 - e^{-x})/(1 + e^{-x})$ with $x = w_{jk}$, and $R_{ij} = |(1 - e^{-x})/(1 + e^{-y})|$ with $y = r_{ij}$. W_{ki} is the weight between the hidden layer neuron k to the input neuron i, and w_{jk} is the weight between the output neuron j and the hidden layer neuron k. The absolute influence coefficient is then $S_j = R_{ij} / \sum_{i=1}^{m} R_{ij}$, which is the required weight.

Based on this method and the data collected, the weights for every type of data were calculated as shown in Figure 8. We can establish a relatively complete evaluation system for Carry heroes by plugging in these weights into Equation (1), with $(W_i) = [0.446598302, 0.240285127, 0.31311657]$, and the normalized $F_i = [AverageDamage, AverageStunTime, AverageHeal]^T$. Then a Carry hero is balanced if and only if $0.4 \leq M \leq 0.6$.

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 likely 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 the old version set and the

new version set. These sets can be used for training the BP-neural network, where the old version set is unbalanced and the new version set is balanced. Once data is available for a new version we can retrain the system, making it more accurate. The final ("production") system should be established after several training sessions with different updates.

In all, the sample experiment described 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 persistent data collection for a long time, covering several updates. This being said, 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 game changes (like new heroes and new strategies) can be balanced through the reevaluation of the related heroes in real time; any new training can be finished in a short time, with assessment readily available for the next update.

3. With more training, this evaluation system will become increasingly accurate and stable.

Since the third point requires substantially more data for training and test in the future, we only elaborate on the first and second advantages.

While 113 unique heroes seem to be enough, this is not actually true for an energetic MOBA game. In fact, Valve Corporation is still designing and releasing new heroes every year to maintain freshness. Some existing heroes are also reconstructed with brand new abilities, which in effect create another kind of new heroes. New (or reconstructed) 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.

With some modifications on 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. Using our evaluation system new clustering analysis and balance evaluation can also be finished in a very short time, as soon as the data have changed.

When new strategies appear, there may be some heroes who can take full (and unexpected) advantage of them. For example, there used to be a popular

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

Figure 9: Average statistics of two heroes.

strategy in mid 2014 named the Pushing Strategy. This strategy requires all five heroes as a group and destroys 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 the respective heroes will go beyond the balanced range (> 0.6). Then a nerf on the heroes in combination with this strategy must be considered, and can be considered in the next update rather than waiting for the results of the tournaments.

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. It can work even more effectively on a new MOBA game, for which keeping the balance in an ideal range in a short time would be important to seize the players and the market.

Example Suppose that two new heroes are released in an update. We use Sven and Troll Lord as examples, since they are pure damage dealer Carry according to experience.

Figure 9 shows the data in the last 15 days from the 10 players sample. 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 Equation (1) to calculate their Balance value with weight coefficients $(W_i) = [0.446598302, 0.240285127, 0.31311657]$. The outcome is that the Balance values are 0.356003529 for Sven and 0.293009308 for Troll Lord.

Assuming that the weight coefficients are accurate enough (which will happen 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 therefore necessary in the next update. All this analysis can be finished in a short time (2 weeks after the new heroes were released).

6 Ideal matching based on the Elo rating system

Inspired by the principle of the DOTA2 matching and rank system and the evaluation system described earlier, we introduce an ideal matching system based on Elo ratings together with other improvements to achieve a system that provides a better balance in support of the MOBA game players' experience. Notice that the DOTA2 matching system is not open-sourced, so we start from an ideal matching system based on matching rules instead.

6.1 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. With $D = R_B - R_A$, we have $E_A = P(D) = 1/(1 + 10^{D/400})$ and $E_B = P(-D) = 1/(1 + 10^{-D/400})$. For convenience we use the Percentage Expectancy Tables [6] provided in the appendix to simplify the calculation process.

As an example, suppose Team A has an average rank score of 3890, and Team B an average rank of 3700. We have $D = 190$ and so $E_A = 0.75$ and $E_B = 0.25$. Therefore 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 so that they observe the following properties:

1. Both sides have a similar average score.
2. The gap between the players with the highest and 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.

We then propose the following ideal matching system. Every team is a node, and every match is a queue with maximum 10 players. A team can be a solo player with his MMR, or a 2-5 party players with their average party MMR.

1. Once a new node comes in, detect if there is an eligible queue for it (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 to 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, otherwise, keep waiting.
6. Every 30 seconds, check if there are new nodes added into the waiting queue.
7. If yes, then keep waiting.
8. Otherwise, expand the condition to a larger acceptable score difference based on the Elo rating system.
9. Repeat from 7.
10. 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 10.

6.2 An improved scoring method for the matching system

The rank system is suitable for most competitive games, in real life as well as in e-sports. However for MOBA games 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.

What if all the players only pick certain strong heroes to play all the time? In this situation the balance of the game will be destroyed, together with the player experience. On other hand, some heroes are meant to be harder to get started than most others; if the designer doesn't encourage players to start

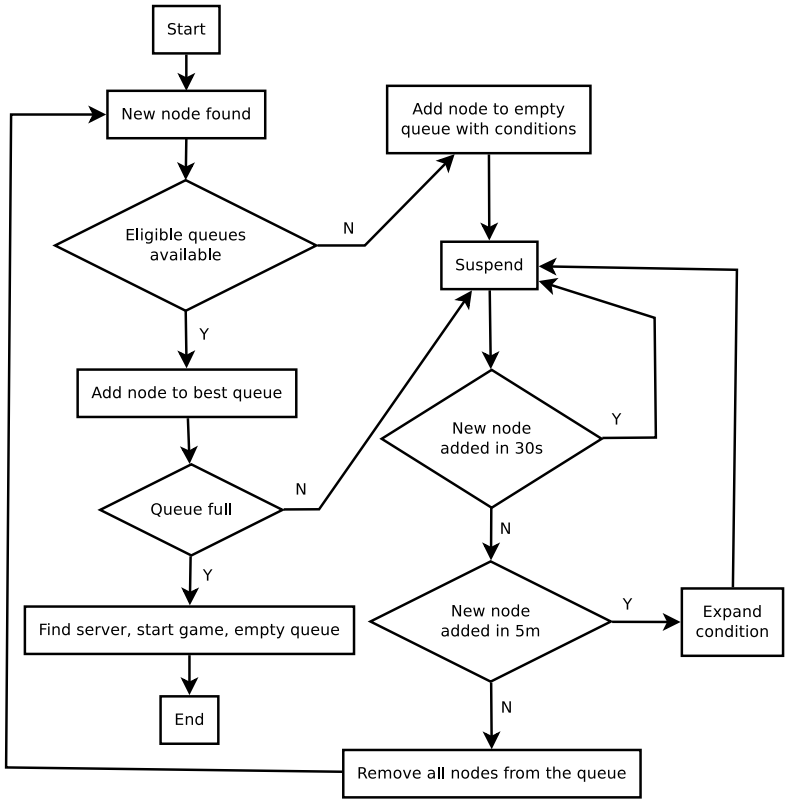

Figure 10: Flow chart for the ideal matching system.

with them, then the game will lack variety. 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 *hero rank*. This rank reflects the players' level while playing a certain hero, and is similar to the assessment of heroes in Equation (1). We establish it as $H = (\sum_{i=1}^{n} K_i \times E_i)/n$. H is the rank score for a certain hero of a player, E_i ranges over 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 are the weights for each type of data, which is the same concept as the weights in Equation (1) (different weights between heroes' types reflect how important the respective ability is for a certain hero), and n is the number of games the player has played with the respective hero. n is effective only when it is greater than a certain, small but not too

small a number (such as 10), to make sure the player is familiar enough to this hero for a precise and stable evaluation; this is the same as the calibration of the DOTA2 rank system.

We then compute the Hero-Pool value $T = (\sum_{i=1}^{n} H_i \times U_i)/\sum_{i=1}^{n} U_i$ 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 it is not realistic to assume that every single player can operate every single hero, so n should be set between 30 and 50, meaning that the Hero-Pool value only considers a player's highest top 30 to 50 unique heroes, encouraging players to play as many heroes as they reasonably can.

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

$$O = T \times k + S \tag{2}$$

where 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 DOTA2. 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 to find, for an overall better balanced environment.

The improved matching system developed above is a bonus on top of the evaluation system (the most important contribution of this paper). Indeed, the evaluation system makes it possible to assess players' ability using different heroes. 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 given by Equation (2) 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 of all the heroes. This is all a virtuous circle.

7 Conclusions

We analyzed the importance of balance and the way to achieve it in MOBA, based on DOTA2 as an outstanding representative. We studied ways of improving balance in DOTA2, both on data and rank/matching system. 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 professional tournaments. Thousands of 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.

We developed a new method which consists of data collection, cluster analysis and neural network classification to quantify the 113 unique heroes in DOTA2, and to measure 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 whether a certain hero is balanced. Further updates can then target the unbalanced factors.

Although applied on limited data (30 days worth of logs, only three representative types of data, some inaccurate assumptions), this evaluation is still feasible and effective, especially with more time and data support. This method would also play a better and more important role in newborn MOBA games than in a mature game that is already balanced such as DOTA2.

Based on the Elo rating system, we also designed an ideal matching system for DOTA2, together with an improvement based on the hero evaluation system for a better balanced environment and more data supply. This is the second contribution of this paper.

An immediate continuation of this work would be to investigate how to accurately calculate all the weights from Equation (1) for each type of data. This paper 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 and new version sets). We can then train the neural network with the old version set as "unbalanced" and new version set as "balanced"; note that this method requires more time and data collection. With enough training and our adjustment of the weights (ideally thought 3–4 patches), we believe that this system can become a standard.

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

References

[1] K. Conley, D. Perry, *How does he saw me? A recommendation engine for picking heroes in Dota 2*, Technical report, Stanford University, 2013. ⇒188

[2] J. Funk, MOBA, DOTA, ARTs: A brief introduction to gaming's biggest, most impenetrable genre, *Polygon*, December 2013. ⇒183

[3] M. van Gerven, Computational foundations of natural intelligence, *Frontiers in Computational Neuroscience*, **11** (2017), 112. ⇒187

[4] S. Huijun, W. Xinhua, Determination of the weight of evaluation indexes with artificial neural network method, *Journal of Shandong University of Science and Technology*, 20:84, 2001. ⇒195

[5] F. Johansson, J. Wikstrom, *Result prediction by mining replays in DOTA2*. Master's thesis, Blekinge Institute of Technology, Faculty of Computing, 2015. ⇒189

[6] P. Kannan, *Elo percentage expectancy table* ⇒198, 204

[7] H. P. Kriegel, M. Schubert, A. Züfle, Managing and mining multiplayer online games, *Advances in Spatial and Temporal Databases (SSTD 2011)*, *Lecture Notes in Computer Science*, **6849**, (2011), 441–444. ⇒187

[8] J. MacQueen. Some methods for classification and analysis of multivariate observations, *Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1967, 281–297. ⇒187

[9] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, *Cluster: Cluster analysis basics and extensions*, 2016, R package version 2.0.4. ⇒191

[10] R. L. D. Mandryk, D. S. Maranan, False prophets: Exploring hybrid board/video games, *CHI: Conference on Human Factors in Computing Systems*, Minneapolis, Minnesota, 2002, pp. 640—641. ⇒183

[11] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, *Nature*, **323** (1986), 533–536. ⇒187

[12] B. G. Weber and M. Mateas. A data mining approach to strategy prediction. In *Proc. Computational Intelligence and Games*, pages 140–147, 2009. ⇒189

[13] Roit Games, *League of legends matchmaking explained*, 2009. ⇒188

[14] Valve Corporation, The international DOTA2 championships official website. ⇒186

A Elo percentage expectancy table

Based on the Elo rating and on $D = R_B - R_A$, E_A and E_B can be easily completed from the Percentage Expectancy Table [6] (Table 1). Further score differences can be obtained from Table 2 with the results in Table 1 as the median.

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.

SD = Score difference, EH = Expected score rate for high scorers, EL = Expected score rate for low scorers											
SD	EH	EL	SD	EH	EL	SD	EH	EL	SD	EH	EL
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	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

Received: May 18, 2020 • Revised: September 15, 2020