Evaluating the Stability of the Smogon Tier List for Competitive Pokémon Battling

Nathan Arnold, Nicholas Fluty, Ryan D. Flores, Judy Goldsmith and Brent Harrison

Abstract

Tier lists are often used to describe the relative power of different competitive game elements. These are used so that players can evaluate the relative power of certain aspects of a competitive game and act accordingly. In the game Pokémon, each Pokémon is assigned into a tier based on its power, performance in tournament play, and potential synergy with other Pokémon it could team with. These tier lists, however, are typically designed based on observation, meaning that their quality could suffer.

In this work, we treat tier lists as *coalition formation games*. By doing this, we can leverage algorithms designed to find *stable coalitions*. In terms of tier lists, this would mean that each Pokémon would be in the correct tier and have no desire to move to a higher or lower tier. To evaluate this, we examine the Smogon tier list for Generation 1 Pokémon to determine its stability.

Keywords

Coalition formation games, Artificial intelligence, Machine learning, Stability

1. Introduction

In competitive games, *tiers* are used to refer to the relative power of some element of the game compared to others. For a first-person shooter, for example, certain more powerful weapons may be considered "higher tier" than less powerful weapons. Similar concepts can be applied to other game genres as well. In this paper, we examine how tiers are assigned in the game, Pokémon. Specifically, we are looking to examine whether or not established tier lists in Pokémon are *stable*. We computed expected win rates for 1v1 battles, and used those values to examine a tier list on a popular fan site. Doing so challenged assumptions we had made about how the tier lists on a popular fan site, Smogon, were constructed, and pointed to differences between individual battle prowess and utility of individuals in teams. Nonetheless, the Smogon tier list had very few significant instabilities, indicating that there's a strong correlation between individual win rates and usefulness in teams of Pokémon.

Pokémon is a game in which users construct a team of six combatants, the titular Pokémon, to do battle against an opponent's team of six Pokémon. For this game, the player community often designs tier lists to rank the relative strength of each Pokémon, which primarily serves as a reference tool for team construction. These tier lists take into account various factors, including but not limited to the Pokémon's stats, their observed usage in tournaments, how well they synergize with other Pokémon, and how well they perform against other commonly used Pokémon. Ultimately, these tier lists aim to reflect the usefulness of a given Pokémon when played on a team that maximizes its potential against teams that one would expect to see used competitively, where each team consists strictly of Pokémon in or below the assigned tier. The team-based nature of this game makes examining the properties of these tier lists interesting compared to other games, such as fighting games, where tiers are assigned based only on each character's ability to defeat any other character in an isolated environment. This paper highlights the difference between these two approaches to tier assignments.

For a tier list to be useful, one would expect that all members of a tier are correctly represented in terms of their power level. Formally describing such a tier list, however, can be challenging. To do so, we choose to frame the formation of a tier list as a coalition formation game. A coalition formation game is a game in which agents form or are assigned to coalitions with other agents in order to maximize some measure of utility. For tier list formulation, the coalition is a tier, and the utility is some measure of win probability against members of their current tier and below.

If we formulate tier list formulation as a coalition formation game, then a useful tier list is one that is *stable*. The notion of stability is central to coalition formation work [1]. The key idea is that agents that are assigned to a stable coalition structure have no motivation to change to another coalition.

There are many notions extant for stability in coalition formation games. They differ in what it means for one or more agents to have motivation to *block* a coalition structure — to destabilize it. One motivation for Pokémon to change tiers would be that their win probabilities, computed over their own and lower tiers, would be higher from another tier. In the work presented here, motivation to change tiers requires not only that the individual Pokémon's win probability (over opponents in its own and lower tiers) goes up, but that no other Pokémon's win probability goes down on account of that move. Thus, the moves that are allowed here appear to be win-win (so to speak).

In this paper, we investigate how stability can be used to describe the quality of tier lists for the game, Pokémon. As a case study, we specifically investigate the Smogon tier lists for generation 1. Smogon is a competitive Pokémon community which hosts tournaments, resources on competitive battling, and more. For each generation of Pokémon, Smogon constructs tier lists based on Pokémon stats, tournament performance, and various other factors. These tier lists are highly regarded in the competitive Pokémon community, which makes them an ideal candidate to analyze in terms of their stability.

The remainder of this paper is organized as follows. First, we present a definitions of coalition formation games and stability that we will use for the rest of the paper. We then provide more details on the Smogon tier lists that we will be using in this paper. We then outline how we choose to model the utility of a coalition. Next, we describe our experiments in modeling the stability of the Smogon tier list and the results of those experiments. Finally, we discuss factors that might explain the difference between tiers based on 1v1 battles and tiers generated by the Smogon community

11th Experimental Artificial Intelligence in Games Workshop, November 19, 2024, Lexington, Kentucky, USA.

based on the Pokémons' performance in 6v6 battles.

2. Coalition Formation Games

Coalition formation games are a mathematical formalism for various ways to partition agents into coalitions in order either to optimize utility or to provide *stability*, as explained below. They are used as a formalism in the businesss world, the world of communication networks [2], for multi-robot team formation [3], and for modeling good partitioning of gamers or game agents into disjoint groups.

More formally, a *coalition formation game (CFG)* consists of a set of agents that are to be partitioned into coalitions according to their preferences over partitions. This is considered a *cooperative game,* because the agents are not assumed to be in competition with each other. One well-studied family of CFGs are the *hedonic games* [4], where players' preferences concern only their own coalition. The CFG of interest here is not hedonic: agents have preferences over their own and a subset of other coalitions, known as *tiers.*

Coalition formation games have a rich history of applications to gaming, see, e.g., [5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. In most of these cases, the goal is to divide human players into groups for the purpose of gaming. For instance, Roles and Teams Hedonic Games allow League of Legends players to specify both the role(s) or champions they prefer and the team compositions [5]. Anchored Team Formation Games are focused on interpersonal preferences in settings with a game manager or other anchor [10]. There, it is assumed that preferences include information about whether another gamer will be GM or player, and allow for preferences over pairs of players that are unpleasant together, or achieve synergy together.

Unlike the previously mentioned CFGs, *Tiered Coalition Formation Games (TCFGs)* [9] were inspired by the groupings of Pokémon into tiers on the Smogon website. We initially assumed — going back to the original paper on TCFGs [9] — that the tiers were based on the ability of the Pokémon in that tier to defeat other individual Pokémon in their own and lower tiers. The Pokémon do not, in most of the work on TCFGs (see also [15, 14]), express preferences, per se. Rather, the value they have for a partition is a function of the likelihoods that they will beat the other Pokémon in their own and "lower" tiers.

Definition 1. *A* tiered coalition formation game *[9] is a coalition formation game $(N, \succeq)$, where $N = \{a_1, a_2, ..., a_n\}$; an outcome, or* tier list*, is a totally ordered spanning set of disjoint coalitions $[T_1, T_2, ..., T_k]$; $Seen(a_i, T)$ for tier list T denotes the set of all agents that are in the same tier as a_i or are in a lower (prior) tier; and the preferences for each agent a_i in each possible tier list T are determined solely by the set of agents a_i "sees" in T.*

Since we are investigating the stability of the Smogon tier list, we can expect a fixed, finite number of tiers. This enables us to utilize a k-tiered coalition formation game to model tier list formulation.

Definition 2. *A* k-tiered coalition formation game *[16] is a coalition formation game $(N, \succeq, k)$, where $N = \{a_1, a_2, ..., a_n\}$, $k \leq n$ is in $\mathbb{N}$ and an outcome is a spanning, totally ordered tier list of exactly k disjoint nonempty coalitions, and $Seen(a_i, T)$ for tier list T is defined and determines preferences as in a standard tiered coalition formation game $(N, \succeq)$.*

There are many potential solution concepts for coalition formation games, such as optimization over agents' satisfaction (here, measured for each agent as the average win values over all seen agents). Even this must be modified with some form of optimization, such as total satisfaction, or maximin satisfaction [17], or perhaps Pareto optimality [18]. However, we are interested *stability* here, particularly Nash stability: the notion that, given a K-tier list T, each agent is in the tier that is best for them.

Definition 3. *A* Nash stable k-tier list *is a k-tier list T such that there is no agent i in a tier with other agents that can move to another existing tier such that it gains utility, and there is no agent j in a tier by itself that can move its tier to a different position in the order such that it gains utility. Equivalently, T is Nash stable if there is no k-tier list T' that differs from T in the tier of one agent a_i such that $Seen(a_i, T') \succ_i Seen(a_i, T)$.*

3. The Smogon Tier List

The following is a brief overview of the Smogon tier list on the fully evolved Pokémon in RBY, as of 2024.

The list divides the agents into eight categorizations: in order from lowest to highest, they are ZU, ZUBL, PU, NU, UU, UUBL, OU, and Uber. However, in the context of RBY, the labels ZUBL, UUBL, and Uber are not considered tiers; rather, these are Pokémon who would otherwise belong to the tier immediately below, but are considered too powerful for balanced matches in that tier [19]. It is computationally less accurate to describe the Smogon list as an 8-tier list than it is to regard each of these three tiers as a subcategory of the one below it. We therefore list the membership of each tier combining ZU with ZUBL, UU with UUBL, and OU with Uber. This tier list is shown in Table 1, with the names of members of *ZUBL, UUBL*, and *Uber* italicized.

ZU+ZUBL	Butterfree, Flareon, Hitmonlee, Lickitung, Magneton, Marowak, Muk, Onix, Parasect, Pidgeot, Primeape, Sandslash, Weezing, Beedrill, Ditto, Farfetch'd, Golbat, Hitmonchan, *Arbok*
PU	Arcanine, Fearow, Machamp, Magmar, Nidoqueen, Pinsir, Porygon, Rapidash, Scyther, Seaking, Vileplume
NU	Aerodactyl, Blastoise, Charizard, Golduck, Kabutops, Kingler, Moltres, Mr. Mime, Nidoking, Poliwrath, Raticate, Seadra, Venomoth, Venusaur, Wigglytuff
UU+UUBL	Articuno, Clefable, Dewgong, Dodrio, Dragonite, Dugtrio, Electabuzz, Electrode, Golem, Gyarados, Kangaskhan, Ninetales, Omastar, Persian, Raichu, Tangela, Tentacruel, Vaporeon, *Hypno, Lapras*
OU+Uber	Alakazam, Chansey, Cloyster, Exeggutor, Gengar, Jolteon, Jynx, Rhydon, Slowbro, Snorlax, Starmie, Tauros Victreebel, Zapdos, *Mew, Mewtwo*

Table 1
Smogon's tier list on RBY fully evolved Pokémon, 2024

4. Modeling Utility

In order to analyze the stability of a tier list, we need to model the utility of an agent based on an assigned tier. In a TCFG, agent a_i gains utility based on the set of agents in its own tier and in all lower tiers. For Pokémon battles, this set of seen agents represent what Pokémon it could possibly have on its team or play against. This means that the utility of seeing each agent would ideally reflect the advantage or disadvantage of seeing the Pokémon.

However, this change in advantage cannot be perfectly represented in terms of pairings of Pokémon, evidenced by the fact that a combination of three particular Pokémon on a team might yield a notable advantage. However, any other approach would likely involve a lot more computation and make the data harder to analyze.

To simplify this, we choose to examine 1v1 battles to determine the utility of each pair, using a measure based on pairwise matchup outcomes. We give Siler's original definition, and Arnold's probabilistic variant.

Definition 4. *A deterministic matchup-oriented preference framework [9] is a framework in which an agent's utility is equal to the number of favorable matchups minus the number of unfavorable matchups among its seen agents. In other words, an agent derives a utility of* 1 *from each seen agent it is projected to win against, and a utility of* -1 *from each seen agent it is projected to lose against.*

We note that Pokémon battles are not determined, even given identical timings, due to the probabilities associated with attacks.

Definition 5. *A probabilistic matchup-oriented preference framework [16] is a preference framework in which, for agents* i *and* j *such that* i *has a probability* p *of winning against* j, $0 \leq p \leq 1$, $Win[i, j] := 2(p - 0.5)$.

Using reliable open source tools (see Section 5.1.2), we are able to quickly and accurately simulate 1v1 battles, and all state information can be observed and modified in a way that allows for the implementation of a large number of machine learning techniques. In order to collect win rates for all 1v1 battle combinations for the 81 fully evolved Pokémon on the Smogon tier list (excluding agents playing against themselves), we would simulate the 3240 unique battle configurations, where each agent learns a policy to optimize its chances of winning. The resulting win percentages using the policies can be linearly translated into the value of each agent seeing each other agent, where 0 represents a 50% chance of each Pokémon winning the battle. These win rates were then used to construct a Nash stable tier list, using the work of [14]. They were also used to assess the stability of Smogon's tier list.

5. Experimentation

In this section, we describe in detail the processes we used first to determine win percentages through simulated battles and second to test the stability of the Smogon tier list.

5.1. Calculating Win Percentages

5.1.1. Pokémon Used

We set the stat values for each Pokémon to the highest value that the games allow, as is done by default in most tournaments. To avoid overcomplicating the nature of our experiments, we determined that each Pokémon would have a single moveset for all battles. This is a reasonable representation of competitive play since players do not know in advance what Pokémon their opponent will use and have to configure their movesets in advance. For each Pokémon, we used the most common moveset according to usage statistics on Pokémon Showdown [20].

While variations from this most common moveset for a given Pokémon may be employed for counter-selection or team cohesion, we suspected that this decision would equate to each agent playing optimally against the sum of other agents, within the space of its available actions. As later described in our Results section, we found that some of the movesets used by the Pokémon in our experiment seemed less optimized for 1v1 matches than others. Optimizing the movesets ourselves could have prevented this, but we preferred not to stray from normal usage or risk manipulating our results with any kind of bias.

5.1.2. Software Used

To simulate Pokémon battles, we used the pkmn engine, a Pokémon battle simulation engine optimized for performance in larger scale projects [21]. This open source tool accurately implements battles as done in the original game code and the popular simulator Pokémon Showdown. Pokémon Showdown is sponsored and endorsed by Smogon for competitive battles, as it provides practical implementations of battles for all Pokémon generations and a practical interface for playing online. The pkmn engine currently only fully supports the first generation of Pokémon, but it is able to run faster than Showdown while having less unneeded overhead.

We also used and credit another open source project, Wrapsire, which provides a C++ interface for the compiled library produced by the pkmn engine [22]. We opted to use C++ for this project because of its advantages for memory management, multi-threading, and compiler optimizations.

5.1.3. Monte Carlo Tree Search

Pokémon is a stochastic turn-based game, where a turn is initiated by both players selecting an action independently but concurrently. In order to collect win percentages that are consistent with the skill expected of competitive players, action selection must be mindful of what gives the highest chances of winning. We used Monte Carlo Tree Search (MCTS) to do this.

We chose to use MCTS primarily because of its ability to handle uncertainty and large state spaces. Additionally, we expected that the average number of turns for a 1v1 battle would be under six. This makes it very easy to repeatedly simulate games to the end state and back-propagate values until one of four actions is seen as the most reliable.

Pokémon uses a random seed as a factor in many different calculations, most notably in standard damage calculation and secondary move effects. Because of this, it is possible that a turn initiated from a given game state by given actions has hundreds of possible unique resulting game states depending on the seed. This is not problematic for MCTS because it naturally weighs the value of an action based on the frequency that different states result from it, and these states similarly develop better heuristics for move selection as the search continues.

The primary problem we faced with implementing MCTS for Pokémon battles was with determining how to handle concurrent action selection. Minimax trees are commonly used for turn-based games like chess, but these assume that players alternate turns and know what the enemy previously selected. This assumption contradicts the nature of competitive Pokémon battles, as it almost always involves players intentionally being unpredictable. This is due to a natural rock-paper-scissors-like relationship among strategies. For example, recovery may counter gradual damage, while applying buffs or statuses may counter recovery, and the right damaging move may yield an enemy's buffs ineffective or make them lose before they gain the longer-term benefits.

To address this issue, we randomly determine, at every simulation of a turn, who will select their action first, and who will pick based on that selected action. While this does not perfectly represent the distribution of move combinations used in competitive games, it effectively balances the scenarios where a player either picks their safest action or correctly predicts that their opponent is picking their safest action. This gives an equal advantage to each player, while also producing more consistency in game outcomes, which is helpful for assuring the integrity of our results. It is also significantly less computationally expensive than developing stochastic policies, which makes it easily applicable at all frequently visited states during a search.

Other design notes had a lesser effect on the results. Primarily, since total health points and amounts of damage dealt are rather inflated, we bucketed states together when counting visits and related information, ignoring the bottom five bits of both Pokémon's health points when identifying a state during a search. This allows for much quicker convergence in almost all matchups and is easily modifiable in our code.

5.1.4. Simulating All Battles

To get reasonable precision on our collected win percentages, we aimed for 1,000 battles of each matchup, using MCTS at each turn. Noting that states are often visited many times over the course of this many battles, including the guaranteed start state, we stored actions returned by the algorithm for use in the remaining battles.

We did extra iterations of MCTS for the first turn, as the result would be used in all 1,000 battles. We did fewer iterations as the turn count increases, as these states were expected to have smaller search spaces and be visited in fewer of the battles. For each pair, we did 5,000,000 iterations for the first turn, 500,000 for the second turn, and each following turn had half the iterations of the previous turn, to a minimum of 10,000 iterations. This also helped combat the problem of some battles carrying on particularly long due to recovery moves.

Unfortunately, some matchups required more memory than expected to maintain search data across this many battles. To keep run times reasonable while running multiple battles in different threads, we terminated a small portion of the matchups while their battle count was between 100 and 1,000. We considered 100 battles to be a sufficient number that would not compromise the integrity of our results, but we performed 1,000 battles for a large majority of the matchups.

The code iterated through all unique combinations of the 81 fully evolved Pokémon and output the number of times the Pokémon coming first alphabetically won. Because ties only occurred in a small number of matches, we handled them by simply counting them as half of a win, which can be seen as the match not favoring either side.

6. Results

The full 81×81 Win matrix is too large to be shown here. We used this resulting matrix to consider the 5-TCFG with the Smogon tier list as a partition.

First, in Table 2, we show the Nash deviations from the Smogon list, assuming that the agents' power network is accurately represented by our produced Win matrix. Non-deviating agents are shown in black text. Agents deviating to an adjacent tier for a utility improvement not exceeding 5 are in gray text. Agents deviating to a non-adjacent tier are in cyan text, agents whose deviation would improve their utility by more than 5 are in orange text, and agents with both properties are in magenta text.

ZU+ZUBL	Butterfree, Flareon, Hitmonlee, Lickitung, **Magneton**, Marowak, Muk, Onix, **Parasect**, Pidgeot, Primeape, Sandslash, Weezing, Beedrill, Ditto, Farfetch'd, Golbat, Hitmonchan, *Arbok*
PU	Arcanine, Fearow, Machamp, Magmar, Nidoqueen, Pinsir, Porygon, Rapidash, Scyther, Seaking, **Vileplume**
NU	**Aerodactyl**, Blastoise, **Charizard**, Golduck, Kabutops, Kingler, Moltres, Mr. Mime, Nidoking, **Poliwrath**, **Raticate**, Seadra, **Venomoth**, **Venusaur**, **Wigglytuff**
UU+UUBL	Articuno, Clefable, Dewgong, **Dodrio**, Dragonite, **Dugtrio**, Electabuzz, Electrode, **Golem**, Gyarados, Kangaskhan, **Ninetales**, **Omastar**, Persian, Raichu, **Tangela**, Tentacruel, Vaporeon, *Hypno, Lapras*
OU+Uber	Alakazam, Chansey, Cloyster, Exeggutor, Gengar, Jolteon, Jynx, **Rhydon**, Slowbro, Snorlax, Starmie,Tauros, Victreebel, Zapdos, *Mew*, *Mewtwo*

Table 2
Smogon's list with Nash deviations highlighted

This tier list is not Nash stable. For example, Hypno's utility would increase by 3.74 if it moved to the OU tier. This increase means that Hypno maintains a higher chance of winning than losing against the Pokémon of the OU tier, so it would like to move into that tier. To put this number into perspective, a utility increase of 16 would result from a 100% win rate against the new 16 OU Pokémon it sees. The increase of 3.74 indicates an averaged 61.7% win rate against the 16.

Arnold et al. [13] introduced a stronger notion of stability, socially conscious stability, in which an agent may only move if it does not decrease any other agent's satisfaction with the tier list. In fact, Hypno's move would have the permission of all affected agents, so Smogon's tier list is also not socially consciously stable based on the results of our 1v1 battles.

7. Discussion

Let us assume, for the moment, that the tier list on these agents held by Smogon is the most accurate list on these agents in this setting, and that the matchup probabilities produced by our method are accurate. Then, while we have seen that this list has several potential deviations under matchup-oriented preferences, approximately half of the agents would not deviate, and many of the rest only deviate to an adjacent tier for a relatively small gain in utility. This would indicate that matchup-oriented preferences are able to capture *most* of the complexities of the Pokémon game.

There are a number of complications and confounding variables which may have influenced these results. What follows is an enumeration and discussion of factors that we are aware of.

Team composition. Matchup-oriented preferences are concerned only with the outcome of 1v1 matchups. However, the underlying competitive environment of RBY assumes teams of 6 Pokémon. The use of a Pokémon within the context of a team with synergy is critical when considering its strength in competitive games. For example, in our produced data, the agents Rhydon, Golem, and Dugtrio all perform significantly worse than usage data would suggest. However, these three agents are Ground-type Pokémon, which have several common weaknesses (and therefore many unfavorable matchups) but can play an important role on a team, such as covering an Electric-type weakness.

Counter-selection. If a tier contains some dominating agent, then agents that have a favorable matchup against the dominating agent are elevated in favorability. For example, Articuno is a very commonly used Pokémon in UU matches [19], and its abundance may increase the frequency of use in that tier for Ninetales, who according to our data wins against Articuno 83.1% of the time. However, Ninetales's performance against other Pokémon in UU causes it to prefer demoting itself to the lower tier PU for a gain of 0.684 utility. Counter-selection has already been studied in the context of Pokémon Go [12], and there is a research avenue to examine counter-selection as a temporal element of TCFGs.

Final state of winner. Another consequence of Pokémon being a match between teams of six is that an agent's state after defeating its opponent is significant to the course of a match. Two agents may be equally reliable at defeating the same other agent, but one may often win at low health, while the other consistently wins unscathed. In fact, the low health Pokémon could be paralyzed and thus almost useless to the team, while the full health Pokémon could have raised its stats to pose an even greater threat to the opponent's next Pokémon. Competitive teams often include at least one such Pokémon that can "sweep" the opponent's team under the right circumstances, and their movesets are usually designed around this playstyle. We note that these agents may have had their utility underrepresented by the data we collected. This similarly applies to speedy Pokémon specializing in finishing off opposing Pokémon weakened earlier in battle.

Sleep moves. Sleep moves are usually considered the best moves in the game. Smogon recognizes this and enforces a restrictive rule limiting the number of opposing Pokémon a player can put to sleep to just one at any time. Our 1v1 battles naturally do not get restricted by this rule. Additionally, Smogon only puts a few sleep-inducers into OU because each team usually just has one, making most of them irrelevant to the highest tier of play.

Changing nature of Smogon's list. While we consider Smogon's tier list on RBY a useful metric to compare against, it is not necessarily canonical. When this project began, Smogon only had three categorizations for fully-evolved Pokémon in RBY: UU, OU, and Uber. The community surrounding this game is alive and active, and the list has undergone many changes in the past several years to better accommodate the beliefs and behaviors of that community over time.

Agent flexibility. In our training, we only considered what we found to be the most common set of known moves for each agent. For some agents, such as Alakazam and Slowbro, this is sufficient, as players utilizing those agents rarely deviate from the most common moveset [19]. Others, such as Chansey and Snorlax, exhibit much greater flexibility that players can take advantage of, making them more favorable for use in a team. Further, we applied an assumption that players have full knowledge of their opponents, which ignores a potential advantage one could have by using less common moves. This partial information nature likely caused the less predictable Pokémon with various movesets to have a lower measured utility.

Considering all these issues, we are pleased that many of the agents in the Smogon tier list do not deviate or deviate by a small amount. This suggest that an individual Pokémons' battle prowess is a decent indicator of its value to a team. When we take into consideration the above ways that each Pokémon may have been disadvantaged by our data collection, we can guess even more accurately what tier Smogon puts the Pokémon in. Nonetheless, some dynamics of a Pokémon's purpose in current competitive games may only make sense in the context of the current trends.

8. Conclusion

In this paper, we investigate the stability of the Smogon tier list for competitive Pokémon battling. By casting the tier list formation problem as a coalition formation game, we are able to determine if each tier in the tier list is a stable coalition. This is important because it helps us determine if a Pokémon would achieve more utility by moving up or down a tier in the tier list.

Our results show that while many Pokémon are correctly placed in an appropriate tier, there are several that could achieve higher utility by moving up or down a tier. This means that, by considering only 1v1 battles, their power level is not being accurately described, which could result in an imbalanced and unfun competitive battling environment.

In short, stability is an interesting measure for tier lists in competitive play, but it does not capture the synergies that are possible in teams. On the other hand, the near-stability of Smogon's list with respect to 1v1 battles tells us that individual battles are still a relevant part of team formation.

References

[1] R. J. Aumann, J. H. Dreze, Cooperative games with coalition structures, International Journal of game theory 3 (1974) 217–237.

[2] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, T. Basar, Coalitional game theory for communication networks, Ieee signal processing magazine 26 (2009) 77–97.

[3] B. P. Gerkey, M. J. Matarić, A formal analysis and taxonomy of task allocation in multi-robot systems, The International journal of robotics research 23 (2004) 939–954.

[4] A. Bogomolnaia, M. O. Jackson, The stability of hedonic coalition structures, Games and Economic Behavior 38 (2002) 201–230.

[5] M. Spradling, J. Goldsmith, X. Liu, C. Dadi, Z. Li, Roles and teams hedonic game, in: International Conference on Algorithmic Decision Theory, Springer, 2013, pp. 351–362.

[6] M. Spradling, Role Based Hedonic Games, Ph.D. thesis, University of Kentucky, 2015.

[7] M. Spradling, J. Goldsmith, Stability in role based hedonic games., The International FLAIRS Conference Proceedings 28 (2015) 85–90.

[8] M. J. Spradling, Optimizing expected utility and stability in role based hedonic games, in: The Thirtieth International Flairs Conference, 2017.

[9] C. Siler, Tiered coalition formation games, The International FLAIRS Conference Proceedings 30 (2017) 210–214.

[10] J. Schlueter, C. Addington, J. Goldsmith, Anchored team formation games, in: The International FLAIRS Conference Proceedings, volume 34, 2021.

[11] J. Schlueter, Novel Hedonic Games and Stability Notions, Ph.D. thesis, University of Kentucky, 2021.

[12] D. Crane, Z. Holmes, T. T. Kosiara, M. Nickels, M. Spradling, Team counter-selection games, in: 2021 IEEE Conference on Games (CoG), IEEE, 2021, pp. 1–8.

[13] N. Arnold, S. Snider, J. Goldsmith, Socially conscious stability for tiered coalition formation games, Annals of Math and Artificial Intelligence (2024). https://link.springer.com/article/10.1007/s10472-023-09897-4.

[14] N. Arnold, Tiered Coalition Formation Game Variants, Stability, and Simulation, Ph.D. thesis, University of Kentucky, 2024.

[15] N. Arnold, J. Goldsmith, Core stability and nash stability in k-tiered coalition formation games, Under review (2024).

[16] N. Arnold, J. Goldsmith, S. Snider, Extensions to tiered coalition formation games, The International FLAIRS Conference Proceedings 35 (2022). doi:https://doi.org/10.32473/flairs.v35i.130708.

[17] N. Waxman, S. Kraus, N. Hazon, On maximizing egalitarian value in k-coalitional hedonic games, arXiv preprint arXiv:2001.10772 (2020).

[18] M. Bullinger, Pareto-optimality in cardinal hedonic games., in: AAMAS, volume 20, 2020, pp. 213–221.

[19] Smogon, RB formats, https://www.smogon.com/dex/rb/formats/, 2024.

[20] G. Luo, Pokemon Showdown, https://pokemonshowdown.com/, 2012-24. Accessed on May 6, 2024.

[21] K. Scheibelhut, libpkmn, 2021-24. URL: https://github.com/pkmn/engine.

[22] pasyg, wrapsire, 2023-24. URL: https://github.com/pasyg/wrapsire.