Extensions to Tiered Coalition Formation Games

Nathan Arnold, Judy Goldsmith, and Sarah Snider
University of Kentucky
Lexington, Kentucky, USA

Abstract

In 2017, Siler introduced Tiered Coalition Formation Games (TCFGs), inspired by the tiered organization of Pokémon characters on a fan-based website, Smogon. Siler showed that, for a natural notion of agent preferences, the Nash table tier lists were precisely the core stable tier lists, and provided a polynomial-time algorithm to find a Nash stable tier list. However, the tiers in that list had size one, which eliminated the intra-tier competitions that make tier lists meaningful. We extend Siler's definition to allow for win probabilities, examine k-tier TCFGs, and provide a heuristic algorithm for finding good k-tier partitions that allow intra-tier competition.

Introduction

In this work, we introduce a variant on Tiered Coalition Formation Games (Siler 2017), and show that we can get significantly improved results. In particular, we introduce a parameterized version of TCFGs, for which finding partitions (tier lists) with desirable properties is easy. We also allow matchups to be modeled probabilistically, and provide a heuristic algorithm for both the unconstrained and the parameterized versions of TCFGs. We define two new criteria for evaluating partitions, which we call "friendliness" and "robustness." Siler's original paper gave a P-time procedure that produced a minimally-friendly stable partition. Our experiments show that our algorithm produces consistently more friendly stable partitions.

Coalition formation games are an increasingly studied subtopic of computational social choice. Agents, usually representing actors who may wish to cooperate, form into a *partition* of totally spanning, disjoint *coalitions*. A partition may lack crucial qualities from the perspective of certain agents, and so new partitions may be formed for as long as agents remain unsatisfied.

Many of the games being studied are *hedonic games*, in which an agent's assessment of a partition is derived only from that agent's coalition. This restriction is a natural one when modeling pairing of roommates (Irving 1985) and other scenarios in which an agent is unlikely to be affected by coalitions other than its own.

In 2017, Siler introduced *tiered coalition formation games*, a non-hedonic game that totally orders its coalitions; rather than derive utility from the contents of one's own coalition, an agent considers its position in the hierarchy relative to other agents (Siler 2017). Siler was inspired by Smogon (Smogon a), a fan-run website for the Pokémon series of video games, which organizes the 898 Pokémon game pieces into a *tier list* of 11 strictly ordered *tiers*.

Notably, Pokémon game pieces have an *intransitive* power structure; Bulbasaur may be likely to win against Squirtle, and Squirtle may be likely to win against Charmander, but this does not imply Bulbasaur is likely to win against Charmander. This intransitivity makes the power structure difficult to understand at a glance for humans and computers alike. Tier lists, and by extension TCFGs, aim to represent such power structures in an easy-to-understand manner despite intransitivity. Two agents in the same tier are assumed to be approximately power equivalent, and an agent in a higher tier is assumed to have a higher overall strength than agents in lower tiers, even if some of those lower-ranking agents may be likely to win in a one-on-one matchup.

While many of the most accessible examples of intransitivity relate to video games, intransitivity has been observed in sports matchups and even in preferences on group formation among humans (May 1954) (Tversky 1969). The full range of scenarios TCFGs can be used to model may be quite extensive, and is not considered here. Instead, we address the primary topic of future inquiry identified by Siler.

Siler outlined stability concepts for TCFGs, namely Nash stability and core stability, which are often used in hedonic games, and found that for a natural notion of agent preferences, Nash stability and core stability are equivalent, and that a stable partition may be found in polynomial time. The stable partition found, however, consists solely of singleton coalitions. This was considered limiting in the context of *coalition formation*. Furthermore, Siler considered as an example the game "Rock, Paper, Scissors," a game piece selection game with three agents, writing that it would be "sensible" for these agents to be in a single tier, and that separating them into singletons is "counterintuitive" (Siler 2017). Siler concludes that the next challenge to this area of study should be the addition of new preference criteria or other conditions that will result in a more realistic tier list.

Consider Siler's source of inspiration: Smogon. When justifying the controversial creation of an eleventh tier in their tier list, the Smogon administrators referred to their philosophical justification for the very existence of tier lists (Smogon b). First is that tier lists should be robust, displaying the relative power of all agents rather than only the strongest ones. Second is that each tier should represent a useful environment, allowing players to make decisions in game piece selection and ensuring that matches are fair. In other words, tier lists should be *fun*.

Related Work

There has been substantial work defining coalition formation games and hedonic games, enumerating preference frameworks, and describing notions of stability within those preference frameworks. Bogomolnaia and Jackson (Bogomolnaia and Jackson 2002) are among the very few to distinguish hedonic coalition formation games from other classes of coalition formation game, providing side-by-side comparisons of hedonic games with conceptually similar non-hedonic counterparts, and defining stability alongside notions of fairness and optimality. Banerjee, Konishi, and Sonmez (Banerjee, Konishi, and Sonmez 2001) offer a seminal look into stabilizability of certain classes of hedonic games. Their introduction of additional preference criteria to existing games in an attempt to simplify computational load has inspired countless similar approaches in other games, including our work on TCFGs.

Intransitivity in power relationships complicates our ability to understand those relationships. Chen and Joachims (Chen and Joachims 2016) examined intransitive power relationships in matchup data for two-player competition, representing these relationships as vectors on a two-dimensional graph. Saarinen, Tovey, and Goldsmith also investigated dominance in intransitive round-robin tournaments (Saarinen, Goldsmith, and Tovey 2015). Tversky (Tversky 1969) and May (May 1954) independently offer empirical evidence that points to *humans* having intransitive preferences in contexts of hiring and marriage problems, respectively.

There are other coalition formation games based on gaming environments. Spradling, Goldsmith, Liu, Dadi, and Li (Spradling et al. 2013) introduced Roles and Teams Hedonic Games, a class of hedonic coalition formation game inspired by the video game League of Legends. Role-Based Hedonic Games (Spradling 2015) (Spradling 2017) (Tsogbadrakh and Spradling 2019) were introduced based on RTHGs. Anchored team formation games (Schlueter, Addington, and Goldsmith 2021) were inspired by partitioning gamers into groups for tabletop role-playing games, and the authors introduce heuristics to find nearly-stable partitions.

Preliminaries

In this section, we present some of the basics of coalition formation games, and results by Siler concerning TCFGs, inspired by the fan-run site, Smogon, for Pokémon. The definition of the set of *seen agents* was inspired by the tier structure used by Smogon, where an agent's suitability for a tier is determined by its matchups against other agents in that same and lower tiers (Smogon a).

Definition 1. *(Siler 2017) A* tiered coalition formation game *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 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*.*

Note that $\forall a_i \in N$ and tier lists T and T',

$$(Seen(a_i, T) = Seen(a_i, T')) \implies T \sim_i T'.$$

Definition 2. *(Siler 2017) A* Nash stable tier list *is a tier list* T *such that there does not exist an agent* a_i *that can find a more-preferred tier list by moving. Equivalently,* T *is Nash stable if there is no tier list* T' *that differs from* T *in the tier of one agent* a_i *such that* $Seen(a_i, T') \succ_i Seen(a_i, T)$*.*

Definition 3. *(Siler 2017) A* core stable tier list *is a tier list* T *such that there exists no nonempty subset of agents* B *that could form a new tier together anywhere in the hierarchy such that for the resulting tier list* T'*,* $Seen(a_i, T') \succ_i Seen(a_i, T)$ *for all* $a_i \in B$*.*

Siler showed that a core stable tier list is always Nash stable. However, a stable tier list is not guaranteed to exist in a general case, and determining the existence of a core stable tier list is NP-hard. Siler goes on to introduce preference criteria that strengthen stabilizability.

Definition 4. *(Siler 2017) For given agents* a *and* b*, if* a *is likely to defeat* b*, then* b *is a* good matchup *for* a*, and* a *is a* bad matchup *for* b*.*

Note that Siler's work treats matchups as determined, although the language used refers to "favorability" of matchups. We introduce notation $Win[i, j]$ in Definition 7, which extends Siler's matchups to the probabilistic case.

Definition 5. *(Siler 2017) A* matchup-oriented preference representation *is one in which the scalar utility of an agent* i *derived from* $Seen(i, T)$ *is equal to* $\sum_{j \in Seen(i)} Win(i, j)$ *for an antisymmetric Win.*

Siler then showed, for matchup oriented preferences, a tier list is Nash stable iff it is core stable.

Theorem 1. *(Siler 2017) Under matchup-oriented preferences, a Nash stable and core stable tier list is guaranteed to exist. Furthermore, it can be found in polynomial time.*

Siler defines a search that iteratively sorts agents until a stable tier list is reached, in $O(n^4)$ time. We refer to that process as "Siler sort," which produces "Siler's stable tier list."

In a *hedonic game* (where an agent's preference references only their own coalition), a partition is *contractually individually stable* if no agent can improve their utility by changing coalitions while leaving the agents in the abandoned and receiving coalitions with at least as high utility as before the change. In a hedonic game, such a move necessarily raises the social utility, because the utilities of agents outside the abandoned and receiving coalitions are not affected. However, when an agent in a TCFG moves to another level, the utilities of the agents in intermediate levels are also affected. Thus, we introduce *socially conscious stability.*

Definition 6. *Given an instance of a TCFG, a tier list T is* socially consciously stable *if no agent can improve their own utility by moving to another tier without decreasing the total utility among other agents.*

Observation 1. *Any maximal-utility tier list is, by necessity, socially consciously stable.*

Siler's matchups were deterministic, unlike those in Pokémon, sports environments, and many other real-world competitions. We extend Siler's model to *probabilistic* matchups.

Definition 7. *We define* probabilistic matchup-oriented preferences *as matchup-oriented preferences where the preference relationships are derived from a probabilistic win matrix, here denoted P where $P[i,j] \in [0,1]$ and $P[i,j] = 1 - P[j,i]$.*

We define Win under probabilistic preferences as

$$Win[i,j] = 2(P[i,j] - 0.5).$$

The total utility of an agent in a tier list is still $\sum_{j \in Seen(i)} Win(i,j)$.

This transformation from probability to utility has two purposes. First, it guarantees that the relationship between two agents is antisymmetric; the utility i gains from seeing j is the opposite of the utility j gains from seeing i, for any two agents i and j. Second, the utility i gains from seeing j is in the range [-1, 1] under this model. This is the same range as for Siler's deterministic matchup preferences (Siler 2017), allowing direct comparison.

Because Siler's proofs of the properties of matchup-oriented preferences depend solely on the antisymmetric nature of the Win relationship and not on the exact values of Win, the key properties of this preference framework, including Siler's procedure to find a stable tier list of singletons, hold under probabilistic preferences. In fact, Siler's proofs of these properties hold for any simple, antisymmetric Win.

We adapt the notion of a tier list's 'fun' held by Smogon into an empirical measurement, which we call 'friendliness,' as well as formalizing their notion of 'robustness'(Smogon b).

Definition 8. *The* friendliness *of a tier list T is equal to $|F|/(|F| + |S|)$, where F is the set of coalitions $T_i \in T$ s.t. $|T_i| > 1$, and $|S|$ is the set of coalitions $T_j \in T$ s.t. $|T_j| = 1$.*

Informally, the *friendliness* of a tier list is the proportion of tiers it contains that are not singletons. This measurement is 0 for a tier list of singletons, and 1 for a tier list of non-singletons.

Definition 9. *Let a be an agent in tier list T, and let T_i be a tier such that $a \in T_i$. The* fitness *of a to T_i is defined as $\sum_{b \in T_i/a} Win[a,b]$. The* robustness *of T is the minimum fitness of all agents in T to those agents' tiers.*

Rather than being a summary statistic of an entire tier list (as are utility and friendliness), robustness is an observation of a minimum, relating to the agent least fit to its current tier. Preferences of agents are not determined solely by the composition of that agent's tier, but if a tier list is both robust and friendly, each of its tiers is known to contain agents that are approximately equivalent in power, an important property according to both the philosophical justification of tier lists per Smogon (Smogon b) and the challenge set by Siler to value placing Rock, Paper, and Scissors in a single tier (Siler 2017).

Observe that this measure of robustness is always nonpositive; it is impossible for all agents to have a positive sum-of-win against their peers.

Observation 2. *A tier list of singletons is minimally friendly ($friendly = 0$) and maximally robust ($robust = 0$).*

In addition to this observation, a stable tier list of singletons generally has high total utility, and therefore a high ranking in two of three evaluative criteria. Our goal is to find a tier list with high utility, high friendliness, and high robustness.

k-Tier Lists

We introduce a new variation on tiered coalition formation games, the TCFG with fixed tier count, inspired again by Smogon's model (Smogon a).

Definition 10. *A* tiered coalition formation game with k-fixed tier count *(k-TCFG) 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 tier list of exactly k totally ordered disjoint coalitions, and $Seen(a_i, T)$ for tier list T is defined and determines preferences as in a standard tiered coalition formation game $(N, \succeq)$.*

We observe that a tier list T is in a given $(N, \succeq, k)$ if and only if T has exactly k tiers and is in $(N, \succeq)$.

Theorem 2. *If a tier list T with k tiers is Nash stable on a given standard TCFG $(N, \succeq)$, it is Nash stable on the k-TCFG $(N, \succeq, k)$.*

Proof. Suppose that T with k tiers is Nash stable on a (standard) TCFG but not Nash stable on the corresponding k-tiered coalition formation game. Then $\exists i$ such that a_i that can move from T without changing the number of tiers s.t. for the resulting tier list T', $Seen(a_i, T') \succeq Seen(a_i, T)$. However, the preferences for a_i on the k-TCFG and the standard TCFG are the same. $\Rightarrow\Leftarrow$ □

Similarly, a core stable tier list on a standard TCFG is core stable on its corresponding k-TCFG. We omit the proof here.

Note that the converse does not hold. For example, in a two-agent matchup-oriented instance in which one agent wins over the other, a single tier of two agents is not Nash stable in the standard TCFG, but **is** Nash stable in the 1-TCFG.

Let Game 1 be a 2-TCFG on agents a_1, a_2, a_3, a_4, with probabilistic matchup-oriented preferences shown in Table 1, where row a_i, column a_j denotes $Win[a_i, a_j]$.

Claim 1. *Game 1 has no Nash stable partitions.*

Proof. There are fourteen partitions of these four agents into nonempty lower tier T_1 and nonempty higher tier T_2. If $a_4 \in T_2$ and at least one other agent is also in T_2, a_4 can profitably

Table 1: GAME 1

	a_1	a_2	a_3	a_4
a_1		0.9	-0.1	1.0
a_2	-0.9		0.1	1.0
a_3	0.1	-0.1		1.0
a_4	-1.0	-1.0	-1.0	

move to T_1. In the only other tier list in which $a_4 \in T_2$, namely $\{\{a_1, a_2, a_3\}, \{a_4\}\}$, any of the three other agents can profitably move to T_2.

The following shows the remaining seven partitions and possible profitable moves for an agent from each.

$\{\{a_4\}, \{a_1, a_2, a_3\}\} \to \{\{a_2, a_4\}, \{a_1, a_3\}\}$
$\{\{a_2, a_4\}, \{a_1, a_3\}\} \to \{\{a_1, a_2, a_4\}, \{a_3\}\}$
$\{\{a_1, a_2, a_4\}, \{a_3\}\} \to \{\{a_1, a_4\}, \{a_2, a_3\}\}$
$\{\{a_1, a_4\}, \{a_2, a_3\}\} \to \{\{a_4\}, \{a_1, a_2, a_3\}\}$
$\{\{a_1, a_3, a_4\}, \{a_2\}\} \to \{\{a_3, a_4\}, \{a_1, a_2\}\}$
$\{\{a_3, a_4\}, \{a_1, a_2\}\} \to \{\{a_2, a_3, a_4\}, \{a_1\}\}$
$\{\{a_2, a_3, a_4\}, \{a_1\}\} \to \{\{a_2, a_4\}, \{a_1, a_3\}\}$

Hence, there is no stable partition of Game 1. □

As illustrated by the example of Game 1, using matchup-oriented preferences does not guarantee the existence of a Nash stable outcome for an instance of k-TCFG in the general case.

Socially Consciously Stable k-Tier Lists

As we saw in Observation 1, it is sufficient to find a maximal-utility tier list in order to find a socially consciously stable one. We define a *local search* on tier lists, and claim that, in the k-tier list setting, it always produces a maximal-utility k-tier list. A local search on k-tier lists that moves agents if they desire to move and have permission to move from the rest of the tier list will improve the total utility of the tier list every time it moves an agent. Following the same principle as for Theorem 1, the resulting tier list will be socially consciously stable when local search halts.

Theorem 3. *For any instance I of kTCFG with matchup-oriented preferences, there exists a socially consciously stable tier list.*

Using three separate algorithms, **Siler sort**, a dynamic programming algorithm called **kTierOrderPreserve**, and **local search** (applied in that order), we find k-tier lists that are socially consciously stable, utilitarian, robust, and friendly. We call this three-part algorithm **TriPart(I,k)**.

The dynamic programming algorithm kTierOrderPreserve takes as part of its input a Siler stable tier list, and its outputs describe the highest-utility order-preserving k-tier list for that instance. The first output, $U[1, n, k]$, is numeric and represents the change in total *utility* in the tier list as a result of the best possible order-preserving transformation from n tiers to k tiers. The second output, $V[1, n, k]$, is a *vector* describing this transformation as a sequence of k pairs. For example, let $\{\{a_1\}, \{a_2\}, \ldots, \{a_n\}\}$ be a Siler's stable tier list. An output of $Y, [[1, i], [i + 1, j], [j + 1, n]]$ indicates that the best 3-tier list that preserves order is $\{\{a_1 \ldots a_i\}, \{a_{i+1} \ldots a_j\}, \{a_{j+1} \ldots a_n\}\}$ and that the total utility of agents in this tier list is equal to the total utility of the initial tier list plus Y.

Algorithm 1 kTierOrderPreserve

Input:
 instance I
 number of agents n
 matchup matrix Win
 number of tiers k
 $S(I)$ a Siler tier list
Initialize $Cost$, an $n \times n$ matrix of zeroes
for i *in* $1 \ldots (n - 1)$ **do**
 $c = 0$
 for j *in* $(i + 1) \ldots n$ **do**
  $c = c + Win[i, j]$
  $Cost[i, j] = c$
 end
end
Initialize U, an $n \times n \times n$ matrix of $-\infty$
Initialize V, an $n \times n \times n$ matrix of []
for $i, j \in \{1, \ldots, n\}$*, and* $i \leq j$ **do**
 $U[i, j, 1] = \sum_{h=i}^{j} Cost[h, j]$
 $V[i, j, 1] = [i, \ldots, j]$
end
for r *in* $2 \ldots k$ **do**
 for i *in* $\{1 \ldots n\}$*;* j *in* $\{i + r, \ldots n\}$ **do**
  for m *in* $(i + r - 1) \ldots (j - 1)$ **do**
   $mu = U[i, m, r - 1] + U[m + 1, j, 1]$
   if $mu > U[i, j, r]$ **then**
    $U[i, j, r] = mu$
    $V[i, j, r] = [V[i, m, r - 1], [m + 1, j]]$
   end
  end
 end
end
return $U[1, n, k], V[1, n, k]$

Observe that we need only compute the change in utility on each agent induced by merging with agents above itself, as merging with lower agents does not affect utility. Hence these costs are computed by the first loop in the algorithm, which has complexity of $O(n^2)$.

Proceeding through the algorithm: when $k = 1$, the only option available is to merge all remaining agents into a single tier. Otherwise, we iterate with r from 2 to k. To find the best r-tier list from any agent i to any agent j, we iterate over m to find the value of m that gives the greatest sum of utility of a $(r - 1)$-tier list from i to m and a 1-tier list from $m + 1$ to j. Hence, $U[i, j, r]$ describes the best possible change in utility of an r-tier list from i to j, and $V[i, j, r]$ describes the tier cutoffs for that best tier list. These correspond to return values when $i = 1, j = n, r = k$.

The iteration through r is $O(k)$, and iteration through i, j, and m are each $O(n)$. The time complexity of kTierOrderPreserve is therefore $O(kn^3)$.

The tier list described by the output of kTierOrderPreserve is not necessarily socially consciously stable.

kTierOderPreserve is, however, a heuristic that produces a high-utility argument to local search.

Entities such as Smogon often know ahead of time approximately how many tiers they want to use for a given instance (Smogon b). In the case of unknown k, a manager can select a favorite tier list from among a handful of viable possibilities. Whether or not a value of k is known in advance, k-TCFGs provide an answer to the research question posed by Siler (Siler 2017); our algorithm can now compute high-utility, friendly tier lists with good robustness as well!

Experiments

Here, we empirically test our heuristic algorithm, using generated instances and comparing the result of our procedure against other k-tier lists. Any k-tier list passed to local search will result in a socially consciously stable partition, and we have justified the use of our dynamic programming approach by asserting that its output, passed to local search, will result in a tier list with better results in utility and robustness. We run each component and each pair of component algorithms of TriPart, and compare their outputs to that of TriPart on the same instances, to demonstrate whether each of the three parts of the procedure significantly improves the results.

Unbiased Instances

We generate instances of TCFGs by randomly choosing win values $Win[i, j] \in [-1, 1]$ for $i > j$, and set $Win[j, i] = -Win[i, j]$. We set $Win[i, i] = 0$.

In each generated instance, we pass several k-tier lists to k-tier local search. These are: the list resulting from calling Siler's sort and kTierOrderPreserve (TriPart); a randomized k-tier list (Random); a randomized list of singletons passed to our dynamic algorithm (No S. sort); the output of Siler's sort divided into k approximately equal tiers (Even); and the output of Siler's sort divided into $k - 1$ singletons and one high tier containing $n - k + 1$ agents (Uneven). In addition, we recorded the output of kTierOrderPreserve without subsequent local search (No Search). The total utility, friendliness, and robustness of each of these k-tier lists was recorded from each instance.

We studied instances of 50, 100, and 200 agents, generating 100 instances of each size and comparing the average results for each method of obtaining a k-tier list, for k values selecting all values less than n from $\{2, 4, 8, 16, 32, 64, 128\}$.

Biased Instances

In the previous experiments, win frequencies between agents are generated from a uniform distribution. In many of the intransitive structures that tier lists attempt to capture, clear patterns emerge, and the structure may resemble a linear one. To ensure our procedure is effective in these instances, we generate randomized Win matrices with bias.

For $i > j$, we initially randomly chose a value $x \in [-1, 1]$ from a uniform distribution, and then set $Win[i, j] = min(1, x+(i-j+1)/n$. As before, $Win[j, i] = -Win[i, j]$ and $Win[i, i] = 0$. We examined the same set of k-tier lists for each instance, and once again generated 100 instances each of 50, 100, and 200 agent environments, using the same set of values of k.

Results

We found that the results for the different sizes of instances were similar and we present only the results from the 200-agent instances. What we saw was that the kTierOrderPreserve algorithm consistently performed well, finding high-utility tier lists with few singletons and reasonable worst-case fitness to tiers. By considering pairs of components of that three-part algorithm, we show that each part contributes significantly to the algorithm.

For example, the lower scores in all three of our evaluative criteria for the No Sort tier list indicates the importance of the order of input to the dynamic programming portion. Siler's sort ensures a sensible input order as compared to randomized input order. We cannot disprove the possibility that for certain instances, a stronger input order can be found that will result in a higher-scoring final k-tier list. However, Siler's sort is applicable in all instances, and our results prove its reliability.

The Even and Uneven tier lists were constructed without kTierOrderPreserve, and so these results indicate the usefulness of that algorithm. We see that for large k, the Even tier lists may rank more highly in friendliness, which can be expected from passing equally-sized tiers to local search. By contrast, Even scored lower than TriPart in friendliness and robustness at all levels of k, and the averages for Uneven were always strictly worse than the averages for Even.

The average results for No Search were so marginally different from the results for TriPart that they would not be visible under the number of significant figures in our table, and so these results are not shown. We also observed the average number of iterations of local search for TriPart, which at $n = 200$ ranged from 0.0 (at $k = 2$ in biased instances) to 9.33 (at $k = 2$ in unbiased instances). These results show that including local search in TriPart may not significantly improve utility; however, local search ensures that the output is socially consciously stable, and in TriPart it is completed in relatively few iterations.

Note that in our implementation of local search, if two agents i and j have an available move that is profitable and permissible under socially conscious stability, and if the set of tiers whose utility is affected by the movement of i is disjoint from the set of tiers whose utility is affected by the movement of j, both i and j may move in the same iteration of local search. Although this caused a slight speedup, each iteration of the algorithm is still $O(n^2)$.

Conclusion

Our results demonstrate the effectiveness of TriPart for efficiently finding at a socially consciously stable tier list that is utilitarian, robust, and friendly. Probabilistic matchup preferences, which are useful outside the contexts of kTCFGs, strengthen these results, leading to better and more useful tier lists. Our newly introduced notions of friendliness and robustness are applicable measurements for any future efforts in this topic, helping us to understand which partitions are most useful. Thus, we have provided answers to Siler's challenge, and offered new structures for consideration.

References

Banerjee, S.; Konishi, H.; and Sonmez, T. 2001. Core in a simple coalition formation game. *Social Choice and Welfare*.

Bogomolnaia, A., and Jackson, M. O. 2002. The stability of hedonic coalition structures. *Games and Economic Behavior* 38(2):201–230.

Chen, S., and Joachims, T. 2016. Modeling intransitivity in matchup and comparison data. In *Proceedings of the ninth ACM International Conference on Web Search and Data Mining*, 227–236.

Irving, R. W. 1985. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms* 6(4):577–595.

May, K. O. 1954. Intransitivity, utility, and the aggregation of preference patterns. *Econometrica: Journal of the Econometric Society* 1–13.

Saarinen, S.; Goldsmith, J.; and Tovey, C. 2015. Probabilistic Copeland tournaments. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 1851–1852.

Schlueter, J.; Addington, C.; and Goldsmith, J. 2021. Anchored team formation games. In *FLAIRS '21*.

Siler, C. 2017. Tiered coalition formation games. *FLAIRS 2017*.

Smogon. Smogon university - competitive pokémon community. `https://www.smogon.com/`.

Smogon. Ubers is now a tier; a new metagame called anything goes is added; m-ray is banned from Ubers. https://www.smogon.com/forums/threads/ubers-is-now-a-tier-a-new-metagame-called-anything-goes-is-added-m-ray-is-banned-from-ubers.3523205/.

Spradling, M.; Goldsmith, J.; Liu, X.; Dadi, C.; and Li, Z. 2013. Roles and teams hedonic game. In *International Conference on Algorithmic DecisionTheory*, 351–362. Springer.

Spradling, M. 2015. *Role Based Hedonic Games*. Ph.D. Dissertation, University of Kentucky.

Spradling, M. J. 2017. Optimizing expected utility and stability in role based hedonic games. In *The Thirtieth International Flairs Conference*.

Tsogbadrakh, T., and Spradling, M. 2019. Genetic approach to stable partitioning in online role based hedonic games. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2840–2847. IEEE.

Tversky, A. 1969. Intransitivity of preferences. *Psychological review* 76(1):31.

Unbiased	TriPart			Random			No S. sort			Even			Uneven		
k	Ut.	Friend	Robust	Ut.	Friend	Robust	Ut.	Friend	Robust	Ut.	Friend	Robust	Ut.	Friend	Robust
2.00	566.65	1.00	-15.83	540.69	1.00	-15.78	527.70	1.00	-16.58	547.59	1.00	-16.59	507.60	1.00	-18.39
4.00	987.03	1.00	-11.62	863.24	1.00	-11.51	848.15	1.00	-12.31	958.48	1.00	-12.58	647.52	0.99	-17.35
8.00	1320.72	1.00	-7.97	1177.10	1.00	-8.33	1166.69	1.00	-9.08	1290.26	1.00	-8.94	1026.82	0.98	-13.88
16.00	1574.07	1.00	-5.30	1410.83	1.00	-6.26	1406.35	1.00	-6.77	1538.95	1.00	-6.25	1311.98	0.99	-10.81
32.00	1752.82	1.00	-3.20	1577.56	0.99	-4.62	1579.69	0.98	-4.92	1721.33	1.00	-3.92	1491.48	0.95	-9.14
64.00	1871.82	1.00	-1.72	1681.37	0.79	-3.67	1682.27	0.79	-3.60	1847.70	1.00	-2.30	1638.66	0.70	-8.04
128.00	1942.53	0.49	-0.63	1557.04	0.34	-2.49	1567.76	0.35	-2.46	1923.25	0.56	-0.99	1808.18	0.25	-6.49
Biased	kTierList			Random			No S. sort			Even			Uneven		
k	Ut.	Friend	Robust	Ut.	Friend	Robust	Ut.	Friend	Robust	Ut.	Friend	Robust	Ut.	Friend	Robust
2.00	4178.91	1.00	-35.25	3886.90	1.00	-41.27	3515.21	1.00	-51.14	4154.16	1.00	-36.54	2563.21	1.00	-74.81
4.00	5375.80	1.00	-15.58	4968.31	1.00	-29.07	4900.34	1.00	-29.06	5337.55	1.00	-16.97	2948.47	0.96	-71.22
8.00	5820.71	1.00	-8.39	5516.02	1.00	-17.70	5542.26	0.99	-16.18	5783.36	1.00	-9.52	4216.80	0.97	-53.16
16.00	6074.11	1.00	-5.26	5765.85	0.98	-13.41	5832.57	0.99	-10.98	6035.92	1.00	-6.13	4781.88	0.96	-42.64
32.00	6246.68	1.00	-3.07	5886.50	0.77	-10.34	6003.96	0.88	-7.50	6214.27	1.00	-3.95	5246.84	0.69	-33.89
64.00	6362.61	1.00	-1.67	5719.21	0.44	-7.33	5892.34	0.53	-5.95	6338.47	1.00	-2.30	5686.64	0.33	-25.56
128.00	6432.23	0.49	-0.63	4172.50	0.16	-4.46	4462.92	0.18	-4.09	6413.09	0.56	-1.00	6198.61	0.13	-11.85