NATIONAL JOURNAL OF SPEECH & DEBATE

VOLUME VII: ISSUE 1

SEPTEMBER 2018

SEVEN MODEST PROPOSALS FOR DEBATE TABULATION: REFLECTIONS ON TRADITIONS FROM BRINGING MATHEMATICS TO BEAR

BY T. RUSSELL HANES

SEVEN MODEST PROPOSALS FOR DEBATE TABULATION: REFLECTIONS ON TRADITIONS FROM BRINGING MATHEMATICS TO BEAR

BY T. RUSSELL HANES *

* Mathematics chairperson, Northwest Academy, Portland, Oregon; Master of Arts in teaching mathematics, Lewis & Clark College, Portland, Oregon; Bachelor of Arts in mathematics, Columbia University, New York. The author can be reached at russell.hanes@gmail.com to clarify ideas or methods for the development of software.

Abstract

This article is not a guide to running a debate tournament. For that, I recommend Jon Bruschke's excellent *How to Tab*.[1] Instead, this article is a collection of seven ideas for running debate tournaments more efficiently and fairly, methods for how to: rank teams, assign sides, use standard deviation of opponent strength in pairing teams, create non-rigid elimination rounds, use scheduled elimination rounds, assign judges, and balance judging panels. Some ideas are immediately actionable; most would require new features in software.

Organization

There are four main steps in pairing debate rounds during a tournament:

1. Ranking the teams after each round
2. Assigning the teams to a side
3. Pairing the teams with an opponent
4. Assigning a judge to each debate

The organization of this article follows these steps in order; recommendations are not given in order of importance or ease of implementation. Any proposals can be combined together, or one could be used by itself. Debate communities can decide what works for their specific needs. I provide details, such as formulas and algorithms, as suggestions only for clarification. Any formula or algorithm in this article should be adapted and tested before tournament use.

[1] http://commweb.fullerton.edu/jbruschke/Web/how%20to%20tab%20text.doc

1. Ranking the teams after each round

Debaters do not perform with invariant strength. Good teams have off rounds; weak teams have moments of glory. If we put teams in a ranking of strength, where A is superior to B, B is superior to C, etc., we might naively believe that A always beats B and C, that B always beats C, and so on—but this assumption is false. Although A may be *likely* to beat C, this is not a *certain* outcome. This does not mean the ranking is inaccurate. We can truthfully say that A is the best team if it has mere 51% chances of winning over B and C.

In any competitive activity, there exists a logistic relationship (i.e., the shape of an S-curve) between opponents' strengths and their likelihood of winning the match. Two evenly matched teams might have 50-50 odds, while a stronger team might have a 51-49 or a 90-10 advantage over a weaker opponent. This logistic pattern appears in football, basketball, chess, and even debate.[2]

A. Problems with win-based rankings

No single win should be considered proof of which team is better. We need to think in probabilities instead. If team A has the following probabilities of beating its opponents at a tournament,

90% 80% 55% 50% 45% 45% 10%

then its most likely outcome is $0.90 + 0.80 + 0.55 + 0.50 + 0.45 + 0.45 + 0.10 = 3.75$ wins. But team A could have anywhere from 2 to 6 wins because the 55%, 50%, and 45% rounds could each go either way. One tournament, with only six to eight preliminary rounds, is insufficient to smooth out the variability in a team's performance. Consequently, while it is *likely* that a 6-1 team is better than a 4-3 or a 2-5 team, it is not *certain*, depending on the vagaries of the results in close debate rounds. Win-loss record is an imprecise way to rank debate teams at a tournament.

[2] I tested season-long debate data sets and found they follow the logistic pattern closely: http://art-of-logic.blogspot.com/2012/07/probability-of-upsets.html.

Over an entire season, however, teams fall close to their expected number of wins.[3] The variability cancels out after 30 to 40 rounds. Elo and Glicko ratings are based on the logistic pattern, so work well over an entire season.[4] The difficulty for a tournament director is that eight rounds are not smooth enough data to use Elo or Glicko ratings.

A further problem is the creation of cycles. A cycle looks like this: A beats B, B beats C, and C beats A. What happened is that at least one win was an upset.[5] For example, if the true ranking is {A=1, B=2, C=3}, then C scored an upset against A.[6] In practice, no ranking can be consistent with every win. No matter how the teams are ranked, there must be some upsets and cycles because of teams' variability in performance. In debate, it seems as an empirical matter that about 80% of wins can be consistent with rankings at the upper limit; in other words, 20% of wins are upsets. This does not mean such rankings are wrong or pointless—a good ranking estimates each team's strength by looking at its average performance over many rounds, filtering out upsets and outlier points.[7]

I once worked on a scheme to rank teams by weighted wins: teams earned more points for wins against strong opponents than weak ones; team lost more points for losses to weak opponents than strong ones. Despite many attempts, weighted wins never became an accurate ranking method; cycles create too much noise. Weighted wins takes a variable measure, win-loss record, and *amplifies* the noise rather than filtering it out. Any variant of adjusted wins will suffer the same problem.

B. Ranking teams by speaker points

The variability of team performance in winning rounds during one tournament is a major limitation for any win-based ranking method (the traditional method, Elo and Glicko ratings, weighted wins, and so on). My experiments indicate that median speaker points are a more accurate way to rank teams during

[3] Hanes, T. Russell. (2017). "Introducing the Logit Score: A New Way to Rank Debate Teams." *National Journal of Speech and Debate*, *5*(2), January.

[4] See http://collegedebateratings.weebly.com for an example.

[5] Note well: this is not the same thing as a low-point win. Team A might be more highly ranked overall than B, but A might lose the round *and* get lower speaker points. Team A, though better on average, had an off round and performed below expectation in both points *and* result.

[6] http://art-of-logic.blogspot.com/2009/08/network-graph-theory-and-debate.html

[7] http://art-of-logic.blogspot.com/2015/07/study-of-speaker-points-and-power.html

a tournament than wins.[8] Empirical research also corroborates this.[9] Put in the most basic terms: points are less variable than wins; and points are more informative than the binary of win/loss. Less variability and more information mean points-based rankings are more accurate than win-based rankings. It may surprise many debate communities that points are more accurate than wins. Speaker points seem subjective and arbitrary. This perception is probably a matter of availability bias: we all know some judge whose points are out of line. But why would we expect this judge's assessment of which team won to be more accurate?

It would be worthwhile to post clear rubrics for speaker points and encourage judges to review them. It might also be good to, in a positive way, let judges see their summary statistics after tournaments to understand how they deviate from community norms. These simple measures could help instill more confidence in the accuracy of speaker points, even though the variability of points is already low.

Moreover, a few inaccurate judges do not affect median points much.[10] Speaker points can be made even more accurate by adjusting each judge's scores to better hue to community norms. Second-order z-scores work well for this purpose. I also specified another method for adjusting scores in a previous article.[11] Using the median (not the average) of speaker points and preferring adjusted points would help increase confidence in points as well.

Nevertheless, the debate community may be reluctant to abandon wins entirely when it comes to ranking teams, with good reason. Switching to points-only rankings incentivizes eloquence without substance. There is no downside, however, to using *opponent* speaker points to understand the difficulty of the *schedule* a team faced. Just as points are more accurate than wins, opponent points are more accurate than opponent wins to measure schedule strength, and the switch will not change any team's behavior.

[8] http://art-of-logic.blogspot.com/2015/01/100th-post.html

[9] http://art-of-logic.blogspot.com/2015/07/study-of-speaker-points-and-power.html

[10] For three- and four-round tournaments, drop high/low points is the median. For five- and six-round tournaments, drop two high/two low points is the median. For seven- and eight-round tournaments, drop three high/three low points is the median.

[11] Hanes, T. Russell. (2018). "Testing a New Ranking Method, the Logit Score, With Simulated Tournaments." *National Journal of Speech and Debate*, *6*(2), January.

C. Hybrid measures

The best solution to ranking teams is to use a hybrid measure that factors wins and points into a single score. One option is to add together the z-score for wins and the z-score for speaker points. In my experiments, this hybrid z-score was a more effective way to rank teams than any win-based method.

Another option is the logit score. I showed through empirical research and experiments that the logit score was more accurate than any win-based method and slightly better than hybrid z-scores.[12] The method for calculating the logit score is covered in my previous article, "Testing… the Logit Score," but a basic description is that the logit score is *similar* to the average of the best opponent a team beat, the worst opponent it lost to, and its median speaker points. Wins against weak opponents, losses to strong opponents, and too-high or too-low speaker points are outliers that do not affect the logit score much. What matters is overall performance: earning good speaker points consistently and winning as many rounds as possible.

As the name implies, the logit score is based on the logistic model of winning and losing.[13] It would require a new tool to be added into tabulation programs, but the calculation is not too complex. Because it is a best-fit regression, the logit score takes a considerable amount of computer time to run even in a well-implemented program; I estimate five minutes for a large tournament. It therefore makes sense to use the logit score only to decide which teams break to elimination rounds. Ranking during preliminary rounds could be done by the hybrid z-score method, which is good enough.

Proposal 1: Use opponent speaker points to determine a team's schedule strength. Use hybrid z-scores to rank teams during preliminary rounds and logit scores to rank teams for deciding elimination round breaks.

[12] http://art-of-logic.blogspot.com/2017/02/the-logit-score-new-way-to-rate-debate.html

[13] As such, each debate circuit would need to do logistic regressions in order to find the right model value, *k*, to use in calculating its logit scores. Each college circuit (NDT, Parli, etc.) would find its own *k* based on community norms in giving speaker points. Each high school format (CX, LD, PF) would find its own *k*; I imagine the "national circuit" high school *k* may need to be different than "local circuit" high school *k* for the same format. Tournament results would need to specify the *k* value used in finding logit scores. This would be programmed into tabulation software to run automatically. See: https://en.wikipedia.org/wiki/Elo_rating_system#Most_accurate_K-factor.

2. Assigning the teams to a side

A. Constrained Side Equalization

Although the traditional method ensures each team alternates sides, there is a superior method—constrained side equalization—that gives tournament directors greater flexibility. In the traditional method, teams are assigned to a side at random in odd rounds; in even rounds, teams are assigned to the opposite side as the previous round. This means that odd rounds are open with all p possible pairings are considered.[14] In even rounds, because one-half of the teams are due Aff. and one half are due Neg., only about $0.5p$ pairings are possible.[15] Even rounds have a severe limitation of flexibility which can cause a small tournament to "lock up" and have few good options for pairing some teams.

Constrained side equalization (C.S.E.) works differently.[16] In even rounds, any team can be paired against any other, regardless of sides in the previous round. Unlike the traditional method, even rounds in C.S.E. are open with all p possible pairings considered. Teams are paired first, then assigned a side based on equalization. If one opponent was Aff. and one was Neg. in the previous round, each is assigned to its opposite side to even out both teams. However, if both opponents went Aff. in the previous round (or both Neg.), then the sides are randomly assigned.

Side equalization in even rounds gets approximately one-half of teams to side balance. On average, however, about one-fourth of teams have +2 Aff. rounds and about one-fourth of teams have -2 Aff. rounds after an even round. For example, after two rounds, approximately one-fourth of the teams are 2 Aff. - 0 Neg., one-half are 1 Aff. - 1 Neg., and one-fourth are 0 Aff. - 2 Neg.

Odd rounds in C.S.E. repair the imbalance. First, the +2 Aff. teams are assigned to "due Neg."; the -2 Aff. teams are assigned to "due Aff." This is the constraint in the name "Constrained Side Equalization." Once this is done, the other teams (the balanced ones) are assigned to sides at random, and then all teams are paired with opponents.

[14] Theoretically, $p = 0.5\,n(n-1) = 0.5n^2 - 0.5n$ for n teams. However, because some teams are from the same school, the real p can be lower.

[15] The number is $0.5n \cdot 0.5n = 0.25n^2$, which is approximately $0.5p$ for a large n of teams.

[16] http://art-of-logic.blogspot.com/2018/05/why-debate-tournaments-have-been-doing.html

At the end of an odd round in C.S.E., every team ends up +1 or -1 Aff. For example, after three rounds, half the teams are 2 Aff. - 1 Neg., and the other half are 1 Aff. - 2 Neg. Ending a tournament after an odd round requires no additional considerations. If the tournament ends on an even round, the last round should be paired traditionally so every team has side balance.

The constraint in odd rounds in C.S.E.—that +2 Aff. teams are due Neg. and -2 Aff. teams are due Aff.—eliminates only about one-eighth of possible pairings, leaving 87.5% of possible pairings available for these rounds. This is significantly better than the *half* of possible pairings that are eliminated in even rounds of the traditional method, as shown in Table 1.

Table 1

Pairing possibilities

Method	Odd rounds	Even rounds	Average
traditional	p	$0.5p$	$0.75p$
C.S.E.	$0.875p$	p	$0.94p$

In practical terms, C.S.E. leaves more options are on the table, giving the tournament director more flexibility and preventing the tournament from locking up.

B. Team constraints

It maximizes possible pairings if all teams from a school are on the same side. Putting two teams from a school on opposite sides creates a useless, blocked match. Therefore, the best way to assign teams to sides is semi-randomly based on school.

In this method, side-constrained teams are assigned their side first. Next, schools are lined up in descending order of total teams. Starting with the biggest school, evaluate whether its constrained teams are more heavily on due Aff. or due Neg. If there is an imbalance, then all the *unconstrained* teams from that school are assigned to double down on the imbalance: if there are more “due Aff.” teams, then all the unconstrained teams are assigned Aff. as well. If the constrained teams are balanced Aff. and Neg., then all the unconstrained teams, as one bloc, are randomly assigned to either side.

The schools are assigned in descending order until one side has half the teams. If assigning a school causes one side to go over half the teams, then some of its teams are randomly selected to change sides. Once one side has exactly half the teams, any remaining schools are assigned to the opposite side. While this method does not guarantee the fewest same-school blocked matches, it significantly reduces them.

Proposal 2: Use constrained side equalization to pair preliminary rounds except, if it exists, the final even preliminary round; use semi-random side assignment to put as many teams from one school on the same side as possible.

3. Pairing the teams with an opponent

A. Optimizer

Adding an optimizer to debate tabulation software is the single biggest improvement that could be made to tournament tabulation since the advent of computers. Optimizers would have a myriad of uses. An optimizer exists in Excel, called Solver. Optimization is a well-understood application in computer science,[17] and code exists in every language to borrow from.

The most obvious use of an optimizer is pairing opponents. Rather than pairing opponents one-by-one, an optimizer considers multiple pairings *simultaneously* and picks the optimal one. The process starts by creating a matrix of all possible matches:

Aff → Neg ↓	A	B	C
D			
E			
F			

[17] https://en.wikipedia.org/wiki/Hungarian_algorithm

The matrix is then populated with scores based on the desirability of each pairing. (Scores are created by a formula at the programmer's discretion—more on this later.) Lower scores are more desirable:

Aff→ Neg↓	A	B	C
D	2	2.2	1.1
E	1.5	1	10
F	3	2	2.4

The worst pairing in this tournament is C versus E. The best pairing is B vs. E. Finally, the software selects the set of pairings with the lowest overall score, as noted:

Aff→ Neg↓	A	B	C
D	2	2.2	1.1
E	1.5	1	10
F	3	2	2.4

The optimizer in this instance chose the best opponents for A, C, D, and F and the second-best opponents for B and E. This is the *optimal* set of pairings; it is possible another choice may make a few teams better off, but it would be at the expense of making *more* teams worse off.

Scores in the matrix can be generated by a formula that factors in any number of elements: matches in team rank, wins, or speaker points; style; geography; etc. For example, the formula could give lower scores to teams with the same number of wins or higher scores to teams with similar styles. The formula could add to the score if schools hail from nearby states (+1), the same state (+2), or the same city (+3). Teams would therefore be more likely to debate opponents from different areas of the country. This is a good idea for any national tournament.[18] Any number of variations are possible, and furthermore, the formula could change from round to round. Perhaps early rounds use only geography in the

[18] Hanes, Russell. (2012). "Ensuring Geographic and Skill-level Mixing at Nationals." *National Journal of Speech and Debate*, *1*(1), October.

formula, while later rounds factor in both team rank and geography. The benefit of using an optimizer is that multiple variables can be considered and weighed together in creating a pairing.

B. Running quasi-round robins

Small tournaments can lock up if they are run with power-matching. Because opponents must have the same number of wins in power-matching, there is a small, dwindling number of possible pairings for each team. Quasi-round robins, on the other hand, give teams a wide cross-section of opponents from the top to the bottom of the pool.[19] There are many potential opponents, so the tournament never locks up. The main consideration in running a quasi-round robin is making sure each team has an equivalent schedule strength of opponents to every other team.[20] There are two ways to run a quasi-round robin: pre-match or pair on-the-fly.

Option I: Pre-match

The teams entering a tournament are ranked 1 to *n* based on prior results. Each team is given a cross-section of opponents from the top to the bottom by hand.

The best test of whether the scheduling has been done fairly is *not* simply the team's average opponent strength. Of course, every team's average opponent strength should be similar to everyone else's, but that is insufficient proof the schedules are fair. A team with only good and bad opponents could have an acceptable average opponent strength even though it has no mediocre opponents, as it should. The essential second criterion is that every team's standard deviation of opponent strength is similar to everyone else's. Standard deviations that are too high indicate extreme opponents with no middles, while standard deviations that are too low indicate too many opponents in a cluster.[21] A few sample teams at a tournament is shown in Table 2.

Team A's schedule as shown in Table 2 is problematic. The average is acceptable if a bit high, but the standard deviation is out of line with the other teams. Team A has no mediocre opponents and its schedule needs to be reshuffled. Teams

[19] These are called "quasi" because each team only debates a fraction of all possible opponents. For example, in a six-round tournament of 12 total teams, each team debates only 50% of possible opponents.

[20] http://art-of-logic.blogspot.com/2009/03/half-round-robin.html

[21] http://art-of-logic.blogspot.com/2011/02/method-to-create-balanced-groups.html

B and C, however, are similar to each other on both the average and standard deviation; they have fair schedules relative to each other.

Table 2

Sample teams at 32 team quasi-round robin

Team	Opponents' rank	Average	Std. Dev.
A	1, 3, 4, 29, 31, 32	16.67	14.06
B	2, 5, 13, 19, 22, 25	14.33	8.52
C	1, 6, 10, 15, 21, 28	13.50	9.07

Option II: Use the Optimizer

Optimizers can be used to pair quasi-round robins on-the-fly. The first preliminary round is matched randomly. After it is over, the teams are ranked. Then an optimizer is used to pair teams ignoring win-loss records. The goal is to give each team a wide cross-section of opponents, so the formula for giving scores each team's potential opponents would be:

$$(std.dev.of\ all\ opp.strength\ including\ potential\ opp.)^{-1}$$

The more spread out the opponents are, the higher the standard deviation, which result in a lower, more desirable score. This formula gives the lowest score to the potential opponent that most balances out each team's schedule.

To populate the matrix, this formula is used for each team, squared, and added together to yield the score for a potential pairing.[22] Here is one such matrix with scores:

[22] Squaring first punishes bad scores (low spread of opponents) more heavily. Therefore, squaring first makes choosing pairings that are good for only one team unlikely.

Team=Rank *{Rd 1 & 2 opponents}*	**A=1** *{B=2, C=3}*	**B=2** *{A=1, C=3}*	**C=3** *{A=1, B=2}*
D=4 *{E=5, F=6}*	1.71	0.99	1.29
E=5 *{D=4, F=6}*	0.88	0.75	0.99
F=6 *{D=4, E=5}*	0.69	0.88	1.71

Pairings A vs. D and C vs. F are tied for the single worst choice. Team A's opponents would be the cluster of {B, C, D}; D does not balance out A's schedule. Team F's opponents would be the cluster of {C, D, E}; C does not balance out F's schedule. Team A vs. F, however, is a great pairing that balances both A's schedule {B, C, F} *and* F's schedule {A, D, E}.

After each round, the teams are re-ranked, and the process repeats. In my trials, this method works well to pair a quasi-round robin on-the-fly.[23] Because the rankings develop based on team performance during the tournament, instead of the tournament director's pre-tournament judgment, this on-the-fly method has lower potential for bias.

C. *Strength-of-schedule pairing*

When pairing a quasi-round robin, win-loss records are entirely ignored in pairings. A 5-0 team might debate a 0-5 team in the sixth round. The goal is to give each team a cross-section of opponents, so the spread of opponent strength is key.

With large tournaments, this option is still available and a good one to consider. There is another option for large tournaments, what I have dubbed strength-of-schedule pairings. Teams would be paired against opponents with the win-loss record but also to maximize the standard deviation of opponent strength.[24] An easy schedule (relative to teams with the same win-loss record) in early rounds would lead to tougher opponents later.[25]

[23] http://art-of-logic.blogspot.com/2014/06/world-cup-groups.html
[24] That is, the matrix score would be lowest for two opponents which have the same number of wins *and* if the pairing improves both teams' standard deviation of opponent strength. Scores would be higher if either condition is not met, and highest if both are not met.
[25] http://art-of-logic.blogspot.com/2009/08/another-way-to-visualize-strength-of.html

This method is *not* the same as high-low or high-high power-matching. In strength-of-schedule pairings, the rounds are mixed—some are high-low, while others are high-high, depending on what is required to even out each team's schedule.

Strength-of-schedule pairings are more effective at balancing out schedule strength for each team with the same win-loss record than high-low or high-high power-matching.[26] At the end of preliminary rounds, each 6-0 team will have similar opponent strength to every other 6-0 team; each 5-1 team will have similar opponent strength to every other 5-1 team; and so on.[27]

Proposal 3: Compare the fairness of teams' schedules with both the average and standard deviation of opponent strength; run tournaments such as quasi-round robins or use strength-of-schedule pairings to improve the fairness of schedules.

D. Non-rigid elimination rounds

Currently, elimination brackets are rigid: after teams are first set in the brackets, nothing changes. If the bottom-ranked team upsets the top-ranked team in elimination round one, then that bottom-ranked team inherits the bracket the top team earned—which means the bottom-ranked team now has the easiest elimination schedule of all the teams. One upset leads to a huge advantage. Researchers have shown empirically this creates problems, for example in the N.C.A.A. tournament.[28] The rigid bracket structure creates bias in favor of lower-ranked teams.

The solution is elimination rounds in which teams are paired such that the top-ranked team remaining always debates the bottom-ranked team remaining. If the top-ranked team is eliminated by the bottom-ranked team, then the bottom-ranked team next debates the second-to-top team. The bottom-ranked team continues to have the hardest possible schedule. This rectifies the bias of rigid brackets. Rigid brackets only make sense when teams travel long distances to their next game, which is not an issue for debate tournament elimination rounds.

[26] http://art-of-logic.blogspot.com/2009/03/strength-of-schedule-pairings-work.html

[27] In quasi-round robins, however, every team will have similar opponent strength to every other team, e.g., 6-0 team and a 0-6 team will have similar opponent strength.

[28] https://fivethirtyeight.blogs.nytimes.com/2011/03/15/when-15th-is-better-than-8th-the-math-shows-the-bracket-is-backward/

The teams' rankings could be static from preliminary rounds or could be updated during elimination rounds, for example by the formula $3 \cdot prelim.wins + elim.ballots$, then the tie-breakers of preliminary speaker points, opponent strength, etc.[29] Updating the rankings would mean that a top-ranked team that continues to squeak by on 2-1 wins would eventually lose its top rank, while a bottom-ranked team that continues to win 3-0 elimination rounds would slowly move up in rank. This incentivizes teams to win every ballot. Dropping a ballot allows a team to move on but weakens its rank for later elimination pairings.

Proposal 4: Pair elimination rounds so the top-ranked team remaining debates the bottom-ranked team remaining, second-to-top debates the second-to-bottom, etc., while adjusting teams' ranks in elimination rounds based on both preliminary and elimination round performance.

E. Scheduled elimination tournaments

Double-elimination tournaments pose a unique challenge: odd numbers of teams remain after some rounds. This occurs with regularity and creates many byes at the tournament. In elimination rounds, every team should debate every round on principle.

One solution is the scheduled elimination tournament. The number of teams who will remain in the tournament after each round is fixed and announced before the tournament begins. After each round, the remaining teams are re-ranked, then the correct number of bottom-ranked teams are eliminated, always in twos, fours, sixes, etc. Table 3 shows a possible schedule.

The particular schedule of eliminations can be left to the tournament director or organizing committee. A single-elimination tournament eliminates exactly one-half of teams each round, while a double-elimination tournament eliminates about one-third of teams each round (it varies based on how pull-ups perform). I modeled the schedule in Table 3 somewhat in between these two extremes: never more than one-half of teams are eliminated per round, averaging about one-third each time. Other nice features of a scheduled elimination tournament are that the total number of rounds is predictable and judging needs are

[29] Or hybrid z-scores that include both preliminary and elimination round results.

known in advance.[30] However, running a scheduled elimination tournament would necessitate collecting speaker points during the tournament as tie-breakers.

Table 3

Scheduled elimination tournament

Round	Team
1	34
2	34
3	26
4	20
5	14
6	8
7	4
8	2

The pairing rule might be that the top-ranked team remaining debates the bottom-ranked team remaining that it is eligible to debate, with no repeated opponents until all other possibilities are exhausted.

Proposal 5: Use a scheduled elimination tournament in which a predetermined number of teams is eliminated after each round.

4. Assigning a judge to each debate

A. Preliminary judge assignment

Judge assignment is perhaps the trickiest tournament directing tasks. Tournament directors would like to use every judge's commitment in preliminary rounds. Community ratings might help recognize good judges but also help tournament directors use every judge fully. Each team would receive a survey to rate each judge, including "unknown" as a rating. These surveys would be averaged

[30] Tournament directors could take advantage of this predictability by scheduling division B to start just as division A's judging needs drop, releasing judges from division A into B. For example, PF rounds could start when LD enters round five. A known number of LD judges would then be available to help in PF rounds one and two.

to create community ratings, which could be supplemented by the tournament director for those unknown judges who receive a rating from fewer than 50% of surveys.

After teams are paired, potential judges every pairing could be run through the optimizer.

Round → Judge ↓	standby	A vs E	B vs F	C vs D
w	1	2.4	2	4.5
x	3	5	1.3	4
y	2	3	3.2	3.3
z	3	1	1.1	6

The "standby" score is easy to explain: it is the number of rounds of commitment a judge still owes. In this case, judge *w* only owes one more round, so gets this round off. By the next round, judges *w* and *y* will both owe one more round. The optimizer will decide who gets the round off based on whether *w* or *y* is the best fit to judge the pairings in the next round. Judges *x* and *z* have the most commitment left so will be the last ones to get rounds off.

The other matrix scores are assigned by whether each debate pairing has two, one, or no strong teams and the judge is highly rated. Assigning a highly-rated judge to two strong opponents is given a low, desirable score—but so is assigning a poorly-rated judge to two weak opponents. For example, in the matrix above, judge *z* is highly rated; A, E, B, and F are all good teams; so those potential assignment scores are desirably low. Teams C and D are both weak, however, so *z* is an inappropriate judge for that pairing. Furthermore, high, undesirable scores are given to assignments where a judge has already seen one or both of the teams. Judge *x*, for example, may have already seen team A once before. All these considerations can be written into the formula that populates the matrix cells.

The advantage of an optimizer is that all rounds in all divisions can be assigned judges *simultaneously* from one matrix. If low scores are given to varsity teams plus experienced judges and low scores are given to novice teams plus inexperienced judges, then the optimizer puts experienced judges in varsity and inexperienced judges in novice, while preserving the ability to bump a few judges into other divisions if they are not suited to judge a pairing in their preferred division. An experienced judge might do five varsity rounds and one top-ranked junior varsity round, hoping down for the latter without any active judging pool management from the tournament director. With an optimizer, judging pools do not need to be separated by division. There is one pool of all judges.

Proposal 6: Use community ratings plus an optimizer to assign judges, with scores being lowest for assignments in which a judge's and teams' experience level are commensurate.

B. Balancing elimination round judging panels

Elimination round judging panels should be balanced on several different variables: age, sex, race, stylistic preferences, rating, and so on. Balancing all those variables at once is difficult, but an optimizer solves the problem readily.

First, judges are divided into three groups randomly.

The first third will form the basis of building the panels. Each of these judges is assigned randomly to a panel: panel A, panel B, etc.

The second third of judges is now scored as "partners" against the first:

Judge 1 → Judge 2 ↓	x	y	z
u	4	2	1
v	1	1	3
w	3	0	2

The scores in this matrix indicate the number of variables on which the two judges are the same. For example, judges u and x are both older white males with B+ ratings (four similarities). Judges w and y, on the other hand, are as different as they can be. The optimizer picks the lowest scores, creating the panels u+z, v+x, and w+y.

The optimizer now reruns for the final third of judges, scored against these incipient panels:

Panel → Judge 3 ↓	*u*+*z*	*v*+*x*	*w*+*y*
s	2.5	2	3
r	5	1	1.5
t	1	2	2

The scores are increased for two judges being the same on a variable (+0.5) or for all three judges being the same on a variable (+1). For example, judges *t*, *u*, and *z* represent a very diverse, balanced panel. Judges *r*, *u*, and z are too similar in many regards—a homogeneous panel. The optimizer picks the lowest scores, creating the panels *s*+*v*+*x*, *r*+*w*+*y*, and *t*+*u*+*z*.

Once the panels are created, a debate round can be assigned to that panel.

Proposal 7: Use an optimizer to create elimination round judging panels with maximal diversity.

Conclusion

In terms of the practicability of the proposals, a change from win-based rankings would be an easy adjustment. Ranking opponent strength by points instead of wins is an instant change. The hybrid z-score would be trivial to add to tabulation software. The logit score is a moderately easy addition to make. My research has shown that changing ranking methods affects a sizeable chunk of the teams—perhaps 10% should break but do not. We owe it to all debaters to rank teams as accurately as possible.

Constrained side equalization would make running small tournaments easier but is quite strange at first glance. The software changes are at a moderate level of ease.

Evaluating schedule strength by looking at standard deviations is also an instant change. Pairing tournaments as quasi-round robins or by using the strength-of-schedule method would make for fairer tournaments—but these changes would be major re-orientations in how debate communities think about what preliminary rounds should be. By the same token, pairing non-rigid elimination rounds and using elimination schedules would create fairer tournaments, although these are also drastic adjustments for debate communities.

The biggest and most important change outlined in this article would be addition of an optimizer to tabulation software, which could be used to massively improve opponent pairings and judge assignments. At this point in time, the true benefits of computing power have not yet been fully realized for debate tabulation.