Smart Scan App Proposal

Arjun Vedantham
Programmer, 4001A
javathunderman@nomik.xyz

1 Preface

As programmer for team 4001A, I typically have minimal responsabilities during a Vex Robotics Competition except the occasional tweak to our robot's autonomous functionality. Thus, I am usually the one in charge of scouting teams for potential alliance selections.

4001A has done scouting in two distinct ways. The simpler way, which is what we used during our first competition year (In The Zone, 2017-2018) mainly consisted of watching other matches and robots to see who might be a good partner. While this was successful at some competitions, by and large this was not a good scouting strategy and we decided to give greater attention to scouting during tournaments.

The following season (Turning Point, 2018-2019), we decided to employ a more analytical and statistical method for scouting. This mainly consisted of using various digital tools, including the VexDB site (which contains detailed statistics for team), as well as using developing own scouting tools [1] to supplement ordinary match observation. While this was considerably more successful, it was ultimately too much work to analyze all of the matches and all of the teams. Additionally, there was little to no information about a team's particular robot and your potential compatibility with that robot, unless you were particualrly plugged into the Vex Forum or the VTOTW Discord server. Thus, we are proposing a new solution for scouting and robot discovery: automating the most of the process using a new power rankings algorithm contained in a full fledged scouting mobile app.

[1] See our GitHub page (@phsengineering) for our listskillsutil and matchpredictor scripts.

2 App structure

The app will consist of two main scouting methods: objective, and power-based.

2.1 Objective scouting

The objective scouting functionality of the app will consist of algorithms and tools that are already publicly available and widely used, including a match predictor based on the calculated CCVM statistic provided by the Vex Via app, as well as skills rankings and possibly score notifications for matches.

2.2 Power-based scouting

This part of the app will utilize power-based scouting and our own algorithm for determining the best potential partners at a certain competition. [2] The preliminary power-based ranking statistic will use the following formula:

$$\begin{aligned} R = (3 * CCVM) + (1.5 * Awards) + 1.2 * (CCVM_{previous}) \\ + 2 * (skills_{atcomp}) + 0.3 * (WP) + 0.2 * (AP + SP) \end{aligned} \tag{1}$$

We are using this formula because (in our opinion) a team's CCVM, awards, and skills scores give a better sense of the team's performance. We have intentionally deprioritized WP, AP, and SPs due to the fact that they are awarded based on the performance of an alliance during a match, not a specific team. Based on this, we give an initial ranking to each team. [3]Then, based on user-contributed data, we create a multiplier for each team based on their robot's specific properties. For example, if we use the Turning Point game as an example, we know that each team should have a method for scoring flags. Using team profiles, a robot from a specific team can be described as using a flywheel, puncher, or catapult. If the flywheel robots have a higher average ranking across all of the teams in the app database, the algorithm grants a 1.15X multiplier. This multiplier function can be

[2] CCVM, OPR, DPR, skills, and match scores will all be fetched using the VexDB JSON API.

[3] "CCVM previous" refers to the sum of a team's CCVM scores from the previous two years at that point in the season. Awards won by a team in that season will be factored in accordingly: +8 for excellence, +10 for tournament champions and robot skills, +6 for design, other judged awards and tournament finalists, +3 for tournament semifinalists, and +2 for judges award. The sum of these awards ratings will be used for the awards variable in the algorithm.

expanded to work with any specific scoring or drive system that is important to game competitiveness. After applying this multiplier, we re-rank the teams accordingly. [4]

$$R = M * ((3 * CCVM) + (1.5 * Awards) + 1.2 * (CCVM_{previous}) + 2 * (skills_{atcomp}) + 0.3 * (WP) + 0.2 * (AP + SP)) \tag{2}$$

2.3 Sample initial rankings

If we take the Firebirds Winter VRC Qualifier (SKU: RE-VRC-18-7443) as a sample dataset, we would obtain the following output [5]:

```
4001A 200
729M 120
8637A 87
242A 86
242B 72
4001F 69
9071C 54
9071B 50
4017A 42
242F 39
242Z 33
242D 15
19030C 15
9071X 15
4001E 13
19030P 7
9071A 0
4017B -6
9071Y -6
242E -17
242C -21
19030A -22
```

[4] It should be noted that this multiplier will only help teams, and so there will not be a 0.8X multiplier or something like that.

[5] We are unable to use the re-ranking algorithm since robot specifications for each team are not available.

3 User-contributed data

As mentioned in the previous section, this proposed app would be dependent on user-contributed data. We propose creating a site that ties in with the app allowing teams to create ”team datasheets”. These datasheets display hard statistics pulled from VexDB (OPR, DPR, CCVM, match records, skills, etc.) in addition to robot profiles that teams can add to their datasheets. These profiles can contain scoring mechanisms (which can be used in the formula described above), reveal videos, websites, etc. and can be tied to QR codes that can be placed on the robot as part of a non-functional decoration to allow for more convenient scouting and networking at large tournaments. [6]

4 Conclusion

This app will be first developed for Android, and will use a Node.js and SQL backend for datasheet management. Statistical and awards data that are not user-contributed will be fetched from VexDB's JSON API. We hope to begin work on this app by May of 2019, release it to a group of Eastern Pennsylvania teams first as part of a closed beta, and then release it publicly.

[6] How much data a team decides to reveal about their robot is entirely up to them. If a team chooses not to update their datasheet with robot specifications, they will be given a 1X multiplier during the ranking algorithm's second round of calculations.