JOURNAL OF
EMERGING INVESTIGATORS

Article

Rubik's cube: What separates the fastest solvers from the rest?

Kepler Boyce[1] and Cornelis Storm[2]
[1] Gunn High School, Palo Alto, California
[2] Eindhoven University of Technology, The Netherlands

SUMMARY

The Rubik's Cube is a 3D combination puzzle. Speedcubing is the activity of solving Rubik's Cubes as fast as possible. In this paper, we identified and quantified key factors that enable some speedcubers to be faster than others. Knowledge of these factors could allow speedcubers to focus their practice deliberately into specific areas, accelerating their improvement. We hypothesized that a low fraction of pause times, low regrip frequency, and low move counts would show the strongest correlations with faster solves. To test this, we analyzed 69 solve videos frame by frame across a wide variety of ability levels, as well as survey data collected from 1,385 speedcubers. To our knowledge this study represents the most comprehensive factor study of speedcubing to date. We discovered that the following factors correlate most strongly with solve times: frequency of regrips and rotations, fraction of time spent pausing, cross move count, number of algorithms known, turn speed, duration of pause in transition between cross and first two layers (F2L) steps, and orientation of the last layer (OLL) and permutation of the last layer (PLL) recognition time. Based on our findings, we generated specific recommendations for how speedcubers of different ability levels might most efficiently improve their solve times.

INTRODUCTION

Invented in 1974 by Hungarian professor of architecture Ernő Rubik, the Rubik's Cube is a combination puzzle in the form of a 3x3x3 cube (**Figure 1**) (1). Each of the six faces comprises nine squares, with each square being one of six solid colors: white, yellow, red, orange, green, or blue. Each face of the puzzle can be rotated independently via an internal twisting mechanism. To solve the puzzle, one must turn the faces to rearrange the squares so that each face shows the same color on all nine. The puzzle became a worldwide sensation in the 1980s and since the early 2000s it has seen a revival in popularity (2). The Rubik's Cube remains the most popular puzzle toy ever created, with more than 350 million sold as of 2018 (2, 3).

The combinatorics of the Rubik's Cube gives the puzzle a vast number of possible configurations: 43,252,003,274,489,856,000 or $4.3x10^{19}$ unique states can be reached by turning its faces (3, 4). For most people, merely solving the puzzle is a daunting challenge. However, algorithmic approaches allow one to solve it reliably, and with practice, quickly. In principle, any configuration can be solved in at most 20 moves, which is colloquially referred to as "God's number", as an omniscient being could solve the cube with this efficiency (5). Thus "speedcubing" was born—the challenge of solving the Rubik's Cube from a scrambled state as quickly as possible (6). The first world speedcubing championship was held in 1982, where American Minh Thai set a record of 22.95 seconds, and since then solve times have steadily decreased (7). As of this writing, the fastest solve in a competition is 3.47 seconds, set by Yusheng Du in 2018 (8).

For many speedcubers, the essential question is always "what do I need to do to get faster?" The obvious answer is to turn the faces faster, but this is not specific enough to provide real help to a speedcuber looking to improve. Moreover, it is not clear which factors are most important for solvers of varying skill levels, and different cubers have unique areas of strength and weakness. Should one focus on memorizing more algorithms, or do they have diminishing returns beyond a certain number of algorithms? Is it better to turn slower and more smoothly, or turn fast with longer pauses? What about for a solver who averages 30 seconds versus one who averages 10?

This study discusses some principal elements of speedcubing. For those not familiar with this discipline, we first explain the solving method popularized by Jessica Fridrich, as of 2022 the most widely used speedsolving method (9, 10). The Fridrich method solves the cube in layers, so it is considered a "Layer by Layer" method. In the case of the Fridrich method, these layers are solved from the bottom upwards, so the bottom and middle layers are called the "first two layers" and the top layer is called the "last layer."

Many speedcubers first learn a simpler Layer by Layer method, often called the Beginner's Method, before transitioning to the Fridrich method (11). The Beginner's Method requires fewer algorithms—memorized move sequences that apply specific actions to the cube—but has more steps as a result, making it easier to learn but inefficient compared to the Fridrich method. While the Fridrich method requires 54 moves on average, the Beginner's Method, on average, uses 135 (12).

The Fridrich method is also known as CFOP, an acronym for the method's four steps that are executed in sequence: cross, first two layers (F2L), orientation of the last layer (OLL), and permutation of the last layer (PLL) (**Figure 1**). First, the cross step completes the four edges of the first layer. Next, F2L involves pairing a first layer corner with its respective middle layer edge and inserting both together. This step is done four times, making F2L the longest step. Third, in OLL, the solver executes one of 57 algorithms to reorient the pieces in the last layer such that all pieces have the same color

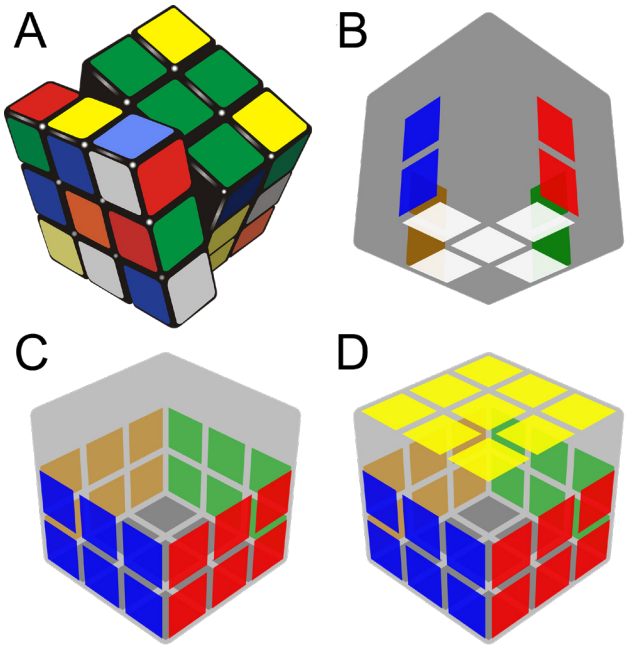

Figure 1: Rubik's Cube and solved sections after each step of the Fridrich method. (A) Rubik's Cube in a scrambled state. The goal of the puzzle is to rotate the layers to make each face a uniform color. Image is used under a Creative Commons license. **(B)** Solved cross step. **(C)** Solved first two layers (F2L) step. **(D)** Completed orientation of the last layer (OLL) step. See text for more description of these steps. Images by Wikimedia user Conrad Rider are used under a GFDL license.

facing upwards. For a slower but less algorithm-intensive variant, one can use 2-look OLL to split OLL into two steps, requiring nine algorithms in total (13). Lastly, the PLL step swaps and cycles the last layer pieces to put them in their correct locations without disturbing their orientations, solving the puzzle. There are 21 PLL algorithms in total, but one can use the 2-look version with 6 algorithms at a small speed cost (14).

Conversely to 2-look OLL and PLL, there are options with higher algorithm counts for speedcubers who desire further efficiency, such as Corners of the Last Layer (COLL), Winter Variation, and Zborowski-Bruchem Last Layer (ZBLL) (15, 16, 17). These techniques combine or skip steps of the solve to save time; for example, COLL performs OLL while permuting the last layer corners, Winter Variation orients the last layer while inserting the final F2L pair, and ZBLL solves the last layer in one algorithm from any position where all four last layer edges are oriented. The disadvantage is that these techniques require learning many additional algorithms. ZBLL, for example, has a staggering count of 493 algorithms. For this reason, many choose to stick with the 78 standard CFOP algorithms. For more information on CFOP and other speedcubing methods, see (10).

Another important concept is look-ahead, which is the ability to analyze the cube holistically, allowing the speedcuber to plan moves in advance and transition between steps more smoothly. A solver skilled at look-ahead can turn at a consistent speed throughout the solve whereas a solver who does not use look-ahead effectively will often pause during the solve to search for pieces required in the next step.

We hypothesized that look-ahead (i.e., duration of pauses) and move count efficiency for the cross and F2L steps would correlate most strongly with overall solve time, followed by OLL and PLL recognition times (i.e., lengths of any pauses prior to these steps). According to former world record holder Feliks Zemdegs, F2L is the step where a solver can expect to see a majority of their improvement (18). The step is largely intuitive (i.e., not strictly algorithmic), meaning it often causes intermediate solvers to make frequent pauses or hesitations as they try to find key pieces and plan their moves. Zemdegs also says that learning a variety of easy algorithms beyond the 78 needed for full CFOP can be highly beneficial for advanced solvers (18).

RESULTS

Video Analysis of Solves

Our most detailed data came from analyzing video recordings of individual cube solves. We analyzed videos from speedcubers of varying skill levels frame by frame to record variables like the time spent in each step, the duration of the solver's pauses, move counts, and total solve time. With this dataset, we performed a correlation analysis.

We expected that faster solvers likely would turn the cube faster than slower solvers. We found a strong measured correlation between average turn rate and solve time ($r(67) = -0.85$, $p < 0.00001$, **Figure 2**). Here, turn rate was quantified by Turns Per Second (TPS). Regardless of whether pause times were included, faster solvers had higher average turn rates.

We found that F2L was the most time-consuming step for virtually all solvers, comprising roughly 52% of the total solve time (**Figure 3**). Though faster solvers appeared to have comparatively faster crosses and slower last layers, the correlations were weak ($r(67) = 0.24$, $p = 0.047$ and $r(67) = -0.04$, $p = 0$ .744, respectively).

We saw more significant differences between solvers when we examined move count, which was the total number of turns used overall and in each step. Faster solvers spent fewer moves on the cross and last layer ($r(67) = 0.42$, $p < 0.00033$ and $r(67) = 0.34$, $p = 0.0043$, respectively) but

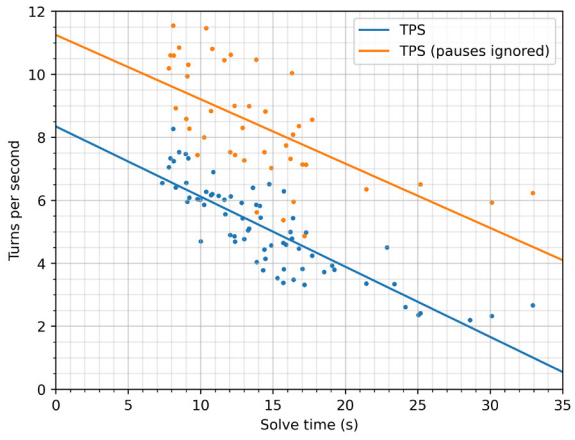

Figure 2: Measured turns per second (TPS) versus total solve time (16 solvers, *N*=69 solves). TPS equals the total number of moves divided by the solve time, while in "TPS (pauses ignored)" pause times were subtracted from the solve time. TPS correlated strongly with solve time (r(67) = -0.84, $p < 0.00001$).

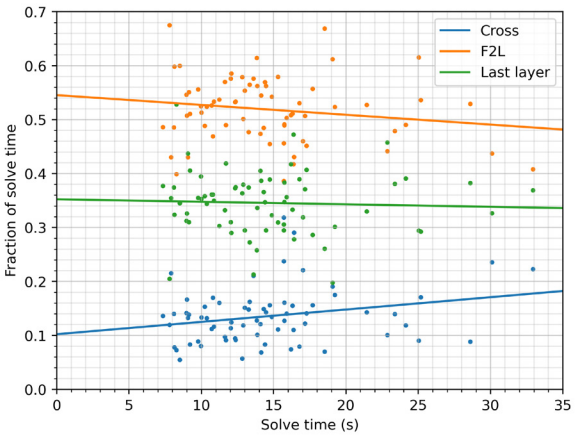

Figure 3: Measured fraction of solve time spent in each CFOP step versus total solve time (16 solvers, *N*=69 solves). The F2L step was the most time-consuming in virtually all cases.

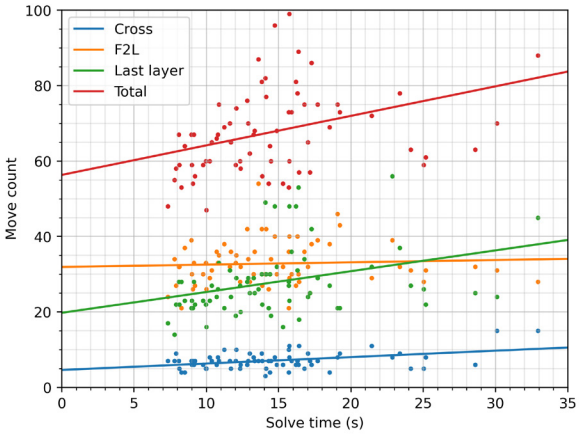

Figure 4: Measured move count for each CFOP step and total move count versus total solve time (16 solvers, *N*=69 solves). Faster solvers spent fewer moves on the cross and last layer steps (r(67) = 0.42, $p < 0.00033$ and r(67) = 0.34, $p = 0.0043$, respectively) and fewer moves overall (r(67) = 0.37, $p = 0.00200$), but we found no significant relationship between F2L move count and solve time (r(67) = 0.05, $p = 0.683$). Nearly all of the move count improvement was in the last layer, with a small amount in the cross.

there was negligible correlation between F2L move count and solve time ($r(67) = 0.05$, $p = 0.683$, **Figure 4**). Faster solvers achieved lower move counts overall ($r(67) = 0.37$, $p = 0.00200$).

Solve time and fraction of time spent pausing showed a strong correlation ($r(41) = 0.72$, $p < 0.00001$, **Figure 5**). We also observed systematic differences between faster and slower solvers regarding when they paused. Solvers who averaged above 20 seconds paused for a larger fraction of time in the cross and F2L steps as compared to the last layer (**Figure 5**). In terms of absolute duration of pauses, F2L was the largest source of pauses for all solvers, followed by the last layer (**Figure 6**).

Finally, our video analysis allowed us to observe key differences in how solvers move. Faster solvers performed regrips and cube rotations less often ($r(67) = 0.75$, $p < 0.00001$ and $r(67) = 0.48$, $p = 0.00003$, respectively, **Figure 7**).

Large-Scale General Survey

As a second source of information, we analyzed data from a survey of 1,385 respondents in the r/cubers Reddit online forum (19). This survey included only general self-reported information such as average solve time, years of speedcubing experience, and number of algorithms known. Our only result from the survey data is that solvers who know more algorithms tend to have a faster average solve time (**Figure 8**). Nearly all solvers faster than 10 seconds knew at least 100 algorithms, which is more than the 78 algorithms used in standard CFOP.

DISCUSSION

Faster solvers had higher turn rates, which made sense as there is a limit of roughly 54 moves for how low one's average move count can be with CFOP (**Figure 2**) (12). Once a solver reaches that point, time improvement must come primarily from increasing TPS. Thus, a question arises: Is a fast solver merely a slower solver "sped up?" That is, if we play recordings of slower solvers at 1.5x or 2x speed, are they indistinguishable from faster solvers? Or are there factors beyond turn speed that differentiate fast solvers from slower ones? To address this question, we needed to consider other aspects of performance.

Faster solvers spent fewer moves on the cross and last layer but showed no improvement in F2L move count (**Figure 4**). The first finding matches what we hypothesized—a faster solver will have gained better intuition for solving the cross in fewer moves and learned more algorithms for the last layer (most often by transitioning from 2-look OLL and PLL to the standard, more efficient variants). The latter finding surprised us, however, since many faster solvers invest time learning advanced techniques intended to reduce F2L move count and these efforts do not appear to benefit their solves.

Faster solvers also spent a smaller fraction of their solve

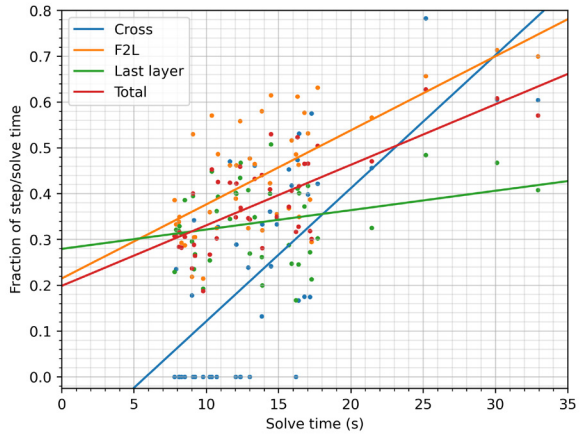

Figure 5: Measured pause time during each CFOP step (as a fraction of that step's total time) and total pause time (as a fraction of solve time) versus total solve time (11 solvers, *N*=43 solves). A pause was any sequence of video frames in which all layers of the cube were stationary. Pauses in between steps counted towards the following step. The fraction of time spent paused was highly correlated with total solve time (r(41) = 0.72, $p < 0.00001$), showing conclusively that a faster solver was not merely a slower solver "sped up." Solvers who averaged above 20 seconds paused for a larger fraction of time in the cross and F2L steps as compared to the last layer.

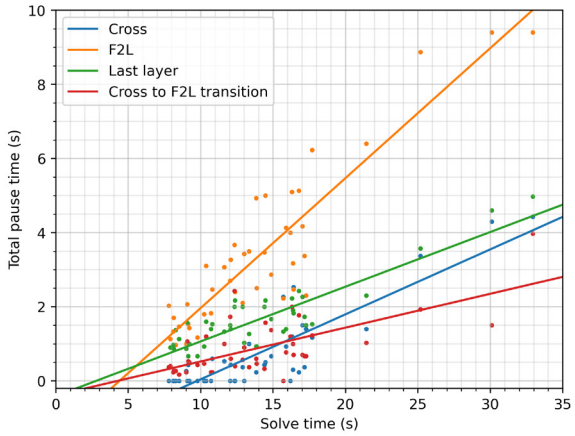

Figure 6: Measured absolute duration of pauses in each CFOP step and total pause time versus total solve time (11 solvers, *N*=43 solves). The F2L step was the largest source of pauses for all solvers, followed by the last layer.

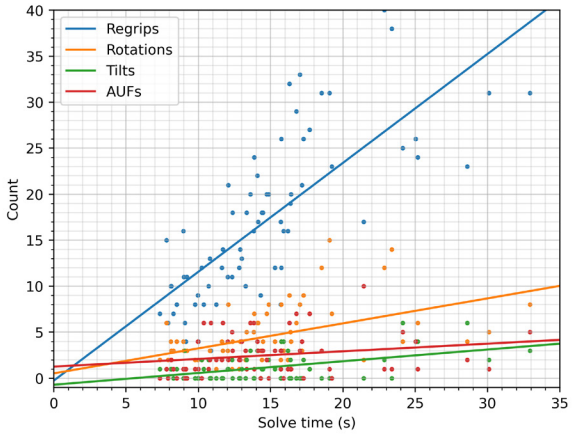

Figure 7: Measured number of regrips, cube rotations, cube tilts, and AUFs versus total solve time (16 solvers, *N*=69 solves). Regrip: when the solver moved a thumb to another face of the cube. Cube rotation: when the solver rotated the entire cube to face another side (also counted as two regrips, as the solver moved both thumbs to another face). Cube tilt: when the solver tilted the cube to view the back or bottom faces. AUF: Adjustment of the Upper Face; one AUF counted for every time the solver turned the upper layer without necessity. Faster solvers performed fewer regrips and cube rotations ($r(67) = 0.75$, $p < 0.00001$ and $r(67) = 0.48$, $p = 0.00003$ respectively).

times pausing. For example, an average 10-second solver spent 33% of their solve time pausing, while an average 20-second solver spent 46% (**Figure 5**). This result suggests that for a typical solver improving from 20 to 10 seconds, about 6.2 seconds of that time improvement is gained simply by reducing pause time. This striking result underscores the critical importance of improved look-ahead and algorithm recall for solvers in this ability range. For the cross, this correlation was also caused in part by the sizable number of solvers faster than 15 seconds who were able to plan all of their moves for the cross before beginning the solve, allowing them to complete the cross step with no pauses. In terms of absolute duration, a majority of every solver's pauses took place in F2L (**Figure 6**). As F2L relies on intuition rather than algorithms, this result suggests that a solver can greatly reduce their overall pause times by improving their F2L look-ahead.

The results described above suggest to us some crucial paths to improvement for solvers at different ability levels. For solvers who average above 15 seconds, key things to practice include: reducing regrips and cube rotations, working to reduce pause time, especially in the cross and F2L, using fewer moves to solve the cross, learning more algorithms to lower OLL and PLL move counts, and turning faster (i.e., increasing TPS). For those who average faster than 15 seconds, the following are important in addition: reducing the pause in the transition from cross to F2L and improving OLL and PLL recognition times to reduce last layer pauses. Regarding learning additional algorithms, there seemed to be an inverse correlation with average solve time even as far as 200+ algorithms, but we cannot say whether the faster times were a direct result of learning more algorithms. It may be that solvers who have spent more time practicing—and thus have faster solve times—also tend to know more algorithms because they have had more time to learn them. Regardless, it is worth noting that nearly all of the fastest solvers knew at least 100 algorithms.

These findings may help speedcubers by showing which specific areas are most important for improving at each skill level. They also provide benchmarks for determining which areas a certain speedcuber is strong in and which areas they need to work on. For example, one can record and analyze their own solves to see how their statistics compare to an average solver of their speed. Rather than relying on very general advice, speedcubers can now see in detail what they should practice based on their own abilities.

For future work, there are some areas left untouched by this study due to a lack of appropriate data. Variables like hours of practice per week, age when the solver started speedcubing, physical aspects of the speedcube, and color neutrality (the ability to solve equally well from any starting color orientation, as opposed to always starting with the white cross), for example, would be interesting to consider.

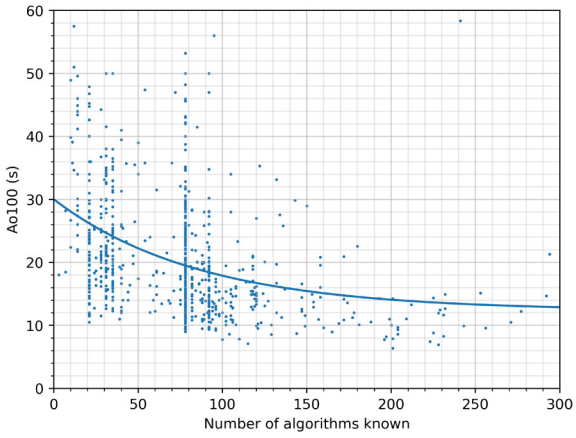

Figure 8: Average time of 100 solves (Ao100) versus number of algorithms known (survey of 1385 solvers). For an average of 100, conventionally the fastest 5 and slowest 5 times are removed and the mean of the remaining 90 times is taken. Nearly all solvers faster than 10 seconds knew at least 100 algorithms.

MATERIALS AND METHODS

This study used two separate datasets. All data and analysis scripts have been made available online (20). The two datasets were collected from different populations of speedcubers, so we did not combine them for any of our analyses. The published Python analysis script generates Figures 2–8 in this paper from the datasets (20).

Video Analysis of Solves

The first dataset contains data derived from video of 69 individual solves from 16 solvers of various skill levels. We obtained the videos from YouTube online and analyzed them using the Hawkeye software program, which allows video to be viewed frame by frame with accurate timestamps for individual frames (21). We recorded the following variables: solve time; TPS; fraction of solve time spent in each CFOP step; move count of each CFOP step; fraction of time spent pausing in each CFOP step and in the overall solve; duration (in seconds) of pauses in each CFOP step and in transition between the cross and F2L steps; and number of regrips, cube rotations, cube tilts, and unnecessary adjustments of the upper face (AUFs). We provide detailed definitions of each variable and explanation of recording methods below.

We defined the start of the solve as the last frame in which the solver had not yet turned any layers of the cube and the end of the solve as the frame when all layers of the cube became stationary in a solved state. The end of each CFOP step was the frame when the final move of that step concluded. We derived solve time in seconds and duration of each CFOP step by taking the duration of the solve or step in frames and dividing by the video frame rate (30 frames per second for all videos we used).

We defined TPS as the total number of moves executed by the solver divided by total solve time. We counted moves using the Slice Turn Metric (STM), where a rotation of any layer by any amount counts as one move (22). If the solver executed a move as two independent motions with a pause in between, however, we considered it as two moves despite being one move in STM; For example, if a solver turned the top layer clockwise, paused for a moment, and then turned the top layer clockwise once more, they performed two moves by our move counting system.

Pause times were all sequences of frames in which no layers of the cube moved by any amount. Pauses in between CFOP steps counted towards the following step. As such, we included pauses in the transition from cross to F2L in the total F2L pause time. We summed pause times across each step and divided by the duration of that step to determine the fraction of time spent pausing for that step.

Regrips were when the solver moved either thumb to another face of the cube. Cube rotations were when the solver reoriented the cube to face a different side; cube rotations also counted as two regrips as the solver moved both thumbs to another face of the cube. Cube tilts were when the solver tilted the cube, often to view the back or bottom layers. Unnecessary AUFs were when the solver turned the top layer without necessity. During the F2L step, for example, some solvers performed multiple rotations of the upper layer in a row while trying to find key pieces, resulting in unnecessary AUFs.

Large-Scale General Survey

The second dataset contains general information: months since the respondent learned to solve the Rubik's Cube, months of speedcubing experience, best single 3x3 solve time, best Ao5, Ao12, and Ao100, and a list of the algorithm sets known by the respondent. AoX refers to an average of X solves and is calculated as a mean with the fastest and slowest solves removed. For an Ao5 or Ao12, two solves are removed—the single fastest and slowest—while for an Ao100, the five fastest and five slowest times are removed (23). With permission, we collected this dataset from a 2020 survey with 1,385 responses by the r/cubers Reddit group (19).

We then created a Python script for analysis in which the data is cleaned by removing any data points that: (a) leave one or more fields blank, (b) have unusual values for best single, Ao5, Ao12, or Ao100 times (e.g. if the reported Ao100 is faster than the reported Ao5), or (c) report times greater than 500 seconds or more than 300 algorithms known, as there are very few responses in these ranges and they hinder the readability of graphs (20). To convert from lists of known algorithms to a quantitative number, the script parses each response's list of known algorithms, summing the number of algorithms in each listed set.

Analysis Methods

All of the data collected was quantitative. For analysis, our main objective was null-hypothesis testing of correlations between quantities of interest. We tested Pearson correlations because linear relationships are simple to model and test, and we have no prior reason to expect non-linear relationships. To this end, we calculated a Pearson correlation coefficient r from our sampled data, and then tested the null hypothesis H_0: $\rho = 0$ against the alternative hypothesis H_A: $\rho \neq 0$, where ρ is the population (true) Pearson correlation coefficient. To test we H_0 used the test statistic:

$$t = r \cdot \frac{\sqrt{n-2}}{\sqrt{1-r^2}}$$

from which we calculated a two-tailed p-value. The p-value expresses the probability of measuring a sample Pearson correlation coefficient r^* at least as extreme as r, assuming the null hypothesis is true (i.e., the populations are uncorrelated bivariate normal distributions). We adopted a criterion of rejecting the null hypothesis when the p-value was less than 0.05.

ACKNOWLEDGEMENTS

We thank the reviewers and editors for their helpful suggestions. We are grateful for the opportunity provided by Palo Alto Unified School District's Advanced Authentic Research program and the support of Rachael Kaci.

Received: August 4, 2021
Accepted: December 20, 2021
Published: July 24, 2022

REFERENCES

1. Rubik, Ernő. *Cubed: The Puzzle of Us All*. Flatiron Books, 2020.
2. Reese, Hope. "A Brief History of the Rubik's Cube." *Smithsonian Magazine*, 25 Sep. 2020, smithsonianmag.com/innovation/brief-history-rubiks-cube-180975911.

Accessed 11 Nov. 2021.
3. Hofstadter, Douglas R. "METAMAGICAL THEMAS." *Scientific American*, vol. 244, no. 3, Scientific American, a division of Nature America, Inc., 1981, pp. 20–39, jstor.org/stable/24964321.
4. "Mathematics of the Rubik's Cube." *Ruwix*, ruwix.com/the-rubiks-cube/mathematics-of-the-rubiks-cube-permutation-group. Accessed 11 Nov. 2021.
5. Rokicki, T., *et al.* "The diameter of the Rubik's Cube group is twenty." SIAM J. Discrete Math. vol. 27, 2013, pp. 1082–1105. doi.org/10.1137/120867366
6. *The Speed Cubers*. Directed by Sue Kim, Netflix, 2020.
7. "World Rubik's Cube Championship 1982." *World Cube Association*, 5 Jun. 1982, worldcubeassociation.org/competitions/WC1982.
8. "Rankings." World Cube Association, worldcubeassociation.org/results/rankings/333/single. Accessed 11 Nov. 2021.
9. Fridrich, Jessica. "My system for solving Rubik's cube." ws.binghamton.edu/fridrich/system.html. Accessed 11 Nov. 2021.
10. "CFOP method." *Speedsolving Wiki*. speedsolving.com/wiki/index.php/CFOP_method. Accessed 11 Nov 2021.
11. Gonzalez, Robbie. "How to Solve a Rubik's Cube, Step by Step." *Wired*, 5 Sep 2019. wired.com/story/how-to-solve-a-rubiks-cube-step-by-step. Accessed 11 Nov 2021.
12. Duberg, D., and Tideström, J. "Comparison of Rubik's Cube Solving Methods Made for Humans." Dissertation, 2015. Retrieved from urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-166727
13. "Step 3 - Orient last layer - OLL." *Ruwix*. ruwix.com/the-rubiks-cube/advanced-cfop-fridrich/orient-the-last-layer-oll. Accessed 11 Nov 2021.
14. "Step 4 - Permutate the last layer - PLL." *Ruwix*. ruwix.com/the-rubiks-cube/advanced-cfop-fridrich/permutate-the-last-layer-pll. Accessed 11 Nov 2021.
15. "COLL." *Speedsolving Wiki*. speedsolving.com/wiki/index.php/COLL. Accessed 2 Apr 2022.
16. "Winter Variation." *Speedsolving Wiki*. speedsolving.com/wiki/index.php/Winter_Variation. Accessed 2 Apr 2022.
17. "ZBLL." *Speedsolving Wiki*. speedsolving.com/wiki/index.php/ZBLL. Accessed 2 Apr 2022.
18. Zemdegs, F. "How To Get Faster?" *CubeSkills*. cubeskills.com/blog/how-to-get-faster. Accessed Jul 15, 2021.
19. "Mega-Survey 4 Results." *Cubers Subreddit*, Jun 28, 2020. reddit.com/r/Cubers/comments/hhmcmp/megasurvey_4_results. Accessed Jul 15, 2021.
20. Boyce, K. "Speedcubing data and analysis software." *Github repository*. github.com/KeplerBoyce/speedcubing-analysis. Accessed Jul 21, 2021.
21. Boyce, J. "Hawkeye video analysis software." *Github repository*. github.com/jkboyce/hawkeye. Accessed Jul 15, 2021.
22. "Metric." *Speedsolving Wiki*. speedsolving.com/wiki/index.php/Metric. Accessed Jul 21, 2021.
23. "Average." *Speedsolving Wiki*. speedsolving.com/wiki/index.php/Average. Accessed Jul 21, 2021.