5-Style Mega Manual

What is 5-style?

The idea of introducing 5-style is to have a better understanding of how higher order commutators or comms+conjugates work. It is one of the biggest algset that exists and even overshadows 1LLL (1 Look Last Layer in CFOP method). The main aim with going down this big algset rabbithole is to understand cubing as a long term project and try to get more involved algsets in order to solve it efficiently. The ultimate aim is to improve a person's transfer learning capabilities by training in a rigorous method like 5style. By transfer learning I mean, getting useful mental improvements out of cubing. For example 5-style will help improve linguistics and memory.

Background about blindsolving

5-Style is a method that tries to solve a 3x3 cube in around 50-70 moves using commutators which is the go-to way of doing a 3x3 blindfolded. This method is to be used in 3BLD to reduce the time of execution which stands at ~11 seconds now using the 3-style method with fancy parity and floating algs subsets. The main applicability of this method is for the MBLD since there is no restriction on the upper limit on the number of cubes as newer methods continue building complexity to improve the MBLD score. I am not sure if it will give a major boost to the 3BLD event, as exec is a lot TPS dependent and on muscle memory. But yes, letter quads, a side product of 5-style will be definitely useful to improve 3BLD memo and shave off more milliseconds than what noddon and throwing the cube cover does.

Origin of the word style is kinda not clear, but after the BH algs were developed a decade ago, they were slowly worked upon and made into a better algset called 3-style. Noah Arthurs in his tutorials introduced the word style to mean that the specific algset has more fingertrickable algs, and are more focussed on 3gen and regripless execution.

Abhijeet

Unblackboxing this method to make it easily understandable
Making a beginner or a person new to speedcubing methods understand 5style can be an uphill task. First the person needs to gain confidence and try out one node of the method as a proof of concept.

Has this idea been previously thought of?
Most probably not. Cubing is quite new and there are still a lot of ideas that have not been tested out. Only time will tell which ideas are worth the effort and to be brought into the mainstream.

1. Context

As a kid, I have always been fascinated by the game of chess, and I play it for quite a while during high school. The thing that I used to find that separated the very high-class GM from a normal amateur chess player, was the amazing preparation the GM used to put to get his technique and repertoire correct. As an amateur player, I had faint level positional chess sense, and I was riding high on attacking chess and tactics. But this thought of how important chess preparation has always stayed in my mind.

I am always perplexed by the Rubik's Cube, and my perception keeps changing as I grow older. At first, it used to be a great feat just to solve one side, upon which I stayed satisfied for years, then one day I learned how to solve it completely by looking up an online tutorial.

Doing the cube blindfolded was the next challenge which I took up, which took me another four years to get the hang of, and I did it by leapfrogging from the Old Pochmann method to using M2/R2 in my solves. As the years went by, I slowly and steadily replaced each inefficient M2/R2 algorithm with a newer and faster 3-style algorithm.
Another breakthrough in the blind scene came with the US BLDers smashing the blindfold times, by making really fast 3-style algorithms. This was the turning point, as TPS and thinkahead and better algs became the primary thing to invest time into.

It was again in the lockdown induced in the year 2020 due to the COVID pandemic that I understood the parallels between chess and cubing and realised that cubing does not have the intense opening preparation that normal classical chess have. Even speedchess has a lot of preparation at the titled players level. Having a lot of cases evaluated beforehand is a very new thing to cubing, as it was new to the chess world in the 1950s when Botvinnik introduced intense chess preparation and methodology that is widely adopted today by all professional chess players. Cubing is still new and the viability of having such preparation is shrugged off as a waste of time, as cubing records still optimized on fingertricks, lookahead/pattern recognition, TPS and lots of practice.

Motivation

I attended my first major competition at the Asian Championships in Beijing in 2016. It was an amazing experience, and the major takeaway I had from the tournament was the impact I got from three cubers, who seemed to be on another level: Shivam Bansal, Kaijun Lin, and Gianfranco Huanqui.

Kaijun Lin had already inspired me to take up the Roux method as my main solving method, and he had shown how BLD times are brought to such low and consistent times with practice and focus.

Gianfranco Huanqui is a revolutionary BLDer, who has made new kinds of finger tricks and made many new algs which are novel and fluently executable.
On the final day of the Asians, Gianfranco Huanqui did over 300 3BLD solves in one day at the venue. I had lost count of the sub-20s, sub-18s he got and it was spectacular to watch him practice. In every solve, he looked at a point where he thought he could have improved, and continued self-learning in this way.

I also remember Shivam Bansal saying a mind-blowing fact after the prize distribution that, our mind is so powerful that we technically should be able to store petabytes worth of information in it which is even more than a supercomputer or a cluster of computer harnesses. By having such brain power the limits of MBLD is unreachable , he said.

After the competition, I headed back to Chennai in India, feeling more driven to create something new. The next month (Nov 2016), I finally thought of taking the plunge into making a new method that I had always thought of but never did. I had decided to list out and memorize all the 5 cycle algorithms for 3x3, for both corners and edges, and also get some 4 cycle comms which come handy in finishing off edges in most of the cases and new parity algs. I wanted to make a memory element for each letter quad which could be retrieved doubly faster than 2 letter pairs, and I wanted a 12ish move count finger trick-able 5-cycle algorithm that could solve the case in the fastest time and with very less finger movement.

Epiphany

I was attending Shaastra Open 2014, my second ever WCA competition. I was 18 years old at that time and had just finished a 4/8 MBLD attempt which felt quite satisfying. The competition went well, and I came second in 3BLD with a time of 2: 06, behind Kabyanil Talukdar who got a 1:20. After the prize distribution ceremony, Arunachaleswarar, an overzealous skewber, who was doing blindfold Skewb solves by insane tracking, saw me doing M U M’ U’ on a 3x3. I showed him that these 4 moves are so efficient that they cycle 5 pieces without affecting the rest of the pieces. He added up to me saying that, you should make a whole system out of this idea. I shrugged it off saying it's just too hard as there are many cases, running into over a million unique cases. The same day, earlier I talked to the MBLD winner Vikram Mada who did 6/6 using only single letter memorization

(not even letter pairs) and discussed with him conveying how I wish to go beyond letter pairs and go to letter quads. I quickly calculated the number and said a quarter million cases. He said that this just looks impossible, saying that he was already having a tough time transitioning to 480 letter pairs and there I was talking about an algset that runs into a hundred thousand cases. I went back to college and borrowed a big Scrabble book from the library to just see how many 4 letter words exist out there, there were a lot but not enough to cover all the cases of letter quads which can emerge on a 3x3.

Roux-inspiration

I have been using the Roux method since the year 2016. The step in a Roux that fascinates even a normal cube solver is the LSE part or the last 6 edges.

Most of the time we try and solve the LSE, we focus on getting the arrow edge orientation shape which will make all the edges “good”, by performing M/M’, U/U’, M/M’ moves.One night when I was going LSE only solves, I realized that the speedsolving approach to the LSE is quite simple and does a great job, and even with EOLR, UL/UR prediction and pinkie pie algsets, we totally avoid the concept of commutators in the solving process. Using 5-cycles on some of the LSE cases might be a thing to dwell upon and can be a possibility in some cases. The cuber will have to option select between doing LSE the normal way or to recognise if a 5-cycle exists that needs to be solved and solve it.

Go Game Complexity and me getting overwhelmed

When I encountered the news of AlphaGo in 2016, I really wanted to learn the game of Go. Sadly, there were no tutorials and enough documentation on it other than AGA Go association, and to learn advanced techniques, I had to start learning the East Asian languages.

Japanese was always a powerhouse in Go, and recently Korea has been competitive in the Go world. China had always produced good Go players, and they were the most shocked by the AlphaGo program. So, I had to learn Chinese, Korean, and Mandarin to really understand the content and new knowledge the Go world keeps creating. So, TLDR Go is so complex that I felt that such complexity will eventually come to other full information games like Twisty puzzles and in that the 3x3 puzzle.

Tabla Riyaaz and my will to grind

Indian rhythmic rich history has always played with the beat of 4, placing 4 beats anywhere has resulted in such rich rhythmic compositions.The major learning of the letter quads happened when I came back to Gujarat for my postgrad. I started learning the Tabla instrument, which I had learned from the age of 13-15, and now I had more maturity in understanding it. The phrases that are played are very important and the way they fit into the entire structure and are time bound within the time cycle (16 beats or 12 beats etc0 is beautiful. My main aim with the kuad language for letter quads is to be able to get fluency with 4 letter phrases in order to memorise LQs better and use them effectively as a memo system, and as a way to represent a unique UF5 case (5-style case from the UF buffer)

2. Why try this method?

I feel similar to a prepared Chess player, or a prepared Go player before I do a 3BLD solve. Rather than a nervy person spamming the Y-perm, and locking up and getting frustrated about the lockups, you will feel composed and at ease during the solve. 3BLD solve would be similar to counting up to the number 5 in mental effort, and if you get comfortable in it there will be, on an average of just five letter quads in a particular scramble. You will come off the beginner tag that every CFOP user or M2/OP user gets when he/she stops learning algs, after they learn OLL, PLL, M2 and just focus on finger tricks, and not newer algs.

Till now I have gotten many easy scrambles officially. I once got a good 10/4 scramble in a competition in 2015. I reached a bottleneck in my improvement after that, which I could only improve on by drilling 3-style algorithms and getting all the algorithms in the algset sub-1 seconds. In hindsight, I do not want to reach another bottleneck, so I thought of developing this method.

Disclaimer: Please delve into this method only if you love speedcubing, and only if 3BLD/MBLD is your main event, otherwise this is not worth investing your time into.

How to make learning 5-cycles less daunting?

Learn how to grind algorithms in an efficient manner to help you get a ton of algorithms into your head and muscle memory. The best way to tackle this humongous algset method is to consider only one alg at a time, learn that one alg in a day, and move on to another alg. The daily rhythm is the best strategy to get everything solidly sorted out in your mind. I also wish to make a video series focussing on subcategories of the algset and how to make a huge memory map of algs in the head. Learn the Yo Notation.

How hard is 5-Style compared to the well known hard algsets like ZBLL?

5-Style has ~12600 edge algorithms and ~60000 corner algorithms. For the corners, the [R U D] 3-style algorithms are already really fast, and the edge algorithms move count has been reduced by ~40%. 5-Style algset size is similar to $(\text{ZBLL})^2$ which is just plain crazy.

5-Cycle Algorithm Count

Edges: 126720 (excluding the flipped edges and cycle break cases)

Abhijeet

Depth	Elements	Reduced(O)	5-Cycles
6	192	8	8
7	480	20	20
8	2,112	88	88
9	5,472	228	228
10	25,632	1,068	958
11	58,320	2,430	2,256
12	138,384	5,766	4,700
13	143,496	5,979	3,914
14	81,384	3,391	496
15	7,056	294	4
Sum	462,528	19,272	12,672

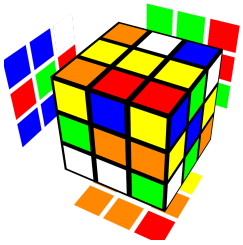
Credit: Bruce MacKenzie

5. 5-Style Solve Examples+ Comparisons

My orientation is Yellow on top and Orange in front
My lettering scheme is also a bit different, so please do check it out before walkthrough these solves. Before the reconstruction and after scrambling, please rotate to the YO orientation.

1. R B2 D2 F2 R2 D R2 B2 L2 D2 U' B' L B' F' L' D' F L' R2 Fw Uw2

Abhijeet



Reconstruction (5-style UF/UFR)

My lettering scheme
Edges: **MFQO LUVI KEGE WX**
Corners : **IQFS VOBM**

Speffz :
Edges: **TQGP RWVL HEME KU**
Corners : **QTFU KHAL**

//Edges
R U S' U' R' U R E2 S' E2 R' U'
L' D' S' M D M' S L
B' M2 U M2 U2 M2 U M2 B
R U' F R' U F' M' F U' R F' U r'
//Corners
U' R F L2 F' R F L' F' R F L' F' R U
F D' F' L2 F L2 D L2 F' L2
67 STM
In **Yo Notation**:

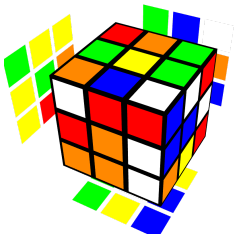
Reconstruction (3-style UF/UFR)

Edges
[M, U' R' U]
[M' U' L' : [E', L2]]
[U' D' R : [E', R2]]
[U' R : [E, R2]]
[S : [R E' R', U']]
[M U' : [M', U2]]
R U' F R' U F' M' F U' R F' U r' //Flipping alg

Corners
[R U' : [R' U R, D]]
[U' R' U : [R U' R', D]]
[U R U' :[U',R D' R']]
[U' D' : [R D' R', U2]]
119 STM

.

2. D R2 D2 R' U2 D' B U2 L' U' R2 U' R2 B2 R2 B2 R2 D' L2 F' Rw2 Uw'



Reconstruction (5-style UF/UFR)

My lettering scheme
Edges: **NDLJ BPVG TKAE**

Corners : **UCOS FNGB WX**

Speffz :
Edges: **NURF AJVM XHED**
Corners : **WBHU FGEA PV**

// Edges
F R F' R E R' F R' E' F'
S U S' U' R' U' S U R S'
D L' F E F2 E2 F L F' E F D'
// Corners
L' D R' F2 R D2 R' D2 F2 D' L D' R D
[R' U' : [D, R U R']]
R' D L' D L' D R U' D' F2 U L2 D2 // 3 corner twist alg
71STM

In **Yo Notation:**
Reconstruction (3-style UF/UFR)

Edges:
[F : [R2, E]]
[L U L', E']
[R' U' : [S, R2]]
[S', R' F R]
[u L : [E, L2]]
F R S R' F' M F' L E' L' F M' //Flipping alg
Corners:
[U R D' R' : [R' D R, U']]
[D' R D U' : [R' U R, D2]]
[R' U' : [D, R U R']]
R' D L' D L' D R U' D' F2 U L2 D2 // 3 corner twist alg
107 STM

3. D' L2 R2 D2 B' L2 B2 F' U2 R2 U' F2 R B U2 L2 U2 L2 F' Rw Uw'

Reconstruction (5-style UF/UFR)

My lettering scheme
Edges: **FAPV NLCI TREC**+ parity
Corners : **GXOJ QSMF EC**

Speffz :
Edges: **QDJV NRBL XSIB**+ parity
Corners : **EVHN TULF IB**

//Edges
S R' S R S' R' U S U' S2 R
U R' F' D' L S' L' S D F R U'
M' U' S M' U' M U S' U M
U R U' F R' U F' M' F U' R F' U r' U'

// Corners
U R2 D2 R' D2 R U' R2 U R' U' D2 R2 D2 R
[D : [R D' R', U']]
D2 R2 U F2 D' U' L D R' D R' D L'
// J Perm
100STM

In **Yo Notation:**

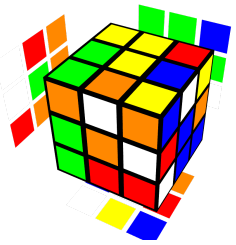
Reconstruction (3-style UF/UFR)
Edges:
[L U L' U', M]
U R' U' R' U R U R U R' U2
[E, L U L']
[U', R E2 R']
[D L F' : [E', L2]]
[L F' L', S']
U R U' F R' U F' M' F U' R F' U r' U' //Flipping alg

Abhijeet

Corners:
[D' R : [R D2 R', U]]
[U R : [R D' R', U2]]
[R U : [R U R', D]]
[D R' U' : [R U R', D']]

[R : [R D R', U2]]
//J perm
144STM

4. F2 L2 D2 R2 F2 R2 B2 U L2 F2 L B' R' B D' B2 L' F2 R' D Fw'



Reconstruction (5-style UF/UFR)

My lettering scheme
Edges: **XFEL TORG VMJT**
Corners : **BANP BXRW CI**

Speffz :
Edges: **KQER XPSM VTFX**
Corners : **ADGS AVOP BQ**

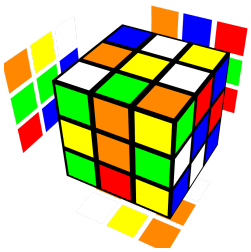
// Edges
F' U' M D' L E' L' U S d F
F R' S R' D' S' D S R S' R F'
F U' R' D' R' D R U F D F2

//Corners
U' D' L U' L' D L U D L' U L D' L'
U2 L' D2 L U2 D' R2 D L' D' R2 D' L
75STM

In **Yo Notation:**

Reconstruction (3-style UF/UFR)
// Edges
[U' : [S, R' F' R]]
[L' u L' : [E, L2]]
[U E L' : [L2,E]]
M2 U M U M' U M U M
R' U R U R U R' U' R' U'
[U E L : [E', L2]]
//Corners
[R' D' U' : [U', R' D R]]
[U D : [R U' R', D]]
[D R D' R' : [R' D R, U2]]
[R D' : [R' D R, U]]
// Sune + Anti-sune
127STM

5. U2 L B R' U2 R' F L F2 R D' F2 D F2 D' F2 D L2 D' B2 D' Rw Uw



Reconstruction (5-style UF/UFR)

My lettering scheme:
Edges: **JELM KTPQ CUGS DBVC**+ parity
Corners: **ENPH VC**

Speffz:
Edges: **FERT HXJG BWMO UAVB**+ parity
Corners: **IGSR KB**

// Edges
R' S L' U' S2 U S U' S U L S' R
R' D2 L' D M D' L D' M' D' R
F' R E R' D' R' E' R D F
D' S2 D' M S D S M D

//Corners
L2 F' L F L F2 U2 L' F L' F' L2 U2 F2
U2 F' U F L' F' U L U L' U2 F L U2
71STM
In **Yo Notation:**

Reconstruction (3-style UF/UFR)

//Edges
[S : [R' E R, U']]
[R U' R : [E', R2]]
[u L' : [E, L2]]
[L' E' : [E', L U L']]
[M2, R U' R' U]
[F' R : [E, R2]]
[M', U2]
[R U R' : [S, R2]]

//Corners
[D R': [R D' R' D, F2]]
[R' D : [R' U R, D2]]
D' : [U, R' D R]]
//J perm
122STM

Effectiveness?

There is about 40% improvement in the move count, with little loss on the finger trick ability of the solve. Since you may not detect any insertion sequence in some 5-cycle alg quickly, it is best if you make triggers of batch 4 moves each and memorize each 4 move block using the Yo notation. Memorizing the algorithm via the Singmaster notation is quite cumbersome and difficult.

Abhijeet

World record potential using this method: On a good 10/8 solve (9 algs), and assuming a memo of 5 seconds, total times of ~11-12 seconds are possible with execution of just 5-6 seconds.

Is 5-style better than 3-style?

5-Style is assumed to be the extension to the efficient and finger-tricky 3x3 blindfolded method of 3-style. 3-style is a really fast method. The current WR is less than 20 seconds using 3-style. 3-style for corners is already quite optimized considering finger tricks and regrips. 5-Style does not immediately triumph over 3-style when it comes to only corners, as we combine two similar corner comms (if they are coming in succession), and get a very efficient and fast execution. Cube explorer is quite slow in finding good 5 cycles for corners, and one of the reasons is due to bad orientation of one of the corners in the 4 corners that need to be cycled, the overall algorithm count is long, hence inefficient. To make 5-Style work on corners, some addendum kind of work has to be done, where 3-style is extended out in some likely and unlikely cases, and categorized, and scaled to 5 cycles fully. Currently UFR corners are faster on the execution as compared to 5-style UFR edges.

6. How to implement this method in your own solves?

The number of unique edge cases in 5-cycles is a whopping **126,720** cases. That's a lot to be honest. These many cases will take a lifetime to learn if you try to learning compute your alg learning time.

If you look into a normal scramble, not everything is a 5-cycle, there are also a few 4-cycles which form due to going into cycle breaks or edge targets finishing up during a trace [Form: ABAC]or going into the parity setup [Form: ABCA].

For corners, using 5 cycles is still questionable, as there are only 8 corners in a 3x3, and 7 targets in a normal setup, so it is best to solve them using [R U D] 3-style algs from the most optimal buffer UFR.

The occurrence of any given letter quad in a scramble is extremely sparse. And there is no way to deal with this sparsity than to be prepared for every case beforehand.

The chance that any letter quad comes up again in a solve in edge memo is 3/126720 = 1/42240 = **0.002367%**.
The chance that any letter quad comes up again in a solve in corner memo is 2/68040 = **0.002939%**.

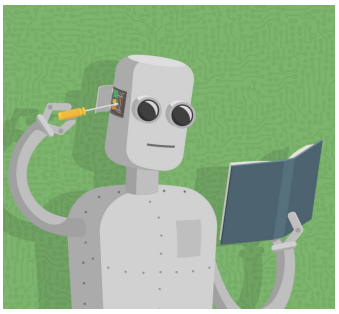
This is just insane sparsity that a normal human being just cannot handle. You need a ton of patience developing each of the letter quad, which has a contribution of only 5/194760 = **0.002567%** in the entire picture!!!

“How long before I master this method?”

There is no estimate on how long it will take, it all depends on the effort you put in, and the focus you garner up as you take a deep dive into this method. The best way to implement this

method right away is to always do some solves with it. And the one way of learning is by deliberate learning (getting very analytical after each solve , on what all things you did and how you improve on it).

7. My current motivation



Currently, I am doing Machine learning in huge sized image cube data. And generally, the number of classes that the model has to classify is ~10,000 classes.Once I wondered, if I am training models using GPU to distinguish between hundreds of thousands of classes, why would I not do the same for making memorization of a cube easier.

In a cube we similarly create thousands of categories, each consisting of a unique 4 letters combination. The think-ahead becomes clearer in a solution because of this categorization.

Another source that spurred me to stick with 4 letter combinations is the Indian art of tabla percussion. In tabla, there is a rich verbal language to represent the rhythmic sounds that the surface of both the drums makes. In that, there are a lot of divisions and basic counting mathematics, that makes it possible to have 16 beat or 10 beat cycles. In tabla playing, there is a concept called the 'rela' which involves playing at very high speeds, with as much as 4-8 sounds in one count of the rhythmic cycle. This gave me the idea of having an impulse of 4 letters at once while memorization too. For those who did not understand the 'tabla instrument' analogy I gave about, I also claim to compare the algorithm complexity of 5-style with the western classical music instrument of the piano in the Western World. A famous pianist tries to bring in a lot of abstract emotions in his/her playing. There are no 21+53 or 480 or 500 set pieces that they have and the number combination of the notes they produce goes into the hundreds of thousands.

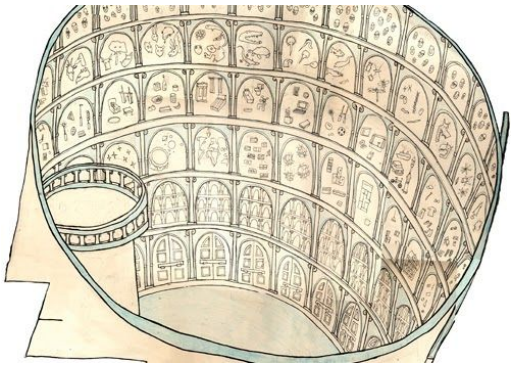
8. How does the method work? (Types of cycles and swaps that emerge through this method)

We know that for a 3-cycle on a cube there are several types of commutators we form out of it where each one of them is derived (I will not be deriving it here as it is a bit more mathematical) They are classified as:

1. Pure Commutator
2. A9
3. Cycle Shifts
4. Columns
5. Per Specials
6. Orthogonals

Reference to this section:
https://www.speedsolving.com/wiki/index.php/Beyer-Hardwick_Method
https://www.speedsolving.com/forum/threads/bh-tutorial.12268/

9. ADVANTAGES: 5-Style increases the Multiple Blindfolded (MBLD) score



I previously made a video on how to memorize algorithms that do not have triggers and how to memorize these algorithms using Memory techniques. The main motivation behind making 5-Style for me, is to make big MBLD attempts more seamless as this event is so information and memorization loaded, and also accuracy has to be spot on throughout the attempt. This method is particularly very useful for the MBLD event since there is no restriction on the upper limit of the number of cubes, the complexity of memorization and execution can be increased to improve in the MBLD event.

10. Future Scope

Get all the top BLDers to contribute to this mammoth alg database, and to make a lot of videos classifying these algorithms , and making new finger tricks, a few new types emerge.. You can use a bluetooth cube too to drill out and time all these algorithms.
https://briefcubing.com/?enable-5-style

I also have created documentation for the Yo notation used to encode the non-intuitive 5-Style algorithms, and I have also created a documentation for Letter Quads. There will have to be new techniques of remembering algorithms, without the involvement of cramming or muscle memory. There is a technique which I have developed which uses Yo notation to memorize the algorithm in batches of 4 moves. So, algorithm memorization will involve a lot of metacognition.

e.g., remembering algorithms via triggers will work in the case (oiag) : [U : [M,F]]
but not in the case (dula): F' U' F D' F' U R' D' R U R' D R U' F D, which have some 3 move insertion in its sequence but no set triggers or [A,B] inside it.

Reducing transitioning effort from a 3-Style repertoire to 5-Style system:

To make the transition path easier, we need to compare whether the tradeoff of 2 shorter 3-style algs is better or one 5-Style is better for all the cases. Eg, the hypothesis that for the 3x3 corners, the margin of move count difference is less for 3-style and 5-style. So, using 5-style for corners is not preferable.

5-Style system for FMC event:
If the 5-cycle is completely made and introduced, then it is very useful to use in FMC events. FMC solvers generally do efficient 2x2x3 block building, get a skeleton and do L5C without trying to look for some lucky insertion to reduce L5C.
If we already know the L5C algset, we focus on block-building in the FMC attempt and do a 5 Corner insertion somewhere in between the solution using a 5-cycle algorithm (~move length 10-16). But at the end of the day, the method of DR or domino reduction or using fancy block building with insertions is much easier to learn and perform for the FMC event.

5-Style for BigBLD:

Making 5-Style algorithm to 4BLD wing algorithms (centers being preserved)
The length of each 5-cycle algorithm for wings on big cubes will shoot up in move count as many slice moves cannot be used in tandem, they will not be center safe.

Also, I have been working on making new comms for x-centers which are 5-cycles, the progress has been documented here: Link

11. How to scale this method and make it complete and well verified?

The letter quads sometimes feel like feature engineering in old Machine learning terms, with a lot of toiling into making the data labeled and complete. The best way to memorize a 3x3

scramble is to not use 2-letter or 4-letter, but do pattern abstract comprehension on the 3x3 (piecewise or sticker wise pattern making). Many new kinds of finger tricks coming out of 5-Style algs need to be analyzed. Because many of the move-sequences are different from the well known CFOP triggers and new ways of finger tricking will be found out.

12. Plan to Learning 5-Style method in 5 Years from scratch

5-Style in 5 years!

12. Creator of this algset

Hi, I am Abhijeet.

I am a Master's student in Machine Learning and planning to do a Ph.D. in Theoretical Astrophysics sometime later. I have been speedcubing for over 7 years and I know how to solve a Rubik's Cube since 2008. I procrastinated a lot before finally deciding to commit to this method. Since I am already a few years into it, I will not be looking back. If this method fails to improve over 3-style, the side-fruit that I will cherish over is the extensive use of letter quads in my memo.

Contact me at mail ID: 5stylerepertoire@gmail.com

Previous versions of 5-Style doc here:

1. **Version 1**
2. **Version 2**
3. **Version 3 (DF comparison removed)**

SS Forum dedicated to the discussion of 5 cycles:
https://www.speedsolving.com/forum/threads/5cyles.61725/