World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:16, No:7, 2022

The Possibility of Solving a 3x3 Rubik's Cube under 3 Seconds

Chung To Kong, Siu Ming Yiu

***Abstract*—**Rubik's cube was invented in 1974. Since then, speedcubers all over the world try their best to break the world record again and again. The newest record is 3.47 seconds. There are many factors that affect the timing including turns per second (tps), algorithm, finger trick, and hardware of the cube. In this paper, the lower bound of the cube solving time will be discussed using convex optimization. Extended analysis of the world records will be used to understand how to improve the timing. With the understanding of each part of the solving step, the paper suggests a list of speed improvement technique. Based on the analysis of the world record, there is a high possibility that the 3 seconds mark will be broken soon.

***Keywords*—**Rubik's cube, convex optimization, speed cubing, CFOP.

I. Introduction

THE 3x3 Rubik's cube world record has been broken many times in this decade. Up to 2020, the fastest time is 3.47 seconds by Du Yusheng. Fig. 1 is the history of the world record in this decade.

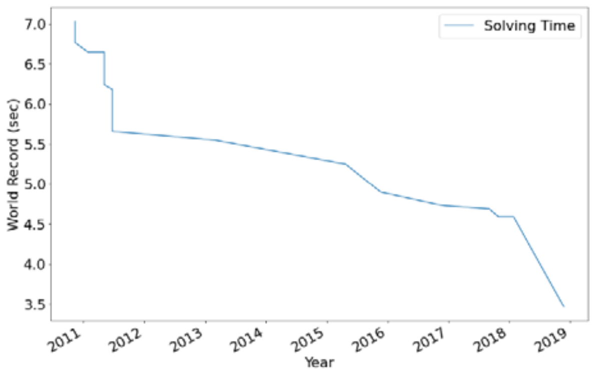

Fig. 1 World Records trend in this decade (2010-2020)

In the World Cube Association (WCA) competition, speedcubers have 15 seconds to inspect the scrambled cube. The 15 seconds inspection time is used for planning the moves ahead. Then, two hands have to be placed on the timer. When the hands leave the timer, speedcubers solve the cube as fast as possible and then hit the timer to stop the counting [1].

In this paper, the background of cubing will be introduced. Then, the total solving time will be broken down into different parts for further analysis. By using the world record of 3.47 seconds, we will investigate which parts have room for improvement and which parts are the limitation of the human and the nature. Then, we predict if it is even possible to break the sub-3 (under 3 seconds) record.

In the second part, other than a typical 2-hands 3x3 solving, one-hand solving will be discussed using the fastest one-hand record of 6.82 seconds [2]. By looking at the difference between the two solves, there is new evidence to show that 2-hands solving can have a high chance of breaking the 3 seconds' limit.

In the third part, the competition rules and the timer hardware will be discussed. If the regulations can be modified, the solving timing will be improved. Also, by expanding the domain of human recognition, a solving method will be discussed.

II. Background

There are 43,252,003,274,489,856,000 combinations in a 3x3 Rubik's cube. Many explanations in the internet try to come up with this value, and here is one of the best explanations [3]. There are 6 center pieces, 8 corner pieces and 12 edge pieces. The 6 centers are fixed. Each corner piece has 3 orientations; each edge piece has 2 orientations.

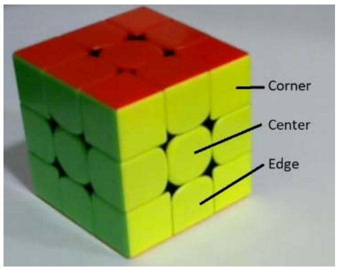

Fig. 2 Center, Corner and Edge of the 3x3 cube

Total combination = corner combination x edge combination

Let us place the corner pieces one by one. There are 8 positions the first corner piece can be placed; 7 positions the second corner piece can be placed. Therefore, there are a total of 8! placement combinations for 8 corner pieces. For each corner, there are 3 orientations. However, for the last corner, the orientation is fixed. Imagine if the cube is solved, if one corner is twisted by 120°, or 240°, the cube is insolvable. Figs. 3 and 4 show the two insolvable cases. Therefore, the total orientations of the corner pieces = 3^7 only.

corner combination = No. of placement x No. of orientation

$$\text{corner combination} = 8! * 3^7$$

Chung To Kong and Siu Ming Yiu are with Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: u3541219@connect.hku.hk, smyiu@cs.hku.hk).

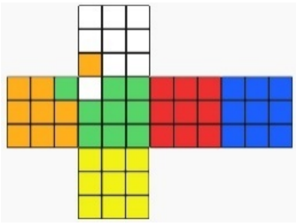

Fig. 3 Corner twisted 120°

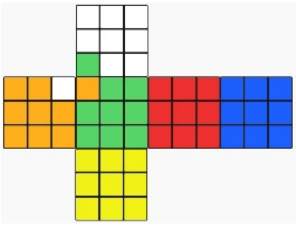

Fig. 4 Corner twisted 240°

Similar to the corner pieces, there are 12 positions the first edge piece can be placed, 11 positions for the second edge piece. Therefore, the total placements should be 12!. However, imagine for a solved cube, if two edges pieces are swapped, the cube is insolvable. So, 12! has to be divided by 2 in order to eliminate any 2 edges-swapping. Fig. 5 shows the insolvable edge swap case.

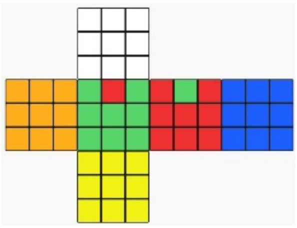

Fig. 5 Edge swap

For the edge rotation, each edge piece has 2 orientations, and the last piece edge orientation is fixed. Therefore, the total edge orientations are 2^{11}. Fig. 6 shows the insolvable edge flip case.

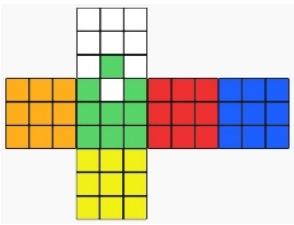

Fig. 6 Edge flip

edge combination = No. of placement x No. of orientation
edge combination = $12!/2 * 2^{11}$

When the two combinations multiply together, here is the equation:

Total combination = corner combination x edge combination
= $(8! \times 3^7) \times (12!/2 \times 2^{11})$
= 43,252,003,274,489,856,000

It is a really big number, but then, a God's number is found. All combination of a scrambled cube can be solved by a maximum of 20 moves. In July 2010, the upper bound of 20 moves was found with the help of Google Cloud computation power [4]. Also, the lower bound 20 of moves is defined by the Superflip scrambled cube [5]. This magic number of 20 will be used in this paper.

Fig. 7 Superflip

The motivation behind speed cubing is mainly about fun and speed. Since the solving time is closely related to the cube hardware, research and developing the cube hardware is a big industry. A magnetic cube can cost over $70 US. Cubing is like sports, but instead of going to the field to practice, anyone can practice at home. The potential market is enormous. Many companies are behind the international and local cubing competitions these days. For example, in 2018, there was a prize money of $13000 US for anyone who could solve a 3x3 cube below 4 seconds [6].

There are many methods of solving a Rubik's cube such as CFOP, ROUX, ZZ, Petrus, etc. [7]. The majority of speedcubers use CFOP nowadays. Although CFOP requires more moves, this method has a better look-ahead and more algorithms that speedcuber can memorize. It is still a big debate about which method is better. Based on the recent analysis of using 4718 solves in speedcube.db.com, over 4646 solves used CFOP method [8]. In this paper, the CFOP method will be used.

CFOP means Cross-F2L-OLL-PLL. These four intermediate steps make the solving easier for humans to understand. First, a cross is made, then, the F2L (first 2 layers) is constructed. OLL is Orientation of the Last Layer. And the PLL is the Permutation of the Last Layer. At the end, after the PLL, the upper layer may not align, an AUF (Adjust Upper Face) (not shown in Fig. 8) is used to align the upper layer [9].

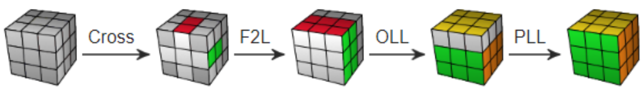

Fig. 8 CFOP step-by-step illustration

Between each step, the average person tends to pause and think of the next step, while the speedcuber can pipeline this thinking step. In general, "look-ahead" is used to describe "to plan the move in advance". Even when the pieces move really fast, a person has the temporary storage of the vision called "Iconic memory" [10]. Many speedcubers use this technique to perform look-ahead.

For the notation of the moves, there are six sides, Upper (U), Down (D), Right (R), Left (L), Front (F), Back (B). The default turn direction is clockwise 90. For example, R means right side turns 90° clockwise, R' means right side turns 90° anticlockwise. Fig. 9 is the full notation diagram [11].

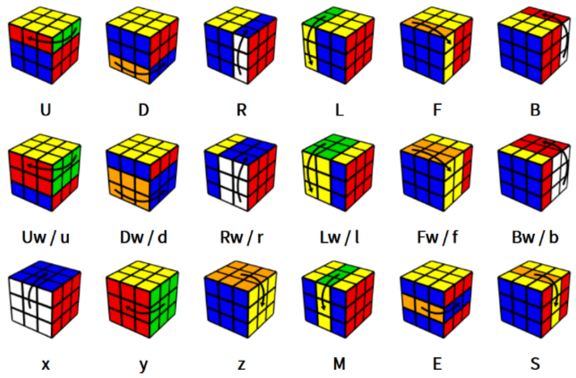

Fig. 9 Cube notation

In the WCA competition, each speedcuber has 15 seconds inspection time to look at the cube before solving [1]. Many speedcubers can visualize the cross and the first F2L. During the solve, look-ahead is the essential skill to skip the idle thinking time.

Other than look-ahead, turning fast is an essential skill too. If a sequence of moves is worth to be memorized, this sequence of moves is called an algorithm. The algorithm should be performed with the fluent finger movement in order to increase the turning speed. The turning speed can be increased with more practice. Some top speedcubers can have 10 turns per seconds [12]. No conscious mind is required to control the movement of the fingers. This unconscious muscle control is called muscle memory [13].

III. Two-Hands World Record Analysis

The fastest time of solving the 3x3 cube is 3.47 seconds. This section of the paper will analyze how to get this fast time. And then, the convex optimization problem will be formulated with different factors and hopefully, the time can go below 3 seconds.

Here is the scramble:

F U2 L2 B2 F' U L2 U R2 D2 L' B L2 B' R2 U2

Here is the solve:

White front and red bottom
U R2 U' F' L F' U' L' // Double X-Cross
U' R U R2 U R // 3rd Pair
U2 R' U R // 4th Pair
U R' U' R U' R' U2 R // OLL(CP)
U // AUF
The solve is 27 moves [14].

The total solving time is broken down as follows:

$$time_{total}=time_{starting}+time_{process}+time_{ending}$$

Here are some terms that will be used in this paper.

- $time_{total}$ = the total time for solving the cube.
- $time_{starting}$ = time between the timer starts and making the first move.
- $time_{process}$ = time between the first move and the last move.
- $time_{ending}$ = time between the last move and the timer stop.

$time_{process}$ is broken down further into this formula.

$$time_{process}= time_{cross}+ time_{CF\_thinking}+ time_{F2L}+ time_{FO\_thinking}+ time_{OLL}+ time_{OP\_thinking}+ time_{PLL}+ time_{PA\_thinking}+ time_{AUF}$$

$time_{cross}$ = time for the making cross (or X-cross); $time_{CF\_thinking}$ = idle time between Cross and F2L; $time_{F2L}$ = time for the F2L; $time_{FO\_thinking}$ = idle time between F2L and OLL; $time_{OLL}$ = time for the OLL; $time_{OP\_thinking}$ = idle time between OLL and PLL; $time_{PLL}$ = time for the PLL; $time_{PA\_thinking}$ = idle time between PLL and AUF; $time_{AUF}$ = time for the AUF.

For the total idle time,

$$time_{thinking}=time_{CF\_thinking}+time_{FO\_thinking}+time_{OP\_thinking} +time_{PA\_thinking}$$

tps = turn per second = $(number\ of\ move)/(time_{process})$. Note: in this paper, the $time_{process}$ is used to calculate the tps. $move$ = number of moves using half turn metric (HTM) [15].

Here is the complete formula of total time.

$$time_{total}=time_{starting} + time_{cross} + time_{CF\_thinking} + time_{F2L} + time_{FO\_thinking} + time_{OLL}+ time_{OP\_thinking}+ time_{PLL} + time_{PA\_thinking} + time_{AUF} + time_{ending}$$

After the formula is defined, based on the 3.47 seconds world record, here is the breakdown of the total timing (in sec) [16].

$time_{starting}= 0.35$

$time_{cross} = 1.34 – 0.35 = 0.99$
$time_{CF\_thinking}= 0$
$time_{F2L}= (1.73-1.34) + (2.41-1.73) = 0.39+0.68 = 1.07$
$time_{FO\_thinking}= (2.70 -2.41) = 0.29$
$time_{OLL}= (3.21-2.70) = 0.51$
$time_{OP\_thinking}= 0\ (skip)$
$time_{PLL}= 0\ (skip)$
$time_{PA\_thinking}= (3.32-3.21) = 0.11$
$time_{AUF}= (3.40-3.32) = 0.08$

$time_{ending}=3.47-3.40 = 0.07$

Recall our equation of $time_{process}$:

$$time_{process}= time_{cross}+ time_{CF\_thinking}+ time_{F2L}+ time_{FO\_thinking}+ time_{OLL}+ time_{OP\_thinking}+ time_{PLL}+ time_{PA\_thinking}+ time_{AUF}$$

$$time_{process}$$
$$=0.99+0+1.07$$
$$+0.29(time_{FO\_thinking})+0.51+0+0$$
$$+0.11(time_{PA\_thinking})+0.08$$
$$= 2.65(time_{solving}) + 0.4(time_{thinking})$$

$= 3.05$

Here is the summary:

$time_{starting} = 0.35$
$time_{process} = 2.65(time_{solving}) + 0.4(time_{thinking})$
$time_{ending} = 0.07$

His tps (turn per second) is total *move/* $time_{process}$ = 27/3.05= 8.85 (exclude $time_{starting}$ and $time_{ending}$)

Fig. 10 3.47 seconds Two-hand World Record timing breakdown

```
# Import packages.
import cvxpy as cp
import math

log_tps= cp.Variable()
log_move= cp.Variable()
log_process_time= cp.Variable()
log_move_min = math.log(27)
log_tps_max = math.log(8.85)
starting_time_min = 0.35
ending_time_min = 0.07

# Form objective.
obj = cp.Minimize(log_process_time)

# Subject to
constraints = [log_move >= log_move_min,
               log_tps <= log_tps_max,
               log_process_time == log_move - log_tps]

# Form and solve problem.
prob = cp.Problem(obj, constraints)
prob.solve()
print("status:", prob.status)
print("optimal value", prob.value)
process_time = math.exp(prob.value)
print("process_time = ", process_time)
total_time = starting_time_min + process_time + ending_time_min
print("total_time = ", total_time)

status: optimal
optimal value 1.1154194069841579
process_time =  3.0508474576261024
total_time =  3.4708474576261024
```

Fig. 11 Code to verify the 3.47 seconds two-hands world record

In Fig. 10, the time started from the blue "Starting Time", through the red "Processing Time" and ended with the green "Ending Time". Other than the light red solving time, a high portion of time is used in the starting time and the thinking time.

A CVXPY python convex optimization program was used to verify the formula and tps.

Minimize:	*log_time_process*
subject to:	*log_move >= log(move_min)*
	log_tps <= log(tps_max)
	log_time_process = log_move – log_tps

$$time\_total = time\_starting\_min + e^{log\_time\_process} + time\_ending\_min$$

where *move_min=27, time_starting_min=0.35, time_ending_min=0.07 and tps_max=8.85*

The world record of 3.47 seconds is obviously higher than 3 seconds. Below is a list of improvements by tuning different parameters.

Eliminate Thinking Time

For the $time_{process} = 2.65(time_{solving}) + 0.4(time_{thinking})$, if the thinking time of 0.4 sec can be pipelined, which means 0.4 seconds can be eliminated, $time_{process}$ becomes 2.65 seconds:

$time_{total} = time_{starting} + time_{process} + time_{ending}$
$= 0.35 + 2.65 + 0.07$
$= 3.07$

The time is still higher than 3 seconds.

Increase Turning Speed

For a simple estimation, all parameters would be kept the same except the tps. Using the convex optimization program to find the minimum turning speed to be 10.47. This tps is still within the human limit [12].

tps = move/(3seconds -time_starting_min + time_ending_min)

```
In [6]: import cvxpy as cp
        tps= cp.Variable()
        constraints = [tps >= 27/(3 - 0.35 - 0.07)]
        obj = cp.Minimize(tps)
        prob = cp.Problem(obj, constraints)
        prob.solve()
        print("optimal value", prob.value)

        optimal value 10.465116279068681
```

Fig. 12 Code to get 10.46 tps for reaching the 3 seconds mark based on the two-hands world record

From the above two examples, a skillful speedcuber with 10.47 tps is essential to achieve fast solve. According the recent analysis using 4718 solves, tps above 12 is possible [8]. However, it is not easy to have this high tps. The other limitations are the $time_{starting}$, and the $time_{ending}$. Also, the number of moves is one of the main factors too.

God's Number

All combinations of the cube can be solved by maximum 20 moves using half turn metrics (HTM) [15]. It has been proved by using Google Cloud computation [4].

In the following test, the God's number is used in world record with other parameters kept the same:

Time = 0.35 + 20/8.85 + 0.07
Time = 2.68 seconds
time_total = time_starting_min + move/tps+ time_ending_min

```
In [14]: import cvxpy as cp
         import numpy as np
         t= cp.Variable() # total time
         move= cp.Variable() #number of move
         constraints = [move >= 20,
                        t == (0.35 + move/8.85 + 0.07)]
         obj = cp.Minimize(t)
         prob = cp.Problem(obj, constraints)
         prob.solve()
         print("optimal value", prob.value)

         optimal value 2.679887006591868
```

Fig. 13 Code to get the time of 2.67 sec if the 20 moves are used based on the two-hands world record

2.67 seconds has broken the 3-seconds mark; however, the God's number is not considered to be finger friendly (i.e. finger trick) [17]. For example, making F or B moves are considered not finger friendly as the hand grip has to change before making the move. And changing hand grip can increase the risk of dropping the cube.

The good news is that God's number is the maximum move that can solve any cube combination, so there is a chance of getting lower move counts of 20. If we put the finger trick into consideration, the total number of moves will be bigger. So, 27 moves in the world record are reasonable. However, the meaning of a scrambled cube is not clear in the general. If the cube is one move away from solving, then, for sure everyone can solve the cube in less than 3 seconds, or even one second.

TNoodle is the software to generate the random scramble for the WCA competition.[18] This software has the mechanism to check if the scramble is too easy to solve, i.e. too few moves. However, in the WCA regulation, Article 4: Scrambling, 4b3 (Specification for a scramble program), it is written as the following, "An official scramble sequence must produce a random state from all states that require at least 2 moves to solve" [19]. Statistically, more low-move counts scramble cubes will appear as speedcubers attend more competitions. Speedcubers should prepare themselves with essential skill set (the skill set will be mentioned in the end of this paper) in order to seize any low move counts opportunity.

Improving the Starting Time

From the above discussion, increasing tps is not easy for a speedcuber; reducing move counts requires luck. For the world record, by comparing the starting time with the ending time, it seems shortening the starting time may give us a chance of achieving the 3 seconds goal. If the starting time is equal to the ending time of 0.07, the calculation is shown in Fig. 14.

time_total = time_starting_min + move/tps+ time_ending_min

The time is 3.19 sec which is still bigger than 3 seconds. Now, we also increase the tuning speed to 9.5 tps.

The time is 2.98 which is below 3 seconds.

In YouTube, there is one example from a youtuber called LaZer0MonKey. He did many practices of the same moves, and he could get 3.008 seconds [14]. And now, sub-3 record can be achieved in theory with some small tunings of starting time and tps. In the section below, one-hand world record will be used to investigate the possibility of breaking the 3-seconds mark.

```
In [27]: import cvxpy as cp
         import numpy as np
         t= cp.Variable() # total time
         move= cp.Variable() #number of move
         constraints = [move >= 27,
                        t == (0.07 + move/8.85 + 0.07)]
         obj = cp.Minimize(t)
         prob = cp.Problem(obj, constraints)
         prob.solve()
         print("optimal value", prob.value)

         optimal value 3.1908474585693356
```

Fig. 14 Code to get the time of 3.19 sec if the starting time is reduced to 0.07 second based on the two-hands world record

```
In [29]: import cvxpy as cp
         import numpy as np
         t= cp.Variable() # total time
         move= cp.Variable() #number of move
         constraints = [move >= 27,
                        t == (0.07 + move/9.5 + 0.07)]
         obj = cp.Minimize(t)
         prob = cp.Problem(obj, constraints)
         prob.solve()
         print("optimal value", prob.value)

         optimal value 2.982105264162378
```

Fig. 15 Code to get the time of 2.98 sec if the starting time is reduced to 0.07 second and tps is increased to 9.5 based on the two-hands world record

IV. One-Hand World Record Analysis

For one-hand, the world record is held by Max Park with 6.82 seconds [2], [20].

Scramble: R2 B2 L' U2 L R' D2 U2 R' B U' R F2 L B L2 D' R' F U'
Red cross, blue front
D U' L R Uw' U R // Cross
L' U L // 1st Pair
U2 R' U' R U L U' L' // 2nd Pair
U R' U2 R U' R' U R // 3rd Pair
y' R' U R U' R' U' R // 4th Pair
Rw U R' U' L' U R U' // OLL
41 moves

$time_{starting} = 0.15$
$time_{cross} = 1.998 - 0.15 = 1.848$
$time_{CF\_thinking} = 0$
$time_{F2L} = 5.164 - 1.998 = 3.166$
$time_{FO\_thinking} = 5.498 - 5.164 = 0.334$
$time_{OLL} = 6.500 - 5.498 = 1.002$
$time_{OP\_thinking} = 0$
$time_{PLL} = 0$ *(skip)*
$time_{PA\_thinking} = 0$
$time_{AUF} = 0$

Recall our equation of $time_{process}$:

$$time_{process} = time_{cross} + time_{CF\_thinking} + time_{F2L} + time_{FO\_thinking} + time_{OLL} + time_{OP\_thinking} + time_{PLL} + time_{PA\_thinking} + time_{AUF}$$

$time_{process}$
$= 1.848 + 0 + 3.166 + 0.334 + 1.002 + 0 + 0 + 0 + 0$
$= 6.016(time_{solving}) + 0.334(time_{thinking})$
$= 6.35$

$time_{ending}$=6.82-6.5 = 0.32

His *tps (turn per second)* is *total move/ $time_{process}$:*

= 41/6.35= 6.457 (exclude $time_{starting}$ and $time_{ending}$)

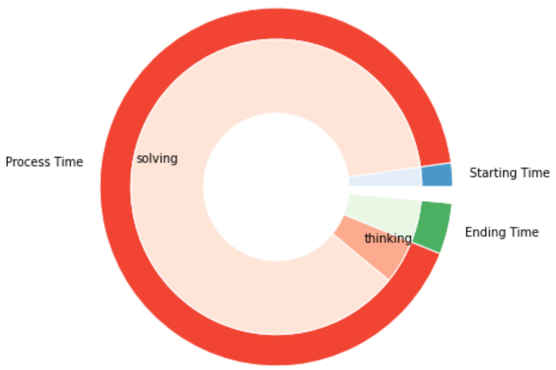

Fig. 16 6.82 seconds One-hand World Record timing breakdown

A convex program is used to verify the formula and timing.

time_total = time_starting_min + move/tps+ time_ending_min

```
In [12]: import cvxpy as cp
         import numpy as np
         t= cp.Variable() # total time
         move= cp.Variable() #number of move
         constraints = [move >= 41,
                        t == (0.15 + move/6.457 + 0.32)]
         obj = cp.Minimize(t)
         prob = cp.Problem(obj, constraints)
         prob.solve()
         print("optimal value", prob.value)

         optimal value 6.8196980028692185
```

Fig. 17 Code to verify the 6.82 second one-hand the world record

As we can see, the starting time is 0.15 second which is much faster than the 2-hands world record. However, for the ending time, it is 0.32 second which is much slower than the 2-hands world record.

Because of the high move counts and relatively slow tps, one hand timing is close to impossible to be sub-3. But with some assumptions, we will try to reach the goal in an ideal case. Assuming a 3-seconds solve in the one-hand world record, the tps has to go up to 16.02 which is over the human limits [12].

tps = move/(3seconds -time_starting_min + time_ending_min)

```
In [11]: import cvxpy as cp
         tps= cp.Variable()
         constraints = [tps >= 41/(3 - 0.15 - 0.32)]
         obj = cp.Minimize(tps)
         prob = cp.Problem(obj, constraints)
         prob.solve()
         print("optimal value", prob.value)

         optimal value 16.20553359683695
```

Fig. 18 Code for reaching the 3seconds mark, the tps has to go up to 16.2

The limiting factor is the high move counts in the one-hand solve. The one-hand move set is limited to R/R', L/L', U/U', D/D', Rw/Rw' and Uw/Uw' without rotation. And the time of rotation is longer as there is a risk of dropping the cube. So, it is closed impossible for human to solve a 3x3 Rubik's cube using one-hand under 3 seconds without the luck of low move counts.

V. Realistic Prediction

Based on the two-hands world record and one-hand world record, we can conclude that the starting time upper bound is 0.15 second and the ending time upper bound is 0.07 second. By combining the observation of the two-hands and one-hand world record, two-hands solving is possible with the combination of,

$time_{starting}$ <= 0.15 (from one-hand world record)
$time_{ending}$<= 0.07 (from two-hands world record)
move count <= 27 (from two-hands world record)
tps = move/(3seconds -time_starting_min + time_ending_min)

```
In [12]: import cvxpy as cp
         tps= cp.Variable()
         constraints = [tps >= 27/(3 - 0.15 - 0.07)]
         obj = cp.Minimize(tps)
         prob = cp.Problem(obj, constraints)
         prob.solve()
         print("optimal value", prob.value)

         optimal value 9.712230215826244
```

Fig. 19 Code for reaching the 3 seconds mark with starting time <=0.15 sec, ending time <=0.07 sec and move count <=27, the tps is 9.71

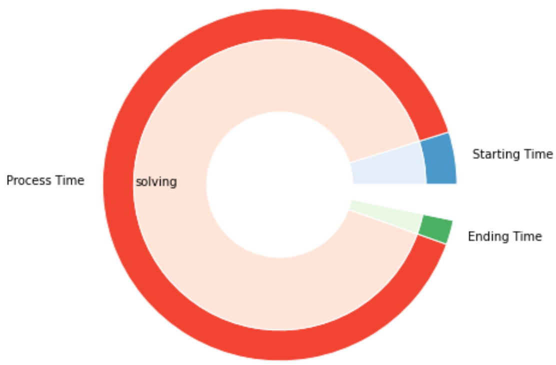

Fig. 20 Predicted 3 seconds timing breakdown

The tps is 9.71 which is within the human limit [12]. The result implies that it is possible to break the sub 3 records. Other than the human factor, the chance of getting lower move counts cube will be higher as more and more competitions happen each year.

VI. Minimum Move Counts

Low move count is mainly based on the lucky scramble. However, with some help with a computer, the chance of getting a lower move count can be higher. The God's number is 20 which means the maximum move count is 20. However, finger friendly move sequence is important to achieve high tps.

For example, RUR'U' is much faster than RBR'B' [21]. There is a duality gap between the minimum move count (d*) and the finger friendly minimum move that humans can visualize (p*). The approach is to compute a list of minimum move counts by using a computer software like "Cube Explorer" [22]. Then, the list is modified to be more finger friendly. By using the same RBR'B' example, we can rotate the whole cube -90° at the right-side axis (i.e., x' in cube notation) and it becomes RUR'U'. Therefore, RBR'B' becomes x'RUR'U'. After a list of finger-friendly minimum move counts is calculated, the next step is to consider the movement of each finger in order to obtain a high tps.

The time for a finger to move from an existing position to a destination position is the "finger traveling time". By comparing the two sequences of moves, UUUU and U'U'U'U', UUUU is slower than U'U'U'U' [23]. Although the move counts are the same, the index finger traveling times are different. Using the convex optimization term, finger traveling time is the maximum of the "finger traveling time" of each finger between two moves. Other than finger movement, how the hand holds the cube, called the gripping position, matters too. Our hand consists of 27 bones, 27 joints, 34 muscles and over 100 ligaments and tendons. It is hard to simulate the hand model in the computer. It is much easier for a human to try the whole list of the finger-friendly minimum moves and decide which one is the most finger friendly.

After achieving the finger-friendly minimum moves, human has to be able to understand the move sequence also. Human tends to group the colors together. Grouping the pieces like 2x1, 2x2, 2x3 blocks gives our mind a state of achievement and free our mind to solve the rest of the puzzle. It is the fundamental concept of the cubing methods like CFOP and ROUX. And it is a key to achieve a good look-ahead technique. In the competition, only 15 seconds preparation time gives speedcubers a big challenge to plan the whole moves sequence. The possibility of modifying the rule in the competition will be discussed in Section VII.

VII. Possibility of WCA Rules Changes and Hardware Upgrade

Longer Preparation Time

There is another competition in WCA called FMC. FMC means Fewest Move Count, cubers have 1 hour to find the fewest move count [24]. The world record is 16 moves only [25].

If this 16 moves-count is used in the two-hands world record, by using the same two-hand world record model with 16 moves counts, the result is 2.23 seconds. So, in theory, for a great speedcuber with some luck of getting 16 moves, the time can be 2.23 seconds.

$$time\_total = time\_starting\_min + move/tps + time\_ending\_min$$

So, if the WCA gives speedcubers more time to inspect the cube instead of 15 seconds, the speedcubers can probably plan out the whole solve efficiently with lower move count and high tps.

```
In [21]: import cvxpy as cp
         import numpy as np
         t= cp.Variable() # total time
         move= cp.Variable() #number of move
         constraints = [move >= 16,
                        t == (0.35 + move/8.85 + 0.07)]
         obj = cp.Minimize(t)
         prob = cp.Problem(obj, constraints)
         prob.solve()
         print("optimal value", prob.value)

         optimal value 2.2279096054619028
```

Fig. 21 Code for checking if the move count is reduced to 16 based on the two-hands world-record, the time can reach 2.227 sec

Timer Hardware

Sensitivity of the timer is also important. It is because even if the timer starts and stops immediately by a fast-moving hand, there is a minimum built-in delay. Inside the timer electronics, the timer uses capacitance sensor to detect the hands, and capacitor needs time to charge and discharge. Also, in order to prevent any double press, a small delay has to be introduced between any trigger [26], [27]. After a quick test in our laboratory, the delay is only of 0.05 seconds which can be neglected. However, using a physical time for our sub-3 goal, $time_{starting}$ and $time_{ending}$ are still the main limitations.

Fig. 22 Stack Timer in the competition

Nowadays, Bluetooth cube is getting popular. The cube has the Bluetooth electronics inside which connects to a phone app. The timer on the apps starts as the first move is performed. And the timer stops when the last move is done [28], [29]. If the Bluetooth cube can be used in the competition, the starting time and ending time will be zero. It can dramatically shorten the total solving time.

VIII. Future Study and Advice to Speedcubers

New Solving Method

Based on the world record FMC, the sequence of moves is to create many 2x1 blocks and then merge them together. It gives us a hint to develop a new method. However, if the new method is hard for human to understand and visualize, it is not practical in the competition. Nowadays, some speedcubers start to mix the solving methods together, like LFCL (hybrid of CFOP and ROUX) [30]. As each method has its own set of algorithms, learning a new method requires time and effort. It is a huge barrier as no one wants to restart the learning curve. Unless the new method is easy to adapt and has a huge benefit of low move counts, CFOP and ROUX will still be the mainstream methods.

The new generation of cubing method will probably be based on expanding our domain of recognition. For example, CFOP solved the cube using the pseudo-block [31]. One example is just a Jb-perm with a pseudo-block [32], but the solve looks totally different from other typical methods. Instead of aligning the color at the first stage like CFOP, this method moves the alignment to the end of the solve. The advantage is to increase the flexibility of block building and let speedcubers use the same algorithm set. However, this method requires a larger domain of recognition and look-ahead technique. This method is still under development and hope a new paper will be released soon.

Based on the recent analysis of nine world record solves [8], the average moves count is 32.2, tps is 9.7, XCross/XXCross [33] is 44%, PLL skip is 78%.

Advice to Speedcubers

In order to have low move counts, high tps and low starting/ ending time, speedcubers should work on the following in order to break the 3 seconds' mark.

1. Lower starting time [34]
2. Double X-cross with color neutral [33, 35]
3. Advance F2L, pseudo-slotting [36]
4. Edge control [37]
5. Look ahead for the ZBLL [38]
6. Lower ending time

Note: Double X-cross means making the cross with 2 F2L. Color neutral means solving the cube without the restriction of color. Edge orientation means controlling the edges in order to make a cross on the top layer. ZBLL is a pool of algorithms that combine the OLL and PLL together.

IX. Conclusion and Limitations

Achieving sub-3 is possible. Many speedcubers focus on the solving technique and the turning speed, however, the time of picking up the cube and hitting the timer are more critical. As more and more speedcubers join the competitions, breaking the 3 seconds mark is truly possible.

For the solving method, this paper only considers CFOP. There are other methods, like ROUX which has lower move counts. However, ROUX has harder look-ahead which means more thinking time is required [39]. But if the speedcubers can expand the domain of recognition, the low move counts ROUX has the advantage for breaking the 3 seconds' mark.

References

[1] cyotheking, "WCA Competitor Tutorial." (Online). Available: https://youtu.be/dPL3eV-A0ww?t=97.
[2] M. Park, "WR [6.82] OH Rubik's Cube 3x3 One handed World Record Single." (Online). Available: https://www.youtube.com/watch?v=LvFtz4QAP5g.
[3] J. Perm, "Why Are There 43,252,003,274,489,856,000 Rubik's Cube Combinations?." (Online). Available: https://www.youtube.com/watch?v=z2-d0x_qxSM.
[4] "God's Number is 20." (Online). Available: http://www.cube20.org/.
[5] E. Shinder, "Mathematics of the Rubiks' cube." (Online). Available: http://e-shinder.staff.shef.ac.uk/rubiks-cube.pdf.
[6] MoYu-Cube, "2018 MoYu World Record Prize Announcement," 2018. (Online). Available: https://www.facebook.com/moyumagiccube/posts/2018-moyu-world-record-prize-announcement-for-cubers-who-break-official-world-re/1580692178678403/.
[7] "List of methods." (Online). Available: https://www.speedsolving.com/wiki/index.php/List_of_methods.
[8] B. Noris, "LEARNING FROM THE PROS," 2021. (Online). Available: https://basilio.dev/cubing/recons/.
[9] "Rubik's Cube solution with advanced Fridrich (CFOP) method." (Online). Available: https://ruwix.com/the-rubiks-cube/advanced-cfop-fridrich/.
[10] M. Coltheart, "Iconic memory." (Online). Available: https://royalsocietypublishing.org/doi/abs/10.1098/rstb.1983.0055.
[11] J. Perm, "Rubik's Cube Move Notation." (Online). Available: https://jperm.net/3x3/moves.
[12] "What Are The Limits?." (Online). Available: https://www.cubeskills.com/blog/what-are-the-limits.
[13] K. Gundersen, "Muscle memory and a new cellular model for muscle atrophy and hypertrophy," *J Exp Biol*, vol. 219, no. Pt 2, pp. 235-42, Jan 2016, doi: 10.1242/jeb.124495.
[14] LaZer0MonKey, "Breakdown of Yusheng Du's 3.47 World Record Solve!." (Online). Available: https://youtu.be/qT5Jan1sLQE?t=153.
[15] "Metric." (Online). Available: https://www.speedsolving.com/wiki/index.php/Metric.
[16] D. Blender, "Yusheng Du 3.47 World Record 3D Visualization." (Online). Available: https://www.youtube.com/watch?v=JDLlEAw5WRM.
[17] "Finger tricks - How To Get Faster At Cubing." (Online). Available: https://ruwix.com/the-rubiks-cube/fingertricks/.
[18] "TNoodle." (Online). Available: https://github.com/thewca/tnoodle.
[19] "WCA Regulations." (Online). Available: https://www.worldcubeassociation.org/regulations/#article-4-scrambling.
[20] LaZer0MonKey, "Breakdown of Max Park's 6.82 One-Handed World Record Solve!." (Online). Available: https://www.youtube.com/watch?v=xlgmyzaWeeo.
[21] "RUR'U' vs RBR'B'." (Online). Available: https://youtu.be/Lh1IOekzB5U.
[22] "Cube Explorer." (Online). Available: https://ruwix.com/the-rubiks-cube/herbert-kociemba-optimal-cube-solver-cube-explorer/.
[23] "UUUU vs U'U'U'U'." (Online). Available: https://youtu.be/OTDEKqwInYA.
[24] "The Fewest Move Count (FMC) Event." (Online). Available: https://ruwix.com/the-rubiks-cube/fmc-fewest-move-count-challenge-method-tips/.
[25] "FMC rankings, World Cube Association." (Online). Available: https://www.worldcubeassociation.org/results/rankings/333fm/single.
[26] "STACKMAT™ PRO TIMER." (Online). Available: https://www.speedstacks.com/store/retail/speed-stacks-stackmat-pro-timer/.
[27] "Capacitive Touch Sensors." (Online). Available: https://www.fujitsu.com/downloads/MICRO/fme/articles/fujitsu-whitepaper-capacitive-touch-sensors.pdf.
[28] "GAN356 i2." (Online). Available: https://www.gancube.com/gan356-i2.
[29] "Go Cube." (Online). Available: https://getgocube.com/.
[30] "LFCL 3x3 Rubik's Cube Method." (Online). Available: https://docs.google.com/document/d/1nJ11uxsT96FJXnPQWd2n1eTPAWqkuxh3Wf_m1EevPck/edit.
[31] "Pseudo block." (Online). Available: https://youtu.be/5cdHQvAaiV8.
[32] "PLL algorithm." (Online). Available: http://www.rubiksplace.com/speedcubing/PLL-algorithms/.
[33] "XCross." (Online). Available: https://www.speedsolving.com/wiki/index.php/XCross.
[34] "How To Solve the Cross FASTER [Cross Boot Camp]." (Online). Available: https://www.youtube.com/watch?v=VWwj_HaPpYI.
[35] "Color neutrality." (Online). Available: https://www.speedsolving.com/wiki/index.php/Color_neutrality.
[36] CubeHead, "Pseudoslotting : Full Tutorial + Example Solves." (Online). Available: https://www.youtube.com/watch?v=xjhjdIFddCk.
[37] J. Perm, "Advanced F2L: Edge Control." (Online). Available: https://www.youtube.com/watch?v=zB5-8HAfldM.
[38] "ZBLL." (Online). Available: https://www.speedsolving.com/wiki/index.php/ZBLL.
[39] W. X. Junshen Kevin Chen, Zhouheng Sun, "DeepCube: Transcribing Rubik's Cube Moves with Action Recognition." (Online). Available: https://russellxie7.me/docs/cube.pdf.

Chung To, Kong is a PhD candidate at HKU and has worked in different industries including robotics, LED screen, FPGA, access control/CCTV, networking, and AI solution. In 2017, he was a TEDx speaker about "Maker mindset". He has received his MSc in Electrical and Electronic Engineering from HKU and his bachelor's degree in Electrical Engineering and Computer Engineering with Commerce minor from the University of British Columbia in Vancouver, Canada.

Siu Ming Yiu is a professor in the Department of Computer Science of HKU. His research interests include cryptography, cybersecurity, and FinTech.