

# **Guida al Fewest Moves**

Risoluzione del Cubo di Rubik nel minor numero di mosse  
di Sebastiano Tronto

# Indice generale

0. Introduzione.....	4
Competizioni Ufficiali.....	4
Competizioni e Varianti Non Ufficiali.....	4
Scopo della Guida.....	5
Fonti e Ringraziamenti.....	5
Contatti.....	5
1. Pensa Fuori dagli Schemi.....	6
Petrus.....	6
Roux.....	7
ZZ.....	7
CFOP (Fridrich) / FreeFOP.....	8
Keyhole F2L.....	8
Heise.....	8
Cosa e Come Imparare.....	9
Petrus.....	9
Roux.....	10
ZZ.....	10
CFOP / FreeFOP.....	10
2. Procedere nella Risoluzione.....	11
Blockbuilding.....	11
Allinea e Unisci.....	11
Spotalo.....	12
Distruggi e Ricostruisci.....	12
Keyhole.....	13
Una Mossa, Due Obiettivi.....	13
Influenzare gli Step Successivi.....	13
Occhio all'EO!.....	14
Quali Blocchi Costruire?.....	14
Blocchi già Fatti: Come Tratarli?.....	15
Trucchi per Velocizzarsi e Tecniche Avanzate.....	15
Trovare uno Skeleton.....	16
Commutatori.....	17
Commutatori di Angoli.....	17
Commutatori di Spigoli.....	18
Commutatori di Blocchi.....	19
Insertions.....	19
Insertions Semplici.....	19
Insertions Multiple: Cicli Separati (3 angoli e 3 spigoli).....	21
Insertions Multiple: 2 o 3 Angoli Flippati.....	22
Insertions Multiple: 4 Angoli.....	22
Insertions Multiple: 5 Angoli.....	24
Insertions Multiple: 5 Spigoli.....	25
Altre Insertions: 2 Angoli e 2 Spigoli.....	25
Altre insertions: Coniuga e Risolvi.....	26
Numero di Mosse Richieste (una Stima).....	27
Insertion Finder.....	27
Ripassa e modifica.....	27
Get Lucky!.....	28
Primo Esempio: Inserire l'Ultima Coppia.....	28
Secondo Esempio: Come Usare gli Algoritmi.....	28

3. Strumenti Avanzati.....	29
Scramble Inverso.....	29
Pseudo Blocchi, Premoves e NISS.....	30
NISS Inverso.....	33
Algoritmi Utili.....	33
Analisi delle Coppie.....	34
4. Come Allenarsi.....	35
No Time Limit e Simulazioni di Gara.....	35
Confrontatevi con i Maestri (e Studiate le loro Soluzioni).....	35
Scramble Difficili.....	35
Allenamenti Mirati.....	35
Scramblare velocemente.....	35
Studiate!.....	36
5. Alcune Alternative.....	37
Partire con l'EO.....	37
EO + Blockbuilding.....	37
Riduzione a “Domino” (e HTA).....	37
Corners First.....	38
6. Consigli per le Competizioni.....	39
Come Scrivere una Soluzione.....	39
Soluzione di Backup.....	39
Gestione del Tempo.....	40
Non Bloccarti!.....	40
Esplorare le Possibilità.....	40
7. Altre Risorse.....	41

# 0. Introduzione

Se tentare di risolvere il cubo di Rubik (o, in generale, un qualunque puzzle) nel minor tempo possibile è interessante, forse ancora più interessante è tentare di risolverlo utilizzando il minor numero possibile di mosse: è questo lo scopo del “Fewest Moves” (o FMC, dall'inglese “Fewest Moves Challenge”).

## Competizioni Ufficiali

Il Fewest Moves è un evento riconosciuto dalla WCA (World Cube Association), l'organizzazione che regola le gare ufficiali di cubo di Rubik e puzzles simili. Le regole ufficiali ([articolo E](#)) possono essere così riassunte:

- Lo scramble (sequenza di mosse per mischiare il cubo) viene distribuito a tutti i concorrenti in forma scritta.
- Tempo limite: 60 minuti
- Materiale consentito:
  - Carta e penna (fornita dal giudice)
  - Fino a 3 cubi di Rubik (portati dal concorrente)
  - Illimitati adesivi colorati (portati dal concorrente)
- La soluzione:
  - Dev'essere presentata in forma scritta, seguendo la notazione [OBTM](#): sono consentite rotazioni (x, y2, [u]...), mosse singole (R, U2, L', ...) e mosse doppie (Rw2, Fw', ...) ma non mosse interne (M, E, S, ...); ai fini del risultato, le rotazioni non contano, mentre tutte le altre mosse contano 1.
  - Deve essere lunga al massimo 80 mosse (contando anche le rotazioni)
  - Non deve essere in alcun modo legata allo scramble; in aggiunta, il concorrente deve essere in grado di spiegare il motivo di ogni mossa scritta.

Il miglior risultato fino ad ora mai ottenuto in una gara ufficiale è di 20 mosse di Tomoaki Okayama (Giappone) mentre il record del mondo per la media sui tre tentativi (in una sola gara) è di 25, detenuto da Sébastien Aurox (Germania) e Vincent Sheu (USA).

## Competizioni e Varianti Non Ufficiali

Gare non ufficiali online vengono tenute settimanalmente in alcuni forum (tra cui [speedsolving.com](#)) e su <http://www.ocf.berkeley.edu/~dadams/fmc>. Ovviamente, nessuno può controllare che le regole vengano rispettate, quindi tali competizioni si basano sulla fiducia reciproca tra i concorrenti.

Le più comuni varianti consistono nel “no time limit”, dove non è previsto il limite di tempo (generalmente, per praticità, il tempo limite è in realtà fissato ad una settimana) e la “linear”, dove

una volta fatta una mossa non si può annullare; non si possono quindi esplorare più soluzioni, ma bisogna scrivere la prima soluzione trovata (il tempo limite viene solitamente fissato a 10 minuti).

## Scopo della Guida

Lo scopo di questa guida è quello di riassumere tutte le migliori tecniche conosciute per riuscire ad ottenere buoni risultati nel fewest moves. In alcuni casi la spiegazione sarà dettagliata e arricchita da esempi, mentre in altri non si potrà fare altro che dare una spiegazione sintetica e suggerire altre risorse per uno studio più approfondito.

## Fonti e Ringraziamenti

Tutto ciò che troverete in questa guida viene dall'esperienza dell'autore, che è stata acquisita non solo con l'allenamento autonomo, ma anche grazie alle molte risorse disponibili gratuitamente online, la maggior parte delle quali citate al paragrafo 7 “Altre Risorse”, alle quali farò riferimento durante tutta la guida. Voglio quindi ringraziare tutta la community internazionale legata allo speedcubing per aver sempre diffuso apertamente le tecniche e i metodi sviluppati, permettendo così a chiunque di imparare gratuitamente e liberamente tutto ciò che c'è da sapere (fino ad ora). Spero con questa guida di fare la mia parte.

## Contatti

Per dubbi/domande o segnalazioni di errori potete contattarmi all'indirizzo:

*sebastiano.tronto [at] gmail [dot] com*

# 1. Pensa Fuori dagli Schemi

Qualunque sia il vostro metodo principale, porsi dei limiti imponendosi di usare solo quel metodo è la peggior cosa che possiate fare nel FMC. **Non bisogna mai limitarsi** ad un unico metodo, ma tentare di sfruttare al meglio ogni situazione.

Per esempio, supponiamo che abbiate costruito un blocco  $2 \times 2 \times 3$ ; le possibilità ora sono molte: potete posizionare l'ultimo spigolo della croce e finire l'F2L (CFOP), potete orientare gli spigoli (Petrus) oppure tentare di costruire altri blocchi in maniera libera (FreeFOP, Heise) o altro ancora. Uno qualunque di questi metodi potrebbe portarvi ad una buona soluzione e la cosa migliore da fare è quindi **tentare di usarli tutti** (o per lo meno una buona parte).

Il modo migliore per aprirsi la mente ed imparare a pensare “fuori dagli schemi” è, un po' paradossalmente, **imparare tanti “schemi”**, cioè tanti metodi. Di seguito descriverò brevemente quelli che sono secondo me i metodi più utili per il FMC, senza accennare alla loro storia o ai loro vantaggi/svantaggi nelle speedsolves. Per ognuno di questi metterò un link ad un tutorial online (in lingua inglese), ma vi suggerisco di cercare su google o su [speedsolving.com](http://speedsolving.com) maggiori informazioni. [Questa guida](#) è un'ottimo punto di riferimento per i 4 metodi più usati (CFOP, Roux, ZZ e Petrus), anche se è pensata principalmente per lo speedsolving.

## Petrus

Gli step del Petrus sono i seguenti:

1. Costruire un blocco  $2 \times 2 \times 2$
2. Espandere il blocco  $2 \times 2 \times 2$  ad un blocco  $2 \times 2 \times 3$
3. Orientare gli spigoli
4. Finire i primi 2 strati (F2L)
5. Completare l'ultimo strato (originariamente diviso in 3 passaggi)

Avere una buona abilità nel **blockbuilding**<sup>1</sup> è obbligatorio se si vuole diventare bravi nel FMC, e praticare il Petrus è il miglior modo per acquisirla. Per imparare a risolvere gli step 1 e 2 in maniera efficiente bisogna ragionare e tentare varie strade di volta in volta; fondamentale è anche guardare **ricostruzioni di soluzioni di cubers più esperti**. Per il primo step può essere anche utile confrontare le proprie soluzioni con quelle di un risolutore ottimale di blocchi<sup>2</sup>, dato che per un esperto dovrebbe essere possibile trovare sempre (o quasi) un blocco  $2 \times 2 \times 2$  ottimale.

Lo step 3 insegna a riconoscere l'orientamento degli spigoli a prescindere dalla loro posizione nel cubo, un'altra abilità importante, dato che gli spigoli flippati sono una delle cose peggiori in cui ci si possa imbattere in una solve. Per questo step, così come per lo step 4, più che il Petrus è utile

---

1 Si chiama “blockbuilding” o “block building” (in inglese, “costruzione di blocchi”) la tecnica che consiste nel creare blocchi di pezzi per poi unirli. Viene solitamente contrapposto al metodo per completare l'F2L usato nel metodo CFOP (Fridrich), ma anche quest'ultimo può essere considerato un tipo particolare di blockbuilding. Un contrasto più appropriato è dato dal paragone con altre tecniche, come ad esempio l'orientamento spigoli (Petrus, ZZ), la risoluzione “Corners First”, l'utilizzo di algoritmi o i commutatori.

2 Ad esempio <http://laire.fi/jarcs/>, che purtroppo nel momento in cui sto scrivendo questa guida non è online.

imparare lo ZZ.

Non è necessario conoscere tutti gli algoritmi per l'ultimo strato (questo vale non solo per il Petrus, ma anche per gli altri metodi): nel capitolo successivo spiegherò più in dettaglio come procedere, per ora vi basti sapere che lo step “ultimo strato” non è solitamente compreso nelle soluzioni Fewest Moves.

[Sito di Lars Petrus](#)

## Roux

1. Costruire un blocco 3x2x1
2. Costruire un altro blocco 3x2x1, opposto al primo
3. CMLL (angoli dell'ultimo strato, senza la necessità di preservare lo strato M)
4. LSE (Last Six Edges, ultimi 6 spigoli, da risolvere solo con mosse degli strati M e U)

Il Petrus è un ottimo metodo per imparare il blockbuilding, ma imporsi di usare solo quello è comunque sbagliato: imparando anche il Roux (specialmente, anche in questo caso, i primi due step) la vostra abilità nel blockbuilding sarà più completa. Anche per questo metodo vi sarà molto utile studiare le soluzioni dei “Big” (al momento in cui viene scritta questa guida, il migliore in assoluto è Alexander Lau, secondo al mondo per la media 3x3 e campione europeo 2014).

Per lo step 3 vale quanto detto prima per gli algoritmi per l'ultimo strato, mentre lo step 4 è da evitare come la peste (per lo meno, evitate di farlo nel modo “standard”): ricordate che ogni mossa di uno stato centrale, come M, vale come 2 mosse normali!

[Guida al metodo Roux, di Waffle](#)

## ZZ

1. EOLine (orientamento di tutti gli spigoli e posizionamento di DF e DB; riduce il cubo ad essere risolvibile solo con mosse degli strati R, L e U)
2. F2L
3. Ultimo strato

Come già accennato prima, sapere riconoscere e risolvere l'orientamento degli spigoli è un'abilità molto utile e lo ZZ è sicuramente il modo migliore per impararla. A prima vista il concetto di “orientamento spigoli” può sembrare un po' complicato, se non altro perché molto astratto, ma non fatevi prendere dal panico, non c'è nulla di impossibile!

Il secondo step è all'incirca equivalente allo step 4 del Petrus, con la differenza che dovrete costruire contemporaneamente<sup>3</sup> i blocchi su entrambi i lati R ed L. Anche questo andrà ad arricchire la vostra esperienza nel blockbuilding.

[Guida di Conrad Rider](#)

---

<sup>3</sup> Per quanto riguarda la velocità di risoluzione, potrebbe essere più vantaggioso costruire prima un blocco e poi un altro, ma non per il FMC.

## CFOP (Fridrich) / FreeFOP

Nel CFOP classico si procede in questo modo:

1. Croce (4 spigoli sulla stessa faccia)
2. Inserimento di 4 coppie angolo/spigolo (F2L)
3. OLL (Orient Last Layer, Orientamento dell'Ultimo Strato)
4. PLL (Permute Last Layer, permutazione dell'ultimo strato)

Il CFOP classico **non** viene considerato un buon metodo per il fewest moves, ma in alcuni casi è utile conoscere diversi modi per inserire una coppia angolo/spigolo.

Nel FreeFOP i primi due step vengono sostituiti da una costruzione “libera” dei primi due strati, con il blockbuilding.

In ogni caso, sconsiglio fortemente di risolvere l'ultimo strato con OLL + PLL, a meno di non riuscire a skippare uno dei due step.

[Guida di Badmephisto](#)

## Keyhole F2L

Non è un vero e proprio metodo per risolvere il cubo, ma piuttosto una tecnica per completare i primi due strati. Viene considerato un metodo intermedio tra il metodo a strati e il CFOP. I passaggi sono i seguenti:

1. Croce
2. Posizionare 3 angoli del primo strato
3. Inserire 3 spigoli del secondo strato sfruttando l'angolo “libero”
4. Inserire l'ultima coppia angolo/spigolo, come nel CFOP oppure come nel metodo a strati

Per una maggiore efficienza sarebbe opportuno sostituire i primi due step con del blockbuilding e tentare di posizionare anche qualche spigolo del secondo strato nel frattempo. Una variante consiste nel costruire la croce e posizionare 3 spigoli del secondo strato, per poi inserire i 3 angoli del primo strato sfruttando lo spigolo “libero”.

Nonostante la sua semplicità, questo metodo può essere molto utile nel FMC.

## Heise

1. Costruire 4 “quadrati” 2x2x1 (con un colore in comune)
2. Unire i quadrati e orientare gli spigoli
3. Posizionare 2 angoli e i 5 spigoli rimanenti
4. Ultimi 3 angoli (commutatori)

Se decideste di non seguire il consiglio degli esperti e di utilizzare solo un metodo per il FMC, questo metodo deve assolutamente essere l'Heise. Da solo può portare a risolvere il cubo con una

media di meno di 40 mosse per soluzioni lineari<sup>4</sup>. È sicuramente un metodo estremo, ma anche estremamente efficiente.

I primi 2 step costituiscono un metodo molto particolare, ma utile nel FMC, per costruire un F2L-1<sup>5</sup> con gli spigoli orientati. Il concetto di non porsi dei limiti e cercare di sfruttare le varie situazioni viene qui applicato alla perfezione, data la grande libertà con cui consente di costruire i blocchi.

Il terzo step è molto complesso, ma è un'alternativa più efficiente al metodo classico di completare i primi due strati per poi finire l'ultimo strato con degli algoritmi. Praticandolo si acquisirà la capacità di costruire e muovere blocchi quando si è limitati dal grande numero di blocchi già presenti (e proprio in questo sta la difficoltà dello step).

L'ultimo passaggio richiede la conoscenza dei commutatori e permette, in una solve FMC, di sfruttare le insertions (entrambe queste tecniche verranno spiegate nel prossimo capitolo).

[Sito di Ryan Heise](#); oltre alla spiegazione dettagliata del metodo, potrete trovare molte altre informazioni utili per il fewest moves (per esempio, alcune [tecniche fondamentali](#)).

## Cosa e Come Imparare

Di certo non dovete puntare a saper risolvere velocemente il cubo con tutti questi metodi. Fare delle speedsolves può aiutare a vedere più velocemente alcune cose ed è se non altro divertente, ma **la velocità non ci interessa**. Ovviamente, dato che l'obiettivo è essere in grado di risolvere il cubo in meno mosse possibili, dovete puntare soprattutto all'efficienza. Essenziale è anche cercare di essere **color neutral**<sup>6</sup> e può essere utile cercare di lavorare con i “**Non Matching Blocks**”<sup>7</sup>.

Ma la differenza principale tra una speedsolve e una risoluzione FMC sta nel fatto che in quest'ultima potete **cercare varie strade**; se per esempio, nel Petrus, dopo aver costruito un blocco 2x2x3 vi ritrovate con 6 spigoli da orientare potete provare a costruire un blocco diverso, oppure a modificare leggermente la costruzione del vostro blocco per migliorare la situazione.<sup>8</sup>

Ecco qualche consiglio per alcuni dei metodi descritti:

### Petrus

- Dopo aver costruito un blocco 2x2x2 potete espanderlo in **3 diverse direzioni**.
- Provate a costruire direttamente un blocco 2x2x3, senza passare per il 2x2x2.
- Nello step 4 potete usare i Non Matching Blocks.
- Sempre nello step 4, provate ad **influenzare l'ultimo strato** per ottenere un caso più

---

4 Come nella variante FMC “linear”, ovvero senza provare strade alternative e senza cancellare mosse

5 Per F2L-1 intendo un F2L (First 2 Layers, primi due strati) completati ad eccezione di una coppia angolo/spigolo.

6 Saper risolvere il cubo partendo da qualunque “colore”; per esempio, partire da qualunque croce nel CFOP o da qualunque degli 8 possibili blocchi 2x2x2 nel Petrus.

7 Letteralmente, “Blocchi non Corrispondenti”, chiamati a volte, soprattutto nel FMC “Pseudo Blocchi”; è una tecnica utile soprattutto in Roux, ZZ e Heise, ma utilizzabile anche in altri metodi; consente di costruire blocchi che non sarebbero quelli previsti dal metodo, ma che possono essere posizionati negli stessi “slot”. Per esempio, nel Roux, il secondo blocco 3x2x1 può essere uno qualunque dei 4 costruibili sulla faccia opposta a quella del primo blocco. Questa tecnica è molto utile se usata assieme alle premoves, che verranno spiegate nel capitolo 3.

8 Cercare di influenzare i passaggi successivi mentre se ne risolve un altro è una buona abitudine, di cui si parlerà ancora.

favorevole (anche in “stile Heise”, se volete).

## Roux

- Tentate di costruire contemporaneamente i due blocchi.
- Provate con i Non Matching Blocks.
- Influenzate il CMLL durante il secondo blocco, e l'LSE durante il CMLL.

## ZZ

- Dopo aver orientato gli spigoli, non è sempre vantaggioso costruire la “Line”: provate direttamente con il blockbuilding (ma senza distruggere l'EO!).
- Una volta orientati gli spigoli, avrete in realtà ridotto il cubo ad essere risolvibile muovendo solo gli strati R, L, U e D: potete costruire l'F2L su una qualunque di queste facce.
- Anche qui come nel Petrus e nel Roux, provate ad usare i Non Matching Blocks e ad influenzare l'ultimo strato mentre costruite l'F2L.

## CFOP / FreeFOP

- Ovviamente, **il FreeFOP è migliore del CFOP**; provate almeno a fare una **Xcross**<sup>9</sup>.
- Cercate di influenzare l'orientamento degli spigoli dell'ultimo strato, almeno per non avere 4 spigoli flippati; possono essere utili alcuni algoritmi dello ZBF2L, ma più che impararli a memoria cercate di **studiarli per capire come funzionano**.
- Alcuni algoritmi ottimali per inserire le coppie non sono molto conosciuti (per esempio **F2 U R U' R' F2**): studiateli, anche in questo caso cercando di comprenderli il più possibile in maniera “intuitiva”.
- Provate a fare del “**multislott**ing”, ovvero inserire più coppie contemporaneamente. Il caso più semplice è quello in cui si utilizza una mossa di setup dello strato D, ad esempio D R U R' D'. Potete trovare alcuni algoritmi [qui](#), [qui](#) e [qui](#), ma vi consiglio anche di provare in maniera “libera”: guardate ad esempio come ho inserito seconda, terza e quarta coppia in [questa soluzione](#).

---

9 Croce più prima coppia, costruite contemporaneamente. Può anche essere vista come un  $2 \times 2 \times 2 + 2$  spigoli.

## 2. Procedere nella Risoluzione

Per citare Per Kristen Fredlund, il modo generale di procedere è:

“Pensala più o meno così: risolvi il cubo in 2 fasi, dove nella prima fase si risolve il maggior numero di pezzi possibile nel minor numero di mosse possibile (cioè costruire uno skeleton<sup>10</sup>), mentre la seconda fase si mettono a posto i pezzi rimanenti, tipicamente con delle insertions<sup>11</sup>”<sup>12</sup>.

Questo è un approccio generale, ma non deve essere sempre usato per forza di cose: a volte si possono ottenere ottime soluzioni, per esempio, costruendo un F2L tramite blockbuilding per poi risolvere l'ultimo strato con un algoritmo. Esistono anche modi di procedere molto diversi, due dei quali verranno spiegati nel capitolo 4.

Se questa descrizione vi sembra troppo generica, è perché non può essere altrimenti: non c'è un vero e proprio metodo standard che permette di ottenere sempre dei buoni risultati, ma bisogna **cercare ogni volta di provare più strategie possibili**, in modo da non lasciarsi sfuggire nessuna possibilità.

Di seguito descriverò le tecniche di base che saranno utili nelle solve FMC. Alcune saranno già state apprese dopo aver fatto pratica con i metodi descritti nel capitolo precedente, altre richiederanno uno studio ad hoc. Alcune verranno spiegate dettagliatamente, altre saranno per forza di cose soltanto accennate e verranno segnalate altre guide per uno studio più approfondito.

### Blockbuilding

Il blockbuilding è forse la tecnica più importante per procedere nel Fewest Moves. Concettualmente è molto semplice, ma richiede moltissima pratica per poter arrivare ad alti livelli. Fare pratica con metodi che si basano su questa tecnica (Petrus, Roux, Heise e ZZ), nelle modalità descritte nel capitolo precedente, è il modo più diretto per migliorare nel blockbuilding.

Di seguito elencherò alcune tecniche fondamentali che possono tornare utili nell'imparare a fare blockbuilding; le prime sono prese dal sito di Ryan Heise, molto ricco di esempi: guardateli!

### Allinea e Unisci

Tecnica assolutamente basilare: per costruire una coppia angolo/spigolo, che il tipo di blocco più semplice in assoluto, per prima cosa bisogna allineare i due pezzi per fare in modo che possano essere uniti con una mossa sola, e poi unirli con quella mossa.

Tale concetto si applica, in senso più generale, al caso in cui si vogliono formare blocchi più grandi, per esempio quando si vogliono unire una coppia e un quadrato 2x2x1 per formare un blocco 3x2x1.

Questa può sembrare una banalità, e in effetti un po' lo è; la cosa importante da imparare è **saper riconoscere quando due pezzi sono già allineati** e possono quindi essere uniti con una mossa sola.

---

10 Per “skeleton” (inglese per “scheletro”) si intende una soluzione parziale, dove non vengono risolti solo alcuni pezzi (in genere dai 2 ai 6).

11 Tecnica che consente di risolvere dei pezzi inserendo delle mosse in un punto precedente della solve. Verrà spiegata fra poco, un po' di pazienza!

12 <http://www.speedsolving.com/forum/showthread.php?1566-Fewest-Moves-Tips-and-Techniques&p=16209&viewfull=1#post16209>

In questo modo è possibile accorgersi in anticipo se una certa mossa unisce due pezzi.

Esempio:

Scramble<sup>13</sup>: F U' D2 L D L2 D F R2 B U2 R2 D L2 D L2 D R2 U' L2 F2

Due coppie sono già formate. Volendo formare anche la coppia spigolo blu-rosso e angolo blu-rosso-giallo si può procedere così:

Allinea: L2

Unisci: U2

## Spostalo

Può capitare che vogliamo costruire un blocco, ma la mossa più diretta per farlo distrugge altri blocchi già formati, oppure sposta dei pezzi che non vorremmo spostare. Uno dei modi per aggirare questo problema è spostare preventivamente il pezzo (o i pezzi) in questione, salvandolo così dalla mossa che lo sposterebbe, per poi, se necessario, rimetterlo al suo posto una volta che la mossa è stata eseguita.

Esempio:

Scramble: F R2 B D2 F D2 L2 B2 F' D2 F' L B U' F2 U2 L2 U B R2 F

Dopo U2 R2 D R2 abbiamo un quadrato 2x2x1 che può essere espanso ad un blocco 2x2x2 con F' D'; la mossa F' distrugge però la coppia formata dallo spigolo rosso-bianco e dall'angolo blu-rosso-bianco, che vorremmo salvare. Possiamo quindi eseguire U' (sposta) F' D' (risolvi) per salvare la coppia.

## Distruuggi e Ricostruisci

Un altro approccio per risolvere il problema descritto prima è quello di distruggere temporaneamente alcuni pezzi, per poi rimetterli assieme successivamente, applicando la tecnica “Spostalo”.

Esempio:

Scramble: R U R' U'

Ignoriamo per un momento l'ultimo strato e facciamo a finta di non sapere che, ovviamente, U R U' R' risolve tutto il cubo. Guardando solo i primi due strati, ci accorgiamo che R' posiziona la coppia pronta vicino agli altri pezzi in F, ma distrugge parte dell'F2L che avevamo già posizionato. Un modo per procedere utilizzando “Distruuggi e ricostruisci” è quindi:

R' (Distruuggi)

F (Spostalo)

R (Ricostruisci)

F' (Rimette a posto i pezzi spostati con F)

---

<sup>13</sup> Da qui in avanti, quando verrà proposto uno scramble d'esempio, nella spiegazione della soluzione (parziale o completa) potrà essere fatto riferimento ai colori dei pezzi. Quando accadrà, supporrò che si stia usando un cubo con schema di colori classico ([BOY](#)) e che lo scramble venga applicato con il bianco sopra (in U) e il verde in F

## Keyhole

Pur avendone parlato prima come un “metodo”, il keyhole può essere in generale considerato una strategia particolare per costruire blocchi. Tale tecnica consiste nello sfruttare le parti non risolte del cubo per risolvere alcuni pezzi. Se avrete fatto un'adeguata pratica con il keyhole non vi saranno necessari esempi per capire che cosa intendo, ma metto comunque una solve fatta utilizzando il Keyhole, per completezza:

Esempio<sup>14</sup>:

Scramble: U' R' L F' B U2 R2 B2 L' B R D F2 D2 L2 F2 D' R2 F2

Soluzione:

F' L' //Faccia -1 angolo

F2 L' B' L //Keyhole

F U' B U //Keyhole, anche se in realtà inserisce anche l'angolo

F' R' B2 R //Keyhole

F B L B L' //F2L

B' //LL

## Una Mossa, Due Obiettivi

Spesso è possibile utilizzare una sola mossa per costruire due blocchi oppure, più in generale, per “raggiungere due obiettivi”. Un esempio renderà tutto più chiaro.

Esempio:

Scramble: D U' F2 U' R' F R2 B D' B R F B' U R' D2 L' R2 F2 B' U' B D B2 F2 U L F U' B2

Tratto da [questa solve](#) di Mirek Goljan e Guus Razoux Schultz (trovata indipendentemente da entrambi, non in collaborazione). È possibile costruire un blocco 2x2x2 in 4 mosse: L U' F2 D'; la mossa in questione è quel F2: notate come con quella singola mossa venga formato il blocco 2x2x1 in DF e contemporaneamente si posizioni lo spigolo arancione-verde, che permetterà con la mossa successiva di completare il 2x2x2.

Situazioni come questa possono capitare “spontaneamente”, come in questo caso, ma è utile imparare a conoscerle nel caso in cui debbano essere forzate.

## Influenzare gli Step Successivi

Abbiamo già parlato (in maniera superficiale) di influenzare la fase del LL mentre viene completato l'F2L<sup>15</sup>. Questo concetto si applica anche al blockbuilding: molte volte conviene costruire un blocco in maniera sub-ottimale<sup>16</sup>, oppure aggiungere in alcuni punti mosse non strettamente necessarie al completamento dello step per avere una continuazione più favorevole, sia in termini di blockbuilding che di altro (ad esempio, di orientamento spigoli).

Esempio:

---

14 Questa solve è una versione modificata di [questa soluzione](#) di Edoardo Disarò adattata per non richiedere la conoscenza di tecniche che verranno spiegate solo successivamente.

15 Esempi degni di nota sono lo ZBLS (detto anche, meno propriamente, ZBF2L) e la Winter Variation, ma ne esistono molti altri: informatevi!

16 Per sub-ottimale o sub-optimal (con o senza il trattino) si intende una soluzione (parziale o totale) che impiega di più mosse della soluzione migliore possibile.

Scramble: L2 D2 U R2 F2 D2 B2 U' R2 B2 U B U F D B2 U L D' R' F

È possibile in questo caso costruire facilmente un quadrato 2x2x1 con L2 R B, ma aggiungendo una sola mossa i quadrati costruiti diventano due: L2 **B** R B.

## Occhio all'EO<sup>17</sup>!

Qualcuno potrà aver fatto caso, studiando i metodi descritti precedentemente, che un passaggio ricorrente è quello dell'orientamento degli spigoli. Come già detto, **più spigoli mal orientati ci sono, peggio andranno le cose**. Orientare gli spigoli alla fine è solitamente poco efficiente. Orientarli all'inizio, come nello ZZ, è comodo, ma si rischia di limitarsi nel blockbuilding. La cosa migliore da fare è, secondo molti (me incluso) tentare di risolvere, almeno in parte, l'orientamento degli spigoli **durante** la fase di blockbuilding. Una adeguata esperienza con metodi come ZZ e Petrus può portare a saper vedere a colpo d'occhio se uno spigolo sarà orientato correttamente o meno dopo alcune mosse; se ancora non si possiede un'esperienza sufficiente, ricordate che in una soluzione FMC potete tornare indietro e modificare la solve tutte le volte che volete: se avanzando nella risoluzione vi accorgete che l'orientamento degli spigoli vi sta causando problemi, provate a tornare indietro e a vedere se modificando qualche step le cose migliorano.

## Quali Blocchi Costruire?

La regola d'oro è **sfruttare le diverse situazioni**: un blocco 2x2x2, un 3x2x1, due quadrati 2x2x1 e molti altri tipi di blocchi essere un buon inizio. Cercate di provare tutte le possibilità che ci sono.

Due possibili approcci sono:

1. Porsi dei grandi obiettivi, come un blocco 2x2x3 oppure un F2L-1.
2. Procedere a piccoli passi, costruendo tanti blocchi piccoli e unendoli alla fine.

Erik Jernqvist propone i seguenti numeri di mosse, per capire cosa sia un buon inizio<sup>18</sup>:

Quadrato 2x2x1 + coppia angolo/spigolo: 3

Blocco 2x2x2: 4

Due quadrati 2x2x1: 5

Blocco 2x2x3: 9

F2L-1: 14

F2L: 17

Personalmente credo che queste siano delle buone stime, soprattutto per i primi 3 casi, ma c'è una cosa importante da ricordare: **la bontà di un inizio dipende dalle possibili continuazioni**. Se costruite un F2L-1 in 12 mosse, ma avete 4 spigoli male orientati, tra cui lo spigolo mancante dell'F2L posizionato ma flippato, potreste doverlo scartare. Al contrario, per esempio, un blocco 2x2x3 trovato sempre in 12 mosse con tutti gli spigoli orientati può essere un'alternativa migliore. Ovviamente l'orientamento degli spigoli non è l'unica cosa da guardare, ma è sicuramente una delle più importanti.

<sup>17</sup> Dall'inglese "Edge Orientation", orientamento degli spigoli.

<sup>18</sup> Presi da [questo post](#) su speedsolving.com. Ovviamente, cosa sia un "buon inizio" dipende anche dal vostro livello attuale; i dati proposti sono da considerarsi un buon obiettivo per chi vuole diventare un esperto, ma se non siete ancora a livelli alti potete accontentarvi di risultati più modesti: non sprecate tutto il tempo che avete per cercare un buon inizio!

Altra regola: **mai**<sup>19</sup> **completare un F2L senza influenzare l'ultimo strato**. Il motivo è semplice: un ultimo strato particolarmente brutto può richiedere molte mosse, per quanti algoritmi si possano conoscere, e un F2L completo garantisce una scarsa libertà nella manipolazione del cubo; al contrario, **un F2L-1 è un buon obiettivo intermedio**, perché lascia una buona libertà d'azione, nonostante il grande numero di pezzi già posizionati.

## Blocchi già Fatti: Come Trattarli?

Può capitare che uno scramble presenti dei blocchi già formati, oppure che con le prime mosse di una soluzione abbiate involontariamente formato dei piccoli blocchi (per lo più coppie angolo/spigolo). In questi casi è preferibile tentare di sfruttare i blocchi già formati piuttosto che lasciarli perdere e andare avanti per la propria strada. Ma come? I modi sono tre:

1. Espandere / unire i blocchi già presenti; questa è la cosa più ovvia da fare.
2. Espandere / unire i blocchi già presenti, ma **facendo attenzione a come si spostano gli altri pezzi**, cercando contemporaneamente di formare altri blocchi; questa è generalmente la cosa migliore da fare.
3. Non espandere subito i blocchi già presenti, ma tentare di formarne di nuovi, preservando, possibilmente, gli altri.

Inoltre, è molto importante, ma terribilmente difficile, **riuscire a capire quando non conviene preservare un blocco**, ma piuttosto distruggerlo per formarne altri. Personalmente, trovo molto difficile “arrendermi” e “buttare via” un blocco di pezzi, ma mi accorgo che spesso questo mi causa problemi, specialmente nella gestione del tempo.

## Trucchi per Velocizzarsi e Tecniche Avanzate

“Velocizzarsi” non va inteso nel senso di una speedsolve, ma come “trovare più velocemente un buon inizio<sup>20</sup> di solve con il blockbuilding”. Essere veloci nel trovare un buon inizio è molto importante, perché aiuta a risparmiare tempo prezioso (un'ora passa in fretta!) e sarebbe bene anche riuscire ad esplorare, o per lo meno a notare, tutti gli inizi più promettenti.

L'unità di blocco più semplice è la coppia angolo/spigolo, o blocco 2x1x1. È molto probabile che ce ne sia almeno una di pronta, senza il bisogno di fare alcuna mossa. Se così è, potete trattarla in uno dei modi descritti nella sezione “Blocchi già Fatti: Come Trattarli?”. Altrimenti, sarebbe una buona cosa riuscire a riconoscere a colpo d'occhio le coppie che possono essere formate con una mossa sola (vedi “Allinea e Unisci”). Se non siete ancora veloci nel riconoscerle, o comunque se volete risparmiare un po' di fatica<sup>21</sup>, potete usare la tecnica “Brute Force”: provarle tutte. Tutte le possibili mosse iniziali, partendo da U, U2, U', poi R, R2, R', eccetera, controllando dopo ogni mossa se siete riusciti a formare una coppia.

Una tecnica più avanzata, molto più utile ma che richiede anche un maggiore dispendio di energia, consiste nel controllare ogni possibile blocco 2x2x2 (sono 8 in tutto). Per farlo, procedete in questo

---

<sup>19</sup> Ovviamente esistono [eccezioni](#).

<sup>20</sup> Come inizio si intende un blocco 2x2x2, un 3x2x1, alcuni blocchi più piccoli o, in alcuni casi, un 2x2x3; in generale, l'inizio può essere considerato qualunque cosa dalle prime 2 alle prime 7 mosse.

<sup>21</sup> Dato che dovrete continuare a pensare per un'altra ora, risparmiare fatica non è una cosa da pigri, ma un accorgimento utile.

modo: **per ogni angolo**<sup>22</sup>, **cercate i tre spigoli corrispondenti** e provate a vedere quale può essere il modo migliore per costruire quel blocco 2x2x2; sforzatevi di non eseguire nessuna mossa “di prova”, così da poter continuare con gli angoli successivi senza dover ri-scramblare il cubo. Spesso, se vedo che un particolare blocco 2x2x2 richiederebbe troppe mosse, semplicemente lo ignoro e passo oltre. Queste tecnica, oltre a permettere di trovare, in molti casi, un blocco 2x2x2 ottimale (che è spesso un buon punto di partenza), consente di avere un'idea di dove si trova ogni pezzo.

Un altro motivo per cui vi consiglio di provare a “vedere” le mosse per formare un blocco senza davvero eseguirle è che per velocizzarsi e migliorare nel blockbuilding è molto utile saper “calcolare”<sup>23</sup> gli spostamenti dei pezzi senza per forza vederli. Alexander Lau, di cui si è già parlato in precedenza, nei 15 secondi di ispezione concessi prima di una speedsolve riesce a pianificare l'intero blocco 3x2x1 (primo step del Roux). La sua pianificazione è talmente accurata che, mentre lo risolve, è in grado di fare look-ahead<sup>24</sup> e pianificare (almeno in parte) il secondo blocco.

Ci si può allenare nella pianificazione anche con questo semplice gioco: chiedete ad un vostro amico di mischiare il cubo con 3 mosse<sup>25</sup> e poi cercate di trovare quelle 3 mosse (dovrebbe essere facile). Poi, se ci siete riusciti, passate a 4 mosse, e avanti così. A seconda della vostra abilità, potreste incontrare delle difficoltà arrivati alle 6, 7 o 8 mosse. Se riuscite ad arrivare senza problemi a 9 o 10 mosse, complimenti!

## Trovare uno Skeleton

Una volta arrivati ad un buon punto (tipicamente, un F2L-1) con il blockbuilding, è difficile proseguire con le tecniche standard appena descritte. Quello che dobbiamo fare ora è trovare un cosiddetto “skeleton”, cioè una soluzione parziale dove mancano da risolvere solo alcuni pezzi. La situazione migliore è quella in cui i 3 pezzi non risolti sono 3 angoli che formano un 3-cycle<sup>26</sup>, ma anche 4 o 5 angoli o 3 spigoli possono andare bene, a seconda di quante mosse sono necessarie per costruire lo skeleton.

Come fare quindi, quando le tecniche viste prima diventano difficili da usare, senza distruggere quanto già fatto? L'Heise è un ottimo punto di partenza. Sia il metodo che il sito: lo [step 3](#) è descritto accuratamente, specialmente per quanto riguarda il “[Two Pairs Approach](#)”.

Lo step 3 dell'Heise prevede che, oltre ad avere un F2L-1, gli spigoli siano già orientati. Se così non fosse, potete decidere di

1. Orientarli a questo punto; generalmente sconsigliato, a meno che non richieda poche mosse.
2. Modificare gli ultimi passaggi fatti fino ad ora, per avere alla fine gli spigoli orientati.
3. Orientare gli spigoli mentre si finisce di costruire lo skeleton.

In sostanza, gli approcci possibili per ottenere uno skeleton sono molti e il modo migliore per

---

<sup>22</sup> Io parto da ULB e procedo in senso antiorario, faccio x2 e ripeto.

<sup>23</sup> Parola che va intesa nel senso degli scacchi: si dice che un giocatore “calcola” 6, 7 o 8 mosse quando è in grado di pensare alle possibili risposte e contro-risposte ad una mossa per un totale di 6, 7 o 8 mosse successive.

<sup>24</sup> Nelle speedsolve, per look-ahead (“Guardare avanti”, con o senza trattino) si intende la capacità di pensare agli step successivi mentre se ne risolve un altro.

<sup>25</sup> Decidete voi se in HTM, QTM o STM, ma mettetevi d'accordo su quale metrica usare!

<sup>26</sup> Un 3-cycle è un ciclo di 3 pezzi del cubo; ad esempio la A-perm è un 3-cycle, così come la U-perm e l'OLL (L F R F' L' F R' F').

allenarsi in questo caso, ma sempre e comunque una buona abitudine, è cercare online (ad esempio, <http://fmcsolves.cubing.net/>) solves di gente esperta; nella maggior parte dei casi, presenteranno uno skeleton.

Può essere utile anche un allenamento mirato: se andate su <http://www.qqtimer.net/> potete selezionare il tipo di scramble 3x3x3 subset → last slot + last layer.

Ovviamente non dovete per forza arrivare ad un F2L-1 prima di completare lo skeleton, ma questa è spesso la soluzione più facile<sup>27</sup>. Cercate comunque di preservare eventuali piccoli blocchi (coppie o quadrati 2x2x1) formati da pezzi del LL se se ne dovessero formare.

## Commutatori

Secondo la [definizione data su speedsolving](#) di “Commutatore”, per quel che riguarda il cubo<sup>28</sup> è una sequenza di mosse del tipo:

$A B A' B'$

dove A e B sono sequenze di mosse e X' indica l'inverso della sequenza X<sup>29</sup>. Tale commutatore si scrive, in notazione compatta, come:

[A, B]

Nella pratica, spesso, per “commutatore” si intende in realtà un “commutatore che risolve un 3-cycle”. Anche noi useremo la parola “commutatore” con questo significato.

In contrasto con il blockbuilding, che risolve molti pezzi modificando però pesantemente altre parti del cubo, i commutatori risolvono un piccolo numero di pezzi (solitamente 3) lasciando tutti gli altri pezzi esattamente dov'erano. Questi, insieme al blockbuilding e alle insertions (che vedremo nel prossimo paragrafo) pongono le basi per una buona solve FMC.

## Commutatori di Angoli

I commutatori per gli angoli sono probabilmente **i più utili** nel FMC. La “Niklas” (R U' L' U R' U' L U) la A perm (R2 B2 R F R' B2 R F' R) e altri algoritmi che probabilmente già conoscete possono essere visti come commutatori.

Per imparare i commutatori (che non spiegherò in dettaglio in questa guida), vi consiglio la [guida di Brian Yu](#) che è veramente ben fatta, sia in versione scritta che sotto forma di video. Se preferite, c'è anche la [traduzione in italiano da Matteo Colombo](#).

Per il FMC basta conoscere i “pure”, commutatori semplici da 8 mosse. Date pure un'occhiata agli A9, e se volete anche a tutti gli altri casi ma, come vedremo quando parlerò delle insertions, non saranno praticamente mai<sup>30</sup> necessari nel FMC classico.

---

27 Nel caso in cui non seguiate questa strada, le possibilità sono molte e richiedono soluzioni ad hoc.

28 Esiste anche una definizione matematica, nella teoria dei gruppi, da cui deriva la definizione data.

29 Ad esempio, l'inverso di (U R) è (R' U'), non (U' R') e non (R U)!

30 La parola “mai” va a braccetto con una nota a piè di pagina, ormai dovrete averlo capito. Anche in questo ci sono delle [eccezioni](#).

## Commutatori di Spigoli

Una volta imparati i commutatori per gli angoli, non dovrebbe esservi difficile intuire come funzionino i commutatori per gli spigoli, soprattutto quelli del tipo:

$$[U R U', M'] = U R U' M' U R' U' M$$

Purtroppo però, commutatori di questo tipo non sono sempre convenienti (anche se a volte utili) perché, come avrete notato, impiegano mosse dello strato M, che valgono doppio.

Il commutatore spigoli che **tutti dovrebbero sapere** è:

$$[M', U2] = M' U2 M U2$$

Che risulta molto utile anche se abbinato a mosse di setup, come:

$$[U: [M', U2]]^{31} = U M' U2 M U2 U' = U M' U2 M U$$

Ricordatevi che nelle gare ufficiali non potete scrivere mosse dello strato centrale, quindi il commutatore di base diventerebbe:

$$M' U2 M U2 = R' L x U2 R L' x' U2 = R' L F2 R L' U2$$

Notate che le prime due mosse (R' L) possono essere invertite. Questo vale per ogni coppia di mosse parallele (opposte).

Un'altra cosa da notare è che le prime due mosse non spostano nessuno dei 3 spigoli che ci interessano: R' L F2 R L' U2 diventa quindi equivalente sia a L F2 R L' U2 R' che a F2 R L' U2 R' L e, dato che possiamo invertire R' ed L, anche a R' F2 R L' U2 L.

Queste considerazioni sono utili soprattutto quando si vogliono **cancellare** delle mosse, ovvero fare in modo che le prime mosse di un commutatore (o, in generale, di qualunque sequenza di mosse vogliamo applicare) siano corrispondenti all'inverso delle mosse immediatamente precedente (oppure, che le ultime mosse siano corrispondenti all'inverso delle prime mosse successive alla nostra sequenza)<sup>32</sup>.

Esistono anche 3-cycles di spigoli che non sfruttano mosse di strati centrali. I due più conosciuti sono:

$$R2 B2 L2 U B2 R2 F2 D$$

$$R2 B2 L2 U L2 B2 R2 D$$

Notate che nel primo algoritmo le prime due mosse non influenzano gli spigoli da permutare: è possibile quindi “shiftarlo”, come abbiamo fatto prima con R' L F2 R L' U2.

Un ultimo appunto per gli spigoli: anche se non sono dei 3-cycles, è bene conoscere le sequenze di mosse M2 U2 M2 U2 e R2 U2 R2 U2 R2 U2, che scambiano due coppie di spigoli.

---

31 La notazione [A: B] indica un *coniugato*, cioè A B A'; la sequenza di mosse A viene comunemente chiamata “mosse di setup”.

32 Ad esempio, se abbiamo costruito un F2L che termina con ...U R2 F', e vogliamo usare l'algoritmo F R U R' U' F', si cancellano 3 mosse e la nostra sequenza diventerà:  
...U R2 F F' R U R' U' F' = U R' U R' U' F'

## Commutatori di Blocchi

Se avete letto attentamente il sito di Ryan Heise, avrete già sentito parlare dei cosiddetti “pairs 3-cycles”, o commutatori di blocchi. Se conoscete i commutatori di angoli, non vi sarà difficile ricavarli intuitivamente. Guardate ad esempio:

$$[L Dw' L', U'] = L Dw' L' U' L Dw L' U$$

Sono molto utili nel terzo step del metodo Heise, ma anche per risolvere contemporaneamente un 3-cycle di angoli e un 3-cycle di spigoli. Ad esempio, l'algoritmo per l'ultimo strato  $M F U F' U' F' L F R'$  può essere visto come:

$$[R: [L' Dw L, U']] = R L' Dw L U' L' Dw' L U R'$$

cioè una mossa di setup + un commutatore di coppie.

Anche la J-perm può essere vista come un 3-cycle di coppie:

$$\text{Scramble: } R U^2 R' U' R U^2 L' U R' U' L U$$

$$\text{Soluzione: } [R^2: [Fw^2, D B^2 D']] = R^2 Fw^2 D B^2 D' Fw^2 D B^2 D' R^2$$

## Insertions

Dopo averle citate più volte nel corso della guida, è (finalmente) il turno delle insertions.

Siamo riusciti, in qualche modo, a trovare un buono skeleton; supponiamo che ci manchi un 3-cycle di angoli per risolvere il cubo. Che cosa facciamo? Ovviamente possiamo risolvere direttamente il 3-cycle usando uno dei commutatori; ma, se avete studiato tutti i casi, saprete che un 3-cycle di angoli può richiedere fino a 12 mosse. Sapendo che il caso migliore ne richiede solo 8, quelle 4 mosse in più sono una bella seccatura. Ma c'è il modo di risolvere un 3-cycle di angoli utilizzando praticamente sempre *meno* di 8 mosse: questo grazie alle insertions.

### Insertions Semplici

Il concetto su cui si basa questa tecnica non è troppo complesso: se mi mancano 3 angoli da risolvere, posso ripercorrere tutto il mio skeleton mossa per mossa e risolverli in un momento in cui il commutatore corrispondente richiede 8 mosse; dato che quel commutatore non influenza altro che i tre pezzi che devono essere influenzati, proseguendo nella solve tutti gli altri pezzi finiranno per essere risolti, tanto quanto prima, e in aggiunta andranno al loro posto anche i 3 angoli che ho permutato con il commutatore inserito.

E con questo si può arrivare praticamente sempre a risolvere un 3-cycle di angoli in 8 mosse. Come fare per trovare una soluzione che richieda meno di 8 mosse? Semplicemente, tra tutti i possibili commutatori inseriti che risolvono il nostro 3-cycle, si sceglierà quello che **cancella più mosse**; generalmente, basta controllare tutte le situazioni in cui i 3 angoli sono risolvibili con un commutatore “pure”, per poi scegliere quello che crea più cancellazioni; in [rarissimi casi](#) l'insertion migliore potrebbe essere in realtà data da un commutatore da 9 o più mosse, ma sono situazioni talmente improbabili che non vale la pena controllare tutti i tipi di commutatore.

Per riuscire più facilmente a seguire i movimenti dei 3 angoli in giro per il cubo mentre ripercorrete

lo skeleton, molti consigliano di attaccare al cubo degli adesivi bianchi<sup>33</sup> su cui scrivere i numeri 1, 2 e 3 (o le lettere A, B e C se preferite)<sup>34</sup>. Personalmente, io consiglio di prendere un cubo economico con adesivi non lucidi e scriverci sopra i numeri con una matita.

Una [soluzione d'esempio](#) renderà tutto più chiaro:

Scramble: D B2 U' F2 L2 D2 R2 U F2 U2 L2 R' D2 B L' U' R2 F2 R B F2

Skeleton:

B' U' D L' F' //EO + blocchi

D2 L2 D' L //Pseudo<sup>35</sup> 2x2x3

U2 R2 U' R' //Pseudo 2x2x1

U L' U R' U' L U2 R' L' //Tutto tranne 3 angoli

Arrivati a questo punto, scrivete con una matita “1” sull'adesivo rosso dell'angolo blu-rosso-giallo, “2” sull'adesivo arancione dell'angolo blu-giallo-arancione e “3” sull'adesivo bianco dell'angolo arancione-blu-bianco.<sup>36</sup> Risolvete il cubo (L B2 L F L' B2 L F' L2) e ri-scramblatelo.

Potremmo risolvere i tre angoli in questo momento, ma servirebbero 9 mosse (R2 F R B2 R' F' R B2 R). Eseguiamo quindi la prima mossa dello skeleton (B') e vediamo se ora il caso è più favorevole: no, sempre 9 mosse. Eseguiamo la mossa successiva (U')<sup>37</sup>: ora possiamo risolvere i tre angoli con un commutatore da 8 mosse (L2 F R F' L2 F R' F')! Se volessimo usarlo, la soluzione finale da scrivere e consegnare sarebbe:

B' U' L2 F R F' L2 F R' F' D L' F' D2 L2 D' L U2 R2 U' R' U L' U R' U' L U2 R' L'

Provate ad eseguirla su un altro cubo e convincetevi che funziona.

Ad ogni modo, possiamo fare di meglio. Eseguite la mossa successiva (D) e notate che ora il caso richiede 9 mosse<sup>38</sup>. Procediamo nello stesso modo per qualche mossa ancora, fino a che non arriviamo a ...L' F' D2. Siamo di nuovo in una situazione in cui i tre angoli possono essere risolti in 8 mosse (D' F2 D B2 D' F2 D B2)<sup>39</sup>. Ma c'è di più: l'ultima mossa eseguita e la prima del commutatore si cancellano! Se volessimo risolvere i tre angoli in questo punto, dovremmo scrivere:

B' U' D L' F' D2 D' F2 D B2 D' F2 D B2 L2 D' L U2 R2 U' R' U L' U R' U' L U2 R' L'

Che è equivalente a:

B' U' D L' F' D F2 D B2 D' F2 D B2 L2 D' L U2 R2 U' R' U L' U R' U' L U2 R' L'

Una mossa in meno, urrà! Abbiamo quindi risolto, a tutti gli effetti, 3 angoli in 7 mosse.

Per completezza, dovremmo ora proseguire fino alla fine dello skeleton cercando insertions

<sup>33</sup> Ecco perché potete portare con voi, ad una gara ufficiale, “illimitati adesivi colorati”.

<sup>34</sup> Gli adesivi vanno attaccati in modo che il 3-cycle che sposta 1 in 2, 2 in 3 e 3 in 1 sia quello che risolve il cubo.

<sup>35</sup> Degli pseudo blocchi parleremo fra poco.

<sup>36</sup> Esistono molte numerazioni equivalenti: potete partire dall'angolo che volete e dall'adesivo che preferite, l'importante è che la numerazione sia *coerente*.

<sup>37</sup> Achtung: quando in uno skeleton avete consecutivamente 2 mosse di strati paralleli, provate a invertirne l'ordine per vedere se in questo caso si trovano insertions migliori. In questo caso non è così, ma non si sa mai.

<sup>38</sup> Le due mosse U' D corrispondono ad una mossa dello strato centrale (E'), quindi non influenzano gli angoli: era ragionevole pensare che, prima e dopo queste due mosse, il 3-cycle di angoli da risolvere fosse lo stesso, a meno di rotazioni (y'/y in questo caso).

<sup>39</sup> Anche B2 U' F' U B2 U' F U

migliori. In effetti, in questo caso l'insertion migliore si trova verso la fine:

B' U' D L' F' D2 L2 D' L U2 R2 U' R' U L' \* U R' U' L U2 R' L'

\* = L F' L' B L F L' B'

Soluzione finale:

B' U' D L' F' D2 L2 D' L U2 R2 U' R' U L' L F' L' B L F L' B' U R' U' L U2 R' L'

Dove ovviamente L ed L' si elidono, ottenendo:

B' U' D L' F' D2 L2 D' L U2 R2 U' R' U F' L' B L F L' B' U R' U' L U2 R' L'

Senza tante spiegazioni dovrete aver capito come trovare insertions per 3-cycle di spigoli. Si procede in maniera analoga anche per trovare insertions di doppi 2-cycles di spigoli, risolvibili con M2 U2 M2 U2 o con R2 U2 R2 U2 R2 U2 o altre varianti<sup>40</sup>.

### Insertions Multiple: Cicli Separati (3 angoli e 3 spigoli)

Non è necessario che uno skeleton lasci da risolvere un solo 3-cycle: le insertions si applicano benissimo anche alla soluzione di più di un ciclo (o di un ciclo più lungo) di pezzi.

Come abbiamo già accennato, due 3-cycles, uno di angoli e uno di spigoli, possono essere risolti con un **commutatore di coppie** (eventualmente preceduto da qualche mossa di setup). Un'altra possibilità da considerare è quella di risolvere i 3 spigoli con una **“Sune”**<sup>41</sup> o varianti, in modo da influenzare solo i 3 angoli ancora non risolti<sup>42</sup>, oltre ovviamente agli spigoli da ciclare. Entrambi questi metodi vanno sempre tenuti a mente, ma sono raramente applicabili<sup>43</sup>. La soluzione “standard” è quella di usare **due commutatori inseriti**.

Dopo aver numerato sia i tre angoli che i tre spigoli<sup>44</sup>, procedo mossa per mossa come nel caso delle insertions semplici, cercando però ad ogni passo sia una soluzione per gli angoli che una per gli spigoli, oltre ad eventuali soluzioni con commutatori di coppie o Sune. Una volta che avrete finito, potete direttamente scrivere la soluzione completa con le due insertions, ma potrete anche fare **un'ulteriore ricerca**: scegliete, ad esempio, di tenere il commutatore di angoli, ma di voler cercare un'insertion migliore per quello di spigoli; se scrivete la soluzione parziale per lo skeleton con il commutatore angoli già inserito (e le eventuali cancellazioni già fatte), quello che ottenete è un **nuovo skeleton**, di qualche mossa più lungo, dove i pezzi rimasti da risolvere sono 3 spigoli. A questo punto potete procedere come nel caso di un'insertion semplice. Perché tanta fatica? Perché è possibile che un buon punto per inserire il commutatore di spigoli sia **all'interno dell'altro commutatore**. Potete anche cercare di inserire prima il commutatore per gli spigoli e poi quello per gli angoli. Un esempio, anche se più complicato di quanto sia in realtà necessario per capire il concetto, è dato da [questa solve](#).

Questa tecnica si applica benissimo in tutti i casi in cui siano necessarie due o più insertions, risolvibili con cicli (3-cycles o doppi 2-cycles) che non si intersecano. In altri casi, le cose si

40 È possibile trattare quest'ultimo caso con una doppia insertion, come spiegherò in seguito per il caso di doppi 2-cycles di angoli.

41 R U R' U R U2 R'

42 Che poi dovranno comunque essere risolti in qualche modo, preferibilmente con un'altra insertion.

43 A meno di non usare un sacco di mosse di setup, il che però li rende svantaggiosi.

44 Solitamente, preferisco usare numeri per gli angoli e lettere per gli spigoli per non rischiare di confondermi.

complicano un po'.

## Insertions Multiple: 2 o 3 Angoli Flippati

Quando vi ritrovate con due angoli da flippare potete cercare un punto nella solve dove inserire l'algoritmo  $[F L' D2 L F', U2]$  (12 mosse); similmente, per 3 angoli potete usare  $U' B U' F U2 B2 D' R2 U D F' U' B$  (13 mosse). Tuttavia questo non sempre il modo migliore, specialmente nel secondo caso.

Il modo più classico di risolvere 2 o 3 angoli flippati è usare **due insertions** di commutatori di angoli. Il primo commutatore non dovrà far altro che ciclare, in un orientamento qualsiasi, i 3 angoli da flippare (o i due angoli da flippare + un terzo angolo libero) e tipicamente porterà a molte cancellazioni. Successivamente, non dovrete far altro che inserire un 3-cycle di angoli nel nuovo skeleton ottenuto, come nel caso delle insertions semplici.

Per fare questo, vi basterà segnare con una X o un altro simbolo gli angoli flippati; nel caso vogliate provare ad usare uno degli algoritmi di “pure flip” descritti prima, vi consiglio di usare come simbolo un freccia, in modo da sapere anche in quale verso (orario o antiorario) l'angolo deve essere ruotato.

Dopo aver trovato il primo ciclo, cancellate i simboli che avete disegnato e ri-numerate gli angoli a dovere, con 1, 2 e 3.

Un procedimento analogo è applicabile al caso di 2 spigoli flippati, ma consiglio di evitare situazioni del genere, perché i commutatori di spigoli richiedono, generalmente, più mosse di quelli di angoli.

## Insertions Multiple: 4 Angoli

Se dovete risolvere 4 angoli, l'unico caso particolarmente ostico (che richiede 3 insertions di commutatori di angoli) è quando i 4 angoli sono posizionati correttamente, ma flippati. Preferibilmente da evitare.<sup>45</sup>

I casi rimasti sono 3:

1. Un angolo è nella posizione corretta ma flippato, gli altri tre formano un “3-cycle flippato”<sup>46</sup>
2. Due coppie di angoli da scambiare di posto, orientate correttamente (doppio swap o doppio 2-cycle)<sup>47</sup>.
3. Un “doppio 2-cycle flippato”.

<sup>45</sup> Una buona lettura in merito è la discussione generata da [questo post](#) su speedsolving.com, FMC thread. Il punto di vista (più che autorevole) di Sébastien Auroux è leggermente diverso. Resta comunque il fatto che 4 angoli flippati sono il peggior caso possibile per quanto riguarda 4 angoli, e che, oltre al possibile numero di mosse, bisogna tener da conto che cercare 3 insertions richiede più tempo che cercarne 2.

<sup>46</sup> Un 3-cycle per quanto riguarda la permutazione, senza considerare l'orientamento. Un ciclo flippato dipende sempre da altri pezzi (o altri cicli) “flippati”

<sup>47</sup> Questo caso si può risolvere, oltre che nel modo che verrà spiegato fra poco, anche trasformando il doppio 2-cycle di angoli in un doppio 2-cycle di spigoli: ricordate che una H perm ( $M2 U M2 U2 M2 U M2$ ) può essere trasformata in una permutazione di angoli (anche detta X-perm) con una sola mossa ( $U2$ ).

Un altro modo ancora consiste nell'inserire algoritmi come  $(R U R' U')^3$  (triple sexy) o  $(R' F R F')^3$  (triple sledge), che scambiano due coppie di angoli.

Questi due metodi alternativi possono essere occasionalmente utili, ma è quasi sempre migliore il metodo generale, che verrà spiegato fra poco.

Tutti questi casi **si possono risolvere con due commutatori**: il primo deve risolvere uno solo dei 4 angoli influenzando in maniera libera<sup>48</sup> gli altri 3, mentre il secondo commutatore, che andrà inserito nel nuovo skeleton ottenuto dopo la prima insertion, dovrà risolvere i 3 angoli rimasti.

Per fare questo, io segno gli angoli nei modi seguenti, a seconda del caso:

1. Segno l'angolo flippato con una X (tanto non mi interessare in che verso deve essere ruotato), poi numero gli altri 3 angoli da 1 a 3; a questo punto però, dato che il ciclo di angoli è "flippato", avrò che 1 dovrà andare in 2 e 2 in 3, mentre 3 non dovrà andare in 1, ma su un'altra faccia dello stesso angolo<sup>49</sup>. Nessuno problema: segnate quello sticker con un 4, e il gioco è fatto.
2. Segno la prima coppia di angoli con due X e la seconda coppia di angoli con 2 A.
3. Se nel caso di un 3-cycle flippato abbiamo bisogno di 4 numeri, per un 2-cycle flippato ne serviranno 3: ragionando come nel primo caso, segno uno dei 2-cycles con i numeri 1, 2 e 3 e l'altro con le lettere A, B e C.

Nel primo caso, per esempio, un possibile "primo ciclo" è quello che manda X in 3, 3 in 4 e 4 in X. Non va bene invece il ciclo che manda 1 in 2, 2 in 3 e 3 in 1, perché vi lascia 2 angoli flippati, anziché un 3-cycle.

Dopo aver trovato il primo ciclo, come prima, cancellate i simboli che avete disegnato e ri-numerate gli angoli a dovere con 1, 2 e 3.

Anche in questo caso, sconsiglio di aver a che fare con l'analoga situazione per i 4 spigoli.

Un ottimo esempio è [questo adattamento](#)<sup>50</sup> dell'attuale record sudamericano (25 mosse) di João Pedro Batista Ribeiro Costa:

Scramble: L U2 D' L' U2 B' D B' L U2 F2 R2 F' R2 L2 F' U2 D2 F

Soluzione:

F D' # U2 \* R //EO

D' F' L2 //2x2x2

F2 D F2 //Pseudo F2L-2

B' D B //F2L-1

F' D' F' //AB4C

\* = U R D' R' U' R D R' //Primo commutatore, 5 mosse si cancellano

# = L' U' R' U L U' R U //Secondo commutatore, 3 mosse si cancellano

Soluzione finale: F D' L' U' R' U L U' R2 D' R' U' R F' L2 F2 D F2 B' D B F' D' F'

---

48 Tranne nel primo caso, in cui dovete stare attenti a comprendere nel vostro commutatore l'angolo flippato ma posizionato.

49 Ricordatevi sempre che, quando si considerano in commutatori, è molto importante distinguere tra "faccia di un pezzo" e pezzo.

50 La solve è adattata (e risulta una mossa più corta) perché la versione originale prevede l'utilizzo di una premove, tecnica che non è ancora stata spiegata.

## Insertions Multiple: 5 Angoli

Tra tutti i casi in cui rimangono 5 angoli da risolvere, l'unico che richiede 2 commutatori per essere risolto è quello in cui i pezzi formano un 5-cycle. Tutti gli altri casi richiedono 3 commutatori, tranne quando i 5 angoli sono tutti posizionati correttamente ma flippati: in quel caso ne servono 4. Noi ignoreremo tutti casi che richiedono 3 o più commutatori (anche se, in casi estremi, potrebbero non essere da scartare) e vedremo solo il primo.

Il modo più semplice di procedere prevede **due passaggi**, come nei casi visti primi (4 angoli) ed è forse quello più utilizzato. Dopo aver numerato gli angoli da 1 a 5, si cerca, sempre ripercorrendo la soluzione mosse per mosse, un commutatore che risolva un ciclo di 3 angoli consecutivi. Questi cicli sono:

1 → 2 → 3 → 1

2 → 3 → 4 → 2

3 → 4 → 5 → 3

4 → 5 → 1 → 4

5 → 1 → 2 → 5

Ovviamente questo richiederà più tempo che non cercare un commutatore per risolvere un solo 3-cycle di angoli. Anche in questo caso sarà sufficiente cercare solo i commutatori “pure”. Ogni volta che trovate un buon commutatore dovrete scriverlo da qualche parte.

Una volta trovati tutti i possibili commutatori, scegliete il migliore (quello cioè che cancella più mosse)<sup>51</sup>. Inserite quel commutatore; ora avrete un nuovo skeleton e vi resterà da risolvere un solo 3-cycle: sapete già come fare.

Questo metodo **non assicura** di trovare la soluzione ottimale: per essere avere un grado di certezza maggiore, dovrete provare tutti (o almeno buona parte di) i commutatori trovati con il primo passaggio e fare quindi molti passaggi aggiuntivi (probabilmente inutili). Questo porta, solitamente, ad un grande spreco di tempo e raramente ad un reale miglioramento della soluzione.

Per sapere se è probabile o meno che esista una coppia di insertions migliore di quella trovata, tenete presente che 10 o 11 mosse totali sono un obiettivo accettabile, per un 5-cycle di angoli.

Esiste anche un metodo più veloce, leggermente più complesso, che richiede **un solo passaggio**. Il procedimento è quasi identico al primo dei due passaggi descritti prima, solo che, oltre a segnare i commutatori trovati, dovrete anche prendere nota di quale ciclo di spigoli risolvono. A questo punto, senza toccare il cubo, potete già scegliere una coppia di commutatori per risolvere i 5 angoli.

Per capire come, serve un po' di “**teoria dei cicli**”<sup>52</sup>; tranquilli, cercherò di ridurla al minimo indispensabile.

---

51 Il commutatore migliore potrebbe non essere unico (ad esempio, potreste aver trovato due diversi commutatori che cancellano 3 mosse); quale scegliere? Se siete a corto di tempo, sceglietene semplicemente uno dei due, a caso. Se invece avete più tempo a disposizione, potete anche pensare di provare entrambe le soluzioni.

52 Una lettura interessante, per approfondire l'argomento, è la discussione generata da [questo post](#) nel “FMC thread” su [speedsolving.com](#). Segnalo in particolare [questo post](#), una derivazione matematica della “regola” per trovare una combinazione di due 3-cycles per risolvere un 5-cycle; [quest'altro](#), che è in pratica la stessa spiegazione data in questa guida (ed è in effetti il post da cui ho imparato questo metodo “one-pass”); infine, [questa](#) è una solve di esempio che impiega questo metodo.

I matematici potrebbero avere una certa familiarità con l'argomento: si tratta in pratica di teoria delle permutazioni.

Anzitutto, con la notazione comunemente usata, il nostro ciclo si scrive come:

(1 2 3 4 5)

il che significa che l'angolo 1 deve andare in 2, il 3 in 3, ... e il 5 in 1. Con questa notazione, il ciclo appena scritto è equivalente a (2 3 4 5 1), (3 4 5 1 2) eccetera. Quello che vogliamo fare noi è spezzare questo 5-cycle in due 3-cycle, ad esempio:

(1 2 3) (3 4 5)

Tuttavia questa scomposizione non funziona. Se volete sapere perché potete leggere alcuni dei post linkati nelle note; oppure, se preferite un approccio più pratico... provate! Quello che conta è che le possibili scomposizioni corrette sono:

(1 2 3) (4 5 1)

(2 3 4) (5 1 2)

(3 4 5) (1 2 3)

(4 5 1) (2 3 4)

(5 1 2) (3 4 5)

Cioè, **il primo numero del primo ciclo deve essere uguale all'ultimo del secondo ciclo.**

Notate che l'**ordine conta**: vedete infatti che (1 2 3) può essere seguito da (4 5 1), oppure preceduto da (3 4 5). Questo significa che, una volta trovato un buon commutatore che risolve, per esempio, (1 2 3), per completare la solve potete scegliere un commutatore che risolve (4 5 1) in un punto *successivo*, oppure un commutatore che risolve (3 4 5) in un punto *precedente*. Potete fare la stessa cosa con una qualunque delle 5 coppie di cicli.

Questo metodo, se pur più veloce del primo, non permette però di controllare possibili “insertions nelle insertions”; consiglio quindi di usarlo solo se siete a corto di tempo, o come “analisi preliminare”: provate a vedere a quante mosse riuscite ad arrivare con questo procedimento, e nel secondo passaggio controllate solo i commutatori che sembrano portare a risultati migliori.

## Insertions Multiple: 5 Spigoli

Come al solito, anche nel caso dei 5-cycles la controparte per gli spigoli è in genere da evitare. Nel caso decidiate di tentare questa strada, potete procedere nello stesso modo descritto per gli angoli.

C'è però una situazione in cui 5 spigoli potrebbero essere risolti con pochissime mosse: si tratta del caso in cui sia possibile sfruttare il **5-cycle di 6 mosse**:

M' U M U'

anche con qualche mossa di setup, potrebbe risultare molto vantaggioso. Cercate di imparare quali casi questo risolve e di conoscere possibilmente anche alcune varianti, come lo “shift” L F L' R U' R'. Se doveste imbattervi in uno skeleton che lascia un 5-cycle di spigoli da risolvere, è una buona idea numerare gli spigoli e ripercorrere velocemente la solve per vedere se riuscite ad inserire uno di questi algoritmi. Comunque, non sperateci troppo.

## Altre Insertions: 2 Angoli e 2 Spigoli

A volte capita di trovare uno skeleton dove restano da risolvere 2 angoli e 2 spigoli che formano un

doppio swap (ad esempio, un PLL come la J, la T, la V e molti altri).

In questi casi, è molto utile conoscere alcuni algoritmi da **10 mosse**:

Fw2 R D R' Fw2 R D' R D R2 (J perm)

Rw' U Rw' U2 R B' R' U2 Rw2 B' (T-perm + corner twist)

Ne esistono anche moltissimi da 11 mosse<sup>53</sup>, tra cui:

R U2 R' U' R U2 L' U R' U' L (J perm)

R2 D B2 D' Fw2 D B2 D' Fw2 R2 U' (J perm)

R2 Uw R2 Uw' R2 F2 Uw' F2 Uw F2 U' (T perm)

R' U R U2 L' R' U R U' L U2 (J perm + corner twist)

Oltre a tutti questi algoritmi (a cui se ne aggiungono comunque altri), anche i loro inversi e tutte le versioni “shiftate” risolvono un doppio swap di 2 angoli e 2 spigoli.

Nota bene: **gli inversi di questi algoritmi risolvono esattamente lo stesso caso**. Quindi, solo con questa osservazione, raddoppiate le vostre possibilità di cancellazioni senza dover imparare altri algoritmi.

Non aspettatevi miracoli in termini di cancellazioni, ma più algoritmi conoscete meglio è.

Un esempio è dato da [questa solve](#), che però usa anche la tecnica del NISS, che verrà spiegata in seguito.

## Altre insertions: Coniuga e Risolvi

Una particolare situazione risolvibile con un'insertion è quella in cui mancano da risolvere 4 spigoli e 4 angoli ed entrambi questi gruppi formano un 4-cycle.

La tecnica per risolvere questa situazione è la seguente: posizionare tutti i pezzi in questione su una sola faccia (setup), di modo che una singola mossa di quella faccia risolva i due 4-cycles, eseguite quella mossa (risolvi) e poi rimettere i pezzi al loro posto (anti-setup). La stessa tecnica può funzionare quando gli 8 pezzi formano in tutto quattro 2-cycles: in questo caso la mossa per risolverli sarà di 180° (ad esempio U2).<sup>54</sup>

Un'esempio è dato da [questa solve](#). Io ho trovato lo skeleton e ho risolto il resto con 3 insertions, mentre Mirek Goljan ha suggerito un'insertion ancora più interessante, usando questo metodo:

Scramble: R2 L2 D2 F2 D' R2 U' B2 D' F2 U2 F' D2 L' F U B F2 U2 F2 L

Soluzione: U2 F B' L2 D2 F' U F2 L' B2 L F2 L' B' L' D L B' D L U L' D' L U2 R F2 R' L2 B

Skeleton: U2 F B' L2 D2 F' \* R F2 R' L2 B

Insertion: \* = (B D R2 B R' B2 D U2 F') U (F U2 D' B2 R B' R2 D' B')

Esistono anche degli algoritmi per l'ultimo strato che funzionano allo stesso modo: un esempio è (R B2 R2 U2 R) B (R' U2 R2 B2 R'), e ne potete trovare altri [qui](#).

---

<sup>53</sup> In alcuni casi, 10 mosse + 1 di AUF.

<sup>54</sup> Nel caso i pezzi non siano 8, oppure non formino due 4-cycle, si può usare uno stratagemma simile, che spiegherò in seguito (il “NISS inverso”).

## Numero di Mosse Richieste (una Stima)

Di seguito darò una stima del numero di mosse che possono richiedere i più comuni tipi di insertions. Si tratta di una **stima euristica**, non ricavata matematicamente, e dovete tenere in considerazione questi numeri che dipendono comunque anche da:

- Il numero di commutatori/algoritmi che conoscete e la vostra abilità nel riconoscerli.
- La lunghezza dello skeleton: con skeleton più lunghi avrete più possibilità per inserire un algoritmo e potete quindi sperare in un numero maggiore di cancellazioni (ma non per questo dovrete preferire uno skeleton lungo ad uno corto!).

Queste stime vi saranno utili nel caso in cui dobbiate decidere se vi conviene o meno dedicare del tempo alla ricerca di insertions: se puntate ad una soluzione sotto le 30 mosse, se dovete risolvere 3 angoli e avete uno skeleton da 23 mosse probabilmente ci riuscirete, mentre se lo skeleton è da 25 avrete bisogno di un po' di fortuna.

Potrete anche in questo modo confrontare diversi tipi di skeleton: uno skeleton con 4 angoli da risolvere in 18 mosse è probabilmente migliore di uno che lascia 3 angoli in 25.

Ecco quindi quali sono delle buone stime:<sup>55</sup>

- 3-cycle di angoli: 6 mosse.
- 3-cycle di spigoli: 7 mosse (in questo caso il numero è molto variabile).
- 2 angoli flippati, 2 commutatori: 8 mosse.
- 3 angoli flippati, 2 commutatori: 9 mosse.
- 4 angoli, 2 commutatori: 10 mosse.
- 5-cycle di angoli, 2 commutatori: 11 mosse.
- 2 angoli e 2 spigoli, doppio swap: 10 mosse.

## Insertion Finder

Uno strumento utile per trovare delle insertions, e quindi controllare se vi siete lasciati sfuggire qualcosa nelle vostre soluzioni, è [insertion finder](#), sviluppato da Baiqiang Dong: permettere di trovare fino a 4 insertions per risolvere uno skeleton.

Se nei casi più semplici (3 angoli o 3 spigoli) è molto utile, nelle situazioni più complesse riesce a trovare situazioni molto elaborate, il che non sarebbero possibili per una persona con un'ora a disposizione: usatelo responsabilmente!

## Ripassa e modifica

Se vi bloccate dopo un buon inizio, provate con questa semplice tecnica: ripercorrete tutte le mosse che avete fatto fino a che arrivate ad un punto in cui c'è almeno una faccia del cubo che contiene tutti pezzi che non avete ancora risolto; a questo punto, muovendo quella faccia non distruggerete

---

<sup>55</sup> Per lo più presi da [qui](#), leggermente modificati secondo il mio parere personale.

nessuno dei blocchi già costruiti: muovetela! Avete 3 possibilità (per esempio, U, U2 e U') e in questo modo otterrete 3 nuovi inizi che sono solo leggermente (1 mossa) più lunghi di quello che avete trovato. Una mossa potrebbe essere un buon prezzo da pagare per una migliore continuazione!

## Get Lucky!

La fortuna non è certo una tecnica da imparare, ma ricordatevi sempre che, almeno nel caso del Fewest Moves, **bisogna cercarsela**: una soluzione semplice che finisce con un LL skip non è meno valida di una solve complessa dove avete avuto sfortuna, se risolvono il cubo nello stesso numero di mosse. Questo è uno dei motivi per cui è bene provare più alternative possibili: è molto più probabile trovare uno skip se si provano 100 strade diverse anziché 10 o 20.

### Primo Esempio: Inserire l'Ultima Coppia

Quando avete completato un F2L-1, una delle possibilità che avete è di completare i primi due strati inserendo l'ultima coppia. In genere non è una buona soluzione, a meno che non abbiate un colpo di fortuna e non vi capiti un ultimo strato molto semplice. Quindi, per aumentare le possibilità che ciò accada, provate ad **inserire l'ultima coppia in tutti i modi** che vi vengono in mente.<sup>56</sup>

### Secondo Esempio: Come Usare gli Algoritmi

Innanzitutto, è fondamentale saper riconoscere le **simmetrie** nei vari casi: l'OLL che si risolve con  $F R U R' U' F'$  è simmetrico rispetto al piano S, quindi lo stesso caso può essere risolto da  $B' R' U' R B U^{57}$ . Se avete intenzione di usarlo, provate anche con il suo simmetrico per aumentare le possibilità di uno skip o di cancellazioni.

Un caso estremo è l'OLL risolvibile con  $R U^2 R' U' R U R' U' R U' R'$ : con lo stesso algoritmo o il suo simmetrico  $L' U^2 L U L' U' L U L' U L$  si può risolvere lo stesso caso da 4 angolazioni diverse, e da altre 4 se si usano anche gli algoritmi inversi.<sup>58</sup>

In secondo luogo, non è necessario che utilizziate un algoritmo per lo scopo per cui lo avete imparato. Riprendendo l'algoritmo di prima,  $F R U R' U' F'$ , è probabile che voi lo conosciate come OLL; potete però decidere di usarlo **ignorando l'effetto che ha sugli angoli**: se finite l'F2L e vi ritrovate con due spigoli opposti da orientare, potete provare questo algoritmo da quattro diverse angolazioni e sperare di rimanere con un qualche caso più semplice (ad esempio una Sune).

---

56 Con una coppia già formata ne avete almeno 3 di semplici:  $R U R'$ ,  $R U^2 R'$  e  $R' F R F'$ .

57 Questo caso è simmetrico anche per quanto riguarda la permutazione degli angoli.

58 Anche per questo caso la permutazione degli angoli rimane sempre la stessa, in quanto non viene modificata.

### 3. Strumenti Avanzati

Nel capitolo precedente sono state introdotte le tecniche di base, necessarie per trovare una buona soluzione. In questo capitolo verranno invece forniti strumenti più avanzati, che non sono strettamente indispensabili, ma consentono di sbloccarsi in situazioni difficili e aumentano le vostre possibilità di riuscita.

#### Scramble Inverso

Se non riuscite a trovare un buon inizio, potete usare lo scramble inverso: se trovate una soluzione per la sequenza di mosse inversa<sup>59</sup>, vi basterà invertire la sequenza trovata per avere una soluzione per lo scramble normale. Sembra contorto, ma in realtà è un concetto semplice.

Prendiamo come esempio l'[ex record nordamericano di Tim Reynolds](#):

Scramble: D2 L2 B R2 U2 F' L2 U2 B2 L2 F' D L2 B U L' U2 L' F' R'

Scramble Inverso: R F L U2 L U' B' L2 D' F L2 B2 U2 L2 F U2 R2 B' L2 D2

Applicate lo scramble inverso e risolvete con:

R' U F' L2

F2 D' B' \* D2 B

R2 F R2 F' R2

F D' F' D

Insertion al \*: B' U2 B D B' U2 B D', 2 mosse si cancellano.

La soluzione così trovata è quindi:

R' U F' L2 F2 D' B2 U2 B D B' U2 B D B R2 F R2 F' R2 F D' F' D

Ma questa risolve lo scramble inverso; la soluzione per lo scramble di partenza è:

D' F D F' R2 F R2 F' R2 B' D' B' U2 B D' B' U2 B' D F2 L2 F U' R

Un errore comune è quello di credere che lo scramble inverso sia completamente scorrelato da quello di partenza. In realtà sono molto simili: ad esempio, se usate lo ZZ, vi accorgete che, per qualunque orientamento, i due scramble hanno lo stesso numero di spigoli “sbagliati”, ma in posizioni diverse. Vi accorgete anche che eventuali blocchi di pezzi presenti in uno sono presenti anche nell'altro, ma a volte comprendono pezzi di altri colori e sono in altri post. La regola generale è questa: **se nello scramble normale il pezzo X è al posto del pezzo Y, nello scramble inverso il pezzo Y sarà al posto del pezzo X.** Quindi, pezzi risolti e pezzi posizionati ma flippati saranno rispettivamente pezzi risolti e pezzi posizionati ma flippati<sup>60</sup> nello scramble inverso. Blocchi di pezzi “fissi” resteranno invariati, mentre blocchi “mobili”<sup>61</sup> saranno formati da altri pezzi e posizionati altrove.

Oltre ad essere una tecnica utile in sé, nel caso in cui ci si blocchi all'inizio di una solve o

59 L'ho già spiegato, ma preferisco ripeterlo: l'inverso di, per esempio, F R U' è U R' F', non F' R' U e non U' R F!

60 Nel caso degli angoli, saranno flippati nel verso opposto.

61 Per blocchi “fissi” intendo quei blocchi, come un 2x2x2 o un 2x2x3, che non hanno libertà di muoversi rispetto ai centri; al contrario, i blocchi “mobili” sono i 3x2x1, i 2x2x1 e le coppie angolo/spigolo.

semplicemente per avere più possibilità da esplorare, i concetti appena spiegati sono alla base delle tecniche spiegate nel prossimo paragrafo.

## Pseudo Blocchi, Premoves e NISS

Per rendere più chiaro ciò che intendo per “pseudo blocchi” sarà meglio un esempio pratico:

Scramble: F' L2 F2 U2 R2 B R2 F' R2 D2 U2 L' U' B' U R U L2 F2 L'

R2 F forma un blocco 2x2x1. L'ideale sarebbe cercare di espanderlo ad un 2x2x2 in 2-3 mosse, ma le 4 mosse necessarie (L' U B' D) sono un po' troppe. Provate però a fare L2 D': quello che otteniamo non è proprio un blocco 2x2x2, ma uno *Pseudo* blocco 2x2x2. Infatti, possiamo pensare che lo strato D sia provvisoriamente spostato di una mossa D2, che possiamo fare alla fine per mettere tutto a posto. Ad esempio, una continuazione in stile CFOP potrebbe essere:

B' U2 R' U2 R2 U R

U2 F' U F U' F' U' F

L U2 L'

B L B' U' B U L U' B'

F2 D' L2 D F2 R2 D B2 D' R2

D2

In questo caso la mossa D2 poteva essere fatta anche prima dell'OLL, o tra OLL e PLL, ma nel caso in cui non completiate l'F2L come passaggio intermedio deve essere fatta alla fine.

Questa “pseudoness<sup>62</sup>” rende però difficile proseguire nella solve (per chi non è esperto, riconoscere quali sono le coppie dell'F2L può essere difficile, per esempio). C'è chi consiglia di sforzarsi di proseguire nella risoluzione in questo modo, perché può essere utile saper ragionare con gli pseudo blocchi, ma c'è un trucco che semplifica di molto le cose: basta eseguire la mossa che andrebbe fatta alla fine (in questo caso, D2) **prima dello scramble<sup>63</sup>**. Provare per credere!

Scramble: D2 F' L2 F2 U2 R2 B R2 F' R2 D2 U2 L' U' B' U R U L2 F2 L'

Soluzione:

R2 F L2 D'

B' U2 R' U2 R2 U R

U2 F' U F U' F' U' F

L U2 L'

B L B' U' B U L U' B'

F2 D' L2 D F2 R2 D B2 D' R2

Una volta trovata la soluzione in questo modo, ricordatevi che la premove (o le premoves, se ce n'è più di una) vanno comunque **aggiunte alla fine** della soluzione, per risolvere lo scramble di partenza.

Si può usare anche **più di una premove**: prendete come esempio [questa soluzione](#), dove ho trovato le due premoves separatamente. Un caso un po' più difficile da riconoscere è quello di pseudo blocchi che richiedono più di una premove: prendiamo come esempio lo scramble precedente e il

---

62 Uso il termine inglese perché non mi viene in mente una traduzione adeguata. Spero che il significato sia chiaro.

63 Da qui il nome “premove(s)”.

nostro  $2 \times 2 \times 1$  iniziale (R2 F); anche i più esperti faranno fatica a notare che aggiungendo come premove D F' si ottiene un blocco  $2 \times 2 \times 2$ :

Scramble: D F' F' L2 F2 U2 R2 B R2 F' R2 D2 U2 L' U' B' U R U L2 F2 L'

$2 \times 2 \times 2$ : R2 F

Tuttavia tali premove non sono difficili da individuare se si conosce la tecnica del **NISS** (Normal-Inverse Scramble Switch). Per riuscire a capire come funziona è necessaria un po' di teoria.<sup>64</sup>

**Lo scramble e la soluzione possono essere considerati come un'unica sequenza di mosse, che non modifica in alcun modo il cubo.** Per esempio:

Scramble: A B C D

Soluzione: p q r s

La sequenza A B C D p q r s riporta il cubo nello stato di partenza. Allo stesso modo, tutte le versioni “shiftate” di questa sequenza:

s (A B C D p q r s) s' = s A B C D p q r

r s (A B C D p q r s) s' r' = r s A B C D

q r s (A B C D p q r s) s' r' q' = q r s A B C D

p q r s (A B C D p q r s) s' r' q' p' = p q r s A B C D

D p q r s (A B C D p q r s) s' r' q' p' D' = D p q r s A B C

...

non modificano il cubo.<sup>65</sup>

Nel nostro esempio precedente con una premove D2, la sequenza sarebbe stata:

(Scramble) R2 F L2 D' (Altre mosse) D2

E l'eseguire una mossa prima dello scramble e risolvere questa nuova situazione anziché quella originale non significa altro che prendere la versione shiftata di questa sequenza:

D2 (Scramble) R2 F L2 D' (Altre mosse)

In altre parole, possiamo considerare “R2 F L2 D' (Altre mosse) D2” come soluzione per “(Scramble)”, oppure “R2 F L2 D' (Altre mosse)” come soluzione per “D2 (Scramble)”.<sup>66</sup>

Questo dovrebbe essere sufficiente a capire le premove. Basandosi su questa logica, possiamo costruire una soluzione lavorando in parte sullo scramble normale e in parte sullo scramble inverso. Come? Consideriamo sempre il solito scramble e il solito inizio R2 F. Sappiamo che, volendo partire da qui, alla fine la nostra soluzione sarà del tipo R2 F (W), dove (W) indica una sequenza di mosse. La sequenza che non modifica il cubo è in questo caso:

(Scramble) R2 F (W)

Abbiamo detto prima che nemmeno l'inversa di questa sequenza modifica il cubo (equivale infatti a trovare la soluzione con la tecnica dello scramble inverso):

---

<sup>64</sup> Presa [questa spiegazione](#) di Tomoaki Okayama.

<sup>65</sup> Inoltre, nemmeno le sequenze inverse a queste modificano il cubo.

<sup>66</sup> Si può anche vedere lo scramble come una soluzione per la soluzione.

(W)' F' R2 (Scramble Inverso)

Per quanto abbiamo detto prima, possiamo considerare “(W)' F' R2” come soluzione per “(Scramble Inverso)” ma anche, per esempio, “(W)” come soluzione per “F' R2 (Scramble Inverso)”. In questo modo, potete pensare all'**inverso delle mosse trovate sullo scramble originale come premoves per lo scramble inverso**. Trovate quindi la soluzione (chiamiamola (K), che nel nostro esempio corrisponde a (W)') per lo scramble inverso con le premoves, e il gioco è fatto: basterà scrivere come soluzione R2 F (K)'.

Il procedimento può essere ripetuto: supponiamo che sullo scramble inverso con le premoves io abbia trovato le mosse F D', che formano un blocco 2x2x2, ma nessuna buona continuazione. A questo punto posso tornare a lavorare sullo scramble normale, mettendo come premoves D F'. Infatti, la sequenza “magica” è ora:

F' R2 (Scramble Inverso) F D' (Mosse non ancora trovate)

Invertendola si ottiene un'altra sequenza magica:

(Inverse delle mosse non ancora trovate) D F' (Scramble) R2 F

E possiamo considerare D F' come premoves per lo scramble di partenza e ripartire con R2 F.

Un esempio (preso da [qui](#)) renderà tutto più chiaro:

Scramble: (fmc.mustcube.net #265) F2 B' R' B D B F' R B F L' D2 U2 B2 D U' B' F D2 L' F2 B' R2 B2 R L' F2 U' F' L

Soluzione (di Guus Razoux Schultz): D' B' U B2 D' L F2 L' D R U2 R U2 B' R' U2 RU' R B' R' F2 U' L B' (25 HTM)

Spiegazione:

Bell'inizio sullo scramble inverso: B L' U F2

Applicando sullo scramble normale: [pre-moves F2 U' L B'] bella continuazione: D' B' U B2 R2

Applicando sullo scramble inverso: [pre-moves R2 B2 U' B D]: B L' U F2

Semplice continuazione sullo scramble inverso:

F2L: R B R' U R' U2 R B

LL: U2 R' U2 R' D' L F2 L' D R2

Correzione premoves: R2 B2 U' B D, 2 mosse si cancellano (25)

Per rendere più compatta la scrittura di soluzioni di questo tipo, ho proposto tempo la seguente notazione<sup>67</sup>: semplicemente, scrivere i pezzi di soluzione trovati sullo scramble inverso tra parentesi. Ad esempio, la solve precedente diventa:

Bell'inizio: (B L' U F2)

Bella continuazione: D' B' U B2 R2

F2L: (R B R' U R' U2 R B)

LL: (U2 R' U2 R' D' L F2 L' D R2)

Notazione semplice e compatta, ed evita di dover scrivere ogni volta “Sullo scramble normale/inverso, con premoves...”.

---

<sup>67</sup> Non ha ancora ottenuto una grande popolarità, quindi precisate sempre che la state usando, dando una breve spiegazione del tipo “mosse tra parentesi eseguite sullo scramble inverso”.

## NISS Inverso

È una tecnica poco usata, ma occasionalmente utile: può essere considerata un “miglioramento” sia delle insertion del tipo “Coniuga e Risolvi” sia, per certi versi, della tecnica “Ripassa e Modifica”. Supponiamo che riusciate a trovare uno skeleton che risolve tutto tranne un certo numero di pezzi (da 4 a 8); vorreste provare ad inserire un algoritmo per risolverli, ma se questi non hanno tutti un colore in comune (non appartengono alla stessa faccia) può essere difficile riconoscere l'algoritmo da usare. Il trucco è questo: se trovate un punto della solve in cui tutti questi pezzi si trovano su una faccia sola, potete **“spezzare” la solve** in quel punto, usando tutte le mosse successive come premoves (il che spiega il nome “NISS inverso”); a questo punto, vi resterà un “Ultimo Strato” da risolvere. Eventualmente, se in nessun punto della solve questi pezzi si trovano tutti coniugati su un'unica faccia, potete usare delle mosse di steup. Prendiamo [questa solve](#) come esempio:<sup>68</sup>

Scramble: L2 D2 F2 U' L2 D L2 D2 B2 U' F' U' R' D B2 L D B' F' R'

Soluzione: F2 U R' U' F D2 F R F D F2 R F R2 D R D2 R' F2 U2 R D R' D2 B2 R2 (26 HTM)

Spiegazione:

Tutto tranne 5 pezzi: F2 U R' U' F D2 \* F' U2 R D R' D2 B2 R2 (La soluzione è semplice da seguire se si usa R2 come premove). Inserendo al \* le mosse F R si coniugano i pezzi rimasti su un'unica faccia. Prendiamo quindi lo skeleton:

F2 U R' U' F D2 F R + R' F' F' U2 R D R' D2 B2 R2

E “spezziamolo” al +: mettiamo come premoves per lo scramble normale R' F' F' U2 R D R' D2 B2 R2; ora:

Premoves: R' F2 U2 R D R' D2 B2 R2

F2L: F2 U R' U' F D2 F R

LL: F D F2 R F R2 D R D2

## Algoritmi Utili

Come avrete notato, nell'ultima solve d'esempio ho usato un OLL forse poco conosciuto: a meno di rotazioni, R U R2 F R F2 U F (U2). Questo in particolare è molto utile, perché è il più corto algoritmo che modifica l'orientamento ma non la permutazione di un certo numero di pezzi.

In generale è utile conoscere qualche algoritmo per l'ultimo strato, specialmente quelli più corti, **fino a 9 o 10 mosse**; [qui](#) potete trovare una lista completa (a meno di rotazioni, simmetrie e inversi).<sup>69</sup>

Se decidete di infrangere la regola del “mai costruire un F2L senza influenzare l'ultimo strato” (a volte vale la pena tentare!) potete sperare che l'algoritmo per risolvere l'LL sia molto breve: in questo caso, più ne conoscete meglio è!<sup>70</sup>

---

68 In realtà in questa soluzione l'algoritmo da usare era facile da riconoscere semplicemente segnando i pezzi con delle frecce e delle X e non ho usato questa tecnica, ma la soluzione si adatta bene al nostro caso.

69 Alcuni algoritmi hanno un nome e/o dei commenti a fianco, che ho messo per uso personale: ignorateli pure. In futuro, potrei pubblicare una lista ben commentata a più ordinata.

70 In ogni caso, quando usate un algoritmo per l'ultimo strato ricordatevi di provare a mettere l'AUF (“Adjust Upper Face”) sia prima che dopo, nella speranza di cancellare qualche mossa con le mosse precedenti o con eventuali premoves.

Ci sono altri due motivi per cui sarebbe bene imparare qualche algoritmo, per lo meno quelli tra le 6 e le 9 mosse: il primo è che anche nel caso non conosciate l'algoritmo esatto per l'ultimo strato oppure non abbiate nemmeno completato l'F2L, uno di questi algoritmi potrebbe lasciarvi in una situazione favorevole (ad esempio, 3 angoli da risolvere inserendo un commutatore), magari cancellando qualche mossa.

Il secondo è che, studiando questi algoritmi, potreste imparare nuove tecniche per unire i blocchi o per completare l'F2L. Prendiamo per esempio la T perm ottimale:

$R2 U_w R2 U_w' R2 y L2 U_w' L2 U_w L2 (U')$

non è altro che la ripetizione di un algoritmo utile nell'F2L.

Insomma, oltre ad imparare gli algoritmi si può **imparare dagli algoritmi**.

## Analisi delle Coppie

Questa è una tecnica molto oscura, basata sull'intuizione, e non è mai stato dimostrato che porti ad un effettivo vantaggio durante la risoluzione. Di che cosa si tratta? Una buona “analisi delle coppie” consiste nel prendere in considerazione alcune cose riguardo allo scramble, tra cui:

- Coppie già pronte. Non c'è molto da dire.
- Coppie che si possono formare **con una sola mossa**. Per trovarle, potete usare la tecnica “brute force” descritta prima, oppure cercare di riconoscerle a colpo d'occhio. Inoltre, può essere utile saper riconoscere le **pseudo coppie**, cioè quelle che sono coppie risolvibili con una sola mossa sullo scramble inverso. Le mosse che formano una coppia, sia sullo scramble normale che su quello inverso, possono essere prese come prima mossa, o come remove per l'inverso dello scramble che si sta considerando.
- **Coppie sbagliate**: sono quelle coppie angolo/spigolo in cui uno dei due pezzi è flipato. Intuitivamente, queste coppie non sono un bene, ed è opportuno cercare di distruggerne il più possibile.

Non c'è molta documentazione riguardo a questa tecnica, specialmente per le coppie sbagliate. Una buona analisi è quella fatta da Guus Razoux Schultz su [questo scramble](#).

## 4. Come Allenarsi

Per migliorare, molti dicono che quello che serve è “allenamento, allenamento, allenamento”. È senz'altro vero, ma bisogna anche sapere *come* allenarsi: ecco alcuni consigli pratici su come allenarsi per migliorare nel FMC.

### No Time Limit e Simulazioni di Gara

Cercare sempre di riprodurre una situazione da gara imponendosi di completare la propria soluzione in un'ora non è la cosa migliore da fare: al contrario, suggerisco di **non porsi limiti di tempo** e di provare e riprovare uno stesso scramble fino a che non si è soddisfatti del risultato.

Questo non vuol dire che fare solve con un tempo limite sia un male: anzi, dà una buona indicazione del vostro livello e vi aiuta a trovare una strategia<sup>71</sup> per la gestione del tempo. A questo proposito consiglio la [competizione settimanale online sul sito di David Adams](#).

Un buon compromesso è provare per la prima come se si fosse in gara, per poi continuare a tentare di migliorare la soluzione fino a che non si raggiunge un risultato per voi buono.

### Confrontatevi con i Maestri (e Studiate le loro Soluzioni)

Quando volete allenarvi, vi suggerisco di trovare uno scramble per il quale si trovino in rete soluzioni di cuber esperti: in questo modo potrete confrontare la vostra soluzione con la loro e scoprirete se vi siete lasciati sfuggire un buon inizio o altro.

Inoltre, come per l'allenamento nel blockbuilding e nei vari metodi di risoluzione, assolutamente obbligatorio è studiare le soluzioni dei maestri: ne potete trovare molte su <http://fmcsolves.cubing.net/>.

### Scramble Difficili

Per vedere come sapete cavarvela “nel caso peggiore”, vi consiglio di provare alcuni scramble che, a detta degli esperti, sono difficili. Potete trovare una lista di scramble ostici [qui](#).

### Allenamenti Mirati

Se pensate che la vostra difficoltà principale stia nel trovare un buon inizio, allenatevi su quello: prendete uno scramble, cercate un 2x2x2, un 2x2x3 o altro che vi soddisfi e poi cambiate scramble. Potete applicare lo stesso procedimento al F2L-1 o a qualunque altro substep.

### Scramblare velocemente

Anche se non è di vitale importanza, se si usano tecniche come il NISS, o semplicemente se, come me, tendete a risolvere e ri-scramblare il cubo molte volte durante una solve, dovrete cercare di essere almeno “non troppo lenti”<sup>72</sup> nell'applicare uno scramble, e soprattutto di essere **precisi** (cioè

---

<sup>71</sup> Alcuni consigli pratici a riguardo verranno dati nel capitolo 6.

<sup>72</sup> 10 secondi o meno è un buon obiettivo per uno scramble da 20 mosse circa.

di non fare errori).

## **Studiate!**

Ultimo ma non meno importante. Studiate questa guida, studiate da altre risorse, studiate algoritmi e tecniche, nuove o già conosciute. Personalmente, io ho letto *due volte* da cima a fondo il “[The FMC Thread](#)” su [speedsolving.com](http://speedsolving.com).

Studiate algoritmi: esistono dozzine di set diversi, per citarne alcuni [LLEF](#) (Last Layers Edges First) o la [Summer Variation](#). Molto spesso ricordatevi che non è tanto utile *memorizzare* gli algoritmi, quanto cercare di capire *perché funzionano*.

## 5. Alcune Alternative

Il metodo “standard”, come spiegato prima, è quello di cercare di costruire uno skeleton utilizzando il blockbuilding, per poi risolvere i pezzi rimasti con delle insertions. Esistono tuttavia approcci alternativi, alcuni dei quali particolarmente degni di nota.

### Partire con l'EO

Una possibilità da tenere sempre in considerazione è quella di **partire orientando tutti gli spigoli**, come in una solve ZZ. Da qui si aprono almeno due strade.

#### EO + Blockbuilding

Dopo aver orientato tutti gli spigoli, la strada più comunemente intrapresa è quella del blockbuilding. Il vantaggio è dato dal non avere alcuno spigolo “sbagliato”, ma questo ci impedisce<sup>73</sup> di usare mosse che distruggono l'EO, limitandoci leggermente.

Tra chi ha fatto di questa possibilità un'abitudine c'è Grzegorz Łuczyna; ecco la [soluzione](#) che gli ha consentito di diventare campione europeo nel 2010:

Scramble: L D2 B' D2 B R' B' U B L B L2 B2 U2 F2 U R2 D' B2 D' B2

Soluzione:

EO: x y2 L2 D F'

EOLine: R L2 D #

2x2x3: R' U2 B2 R2 B2

Tutto tranne 3 angoli: L2 U' R' U L U' L' R U2 L' U'

3 angoli: [D2, R' U' R] a # (4 mosse si cancellano)

#### Riduzione a “Domino” (e HTA)

L'orientamento degli spigoli può essere considerato, a meno di rotazioni, una riduzione del cubo al *sottogruppo* generato dalle mosse <R, L, U, D, F2, B2>. Un ulteriore passo in questa direzione ci porta a ridurre il cubo al sottogruppo generato da <U, D, R2, L2, F2, B2>; per fare ciò, dovete:

- Posizionare gli spigoli dello strato E nello strato E.
- Orientare gli angoli.

Questa particolare riduzione è detta anche “Domino”, per l'analogia con il “cubo” di Rubik 3x3x2 (chiamato appunto “Domino Cube”). Inoltre, questi sono anche i primi due step dello Human Thistlethwaite Algorithm (o HTA), un adattamento del Thistlethwaite Algorithm. Se siete interessati ad approfondire questo metodo per il FMC, vi consiglio [questa guida](#).

Ecco un esempio di [soluzione](#) di Per Kristen Fredlund:

Scramble: D U' R' F B2 R B2 R' U2 R B2 R' U2 R B2 R' U2 R B2 R' U2 B2 F2 U D F D' B2 D F D' B2 D F D' B2 D F D' B2 F' D B' F R'

Soluzione: \_\_\_\_\_

<sup>73</sup> Nessuno vi obbliga ovviamente, ma orientare tutti gli spigoli per poi distruggere quanto appena fatto non sembra una cosa intelligente. Piuttosto, potete provare a partire con un'EO *parziale*.

R' B U' D F //EO (5/5)

L' F2 L //Riduzione Domino (3/8)

D2 L2 F2 D F2 D L2 U' R2 D2 R2 //Fine (11/19)

## Corners First

Il “Corners First” (a volte abbreviato in CF) è, più che un metodo, una classe di metodi per risolvere il cubo che, come dice il nome, prevede di risolvere **per prima cosa gli angoli**, e poi gli spigoli. Per alcuni versi, il metodo Roux può essere considerato un metodo CF.

Molte tra le persone che hanno imparato autonomamente a risolvere il cubo hanno per prima cosa trovato un approccio di questo tipo:<sup>74</sup> ragionare separatamente per angoli e spigoli rende in qualche modo più semplice una soluzione di tipo intuitivo. Inoltre, risolvendo prima gli angoli resta molta libertà per la risoluzione degli spigoli: basta pensare che gli strati centrali possono essere mossi liberamente.

Questo è però anche uno svantaggio per il FMC: le mosse degli strati interni (M, E ed S) valgono doppio! Nonostante questo, esistono almeno due cuber di ottimo livello nel FMC che usano questo tipo di tecnica: **Attila Horváth**<sup>75</sup>, che fino ad ora ha partecipato solo a due competizioni ufficiali e **Javier Cabezuelo Sánchez**<sup>76</sup>, attuale detentore del record nazionale spagnolo. Entrambi sostengono che metodi Corners First siano ottimi per il FMC, ma poco adatti al formato ufficiale con un'ora di tempo limite.<sup>77</sup>

Attila Horváth risolve gli angoli utilizzando di solito un metodo simile al Guimond (prima orienta gli angoli, poi li permuta); la posizione dei centri non è controllata in questo momento; per questa fase utilizza a volte delle premoves o il NISS. Successivamente, ripassa la soluzione trovata per gli angoli e la modifica leggermente, inserendo eventualmente mosse degli strati centrali, fino ad aver risolto 2 o 3 spigoli almeno. Infine, risolve gli spigoli rimasti, senza un ordine preciso. A volte non risolve i centri fino alla fine della solve, per poi metterli a posto con un'insertion<sup>78</sup>, che solitamente cancella molte mosse. [Esempio di soluzione](#).

Javier Cabezuelo Sánchez risolve gli angoli in maniera diversa: prima quelli del primo strato, poi gli altri. Successivamente cerca di risolvere gli spigoli inserendo mosse (o algoritmi) all'interno della soluzione appena trovata. Non usa tecniche quali lo scramble inverso, le premoves o il NISS. A differenza di Attila tiene da conto dei centri durante la risoluzione degli angoli.<sup>79</sup>

Sia Attila che Javier utilizzano esclusivamente il loro metodo e non hanno intenzione di imparare metodi più “standard”, il che violerebbe la regola del “mai porsi dei limiti nel metodo da usare”; del resto, ottengono entrambi ottimi risultati.

---

74 Ad esempio, Valery Morozov, che ha pubblicato [una guida](#) al suo metodo.

75 Profilo WCA: [2012HORV01](#); profilo su speedsolving.com: [Attila](#).

76 Profilo WCA: [2007SANC01](#).

77 Notate infatti che molti dei risultati ufficiali di Javier sono DNF.

78 Algoritmi come M E M' E' o M E2 M' E2.

79 Fonte: [questo post](#).

## 6. Consigli per le Competizioni

### Come Scrivere una Soluzione

Quando scrivete una soluzione, sia in gara che in allenamento, scrivetela **senza rotazioni**. Ci sono molti buoni motivi per farlo:

- Quando si usano le rotazioni, è più facile sbagliare.
- Le rotazioni possono nascondere delle possibili cancellazioni.<sup>80</sup>
- Mentre risolvete, se avete fatto delle rotazioni dovete sempre tenere a mente in un certo momento quale faccia è in U e quale è in F.

Ma come fare quindi a scrivere una soluzione senza rotazioni? Certamente è scomodo fare, ad esempio, un PLL sulla faccia B. Il modo c'è ed è anche semplice: tenendo l'orientamento standard, con lo schema di colori BOY (quello “normale”), basterà ricordarvi che la faccia con in centro bianco è sempre in U, quella con il centro verde in F, quella con il centro giallo in D e così via.<sup>81</sup> Ogni volta che muoverete la faccia, per esempio, bianca non dovrete preoccuparvi di come state guardando il cubo in quel momento: scrivete U e basta.

### Soluzione di Backup

Una buona abitudine, nelle gare con limite di tempo, è quella di scrivere una “soluzione di backup”. Si tratta generalmente di una soluzione non tanto buona, ma che è comunque meglio di un DNF, trovata prima degli ultimi minuti, in cui la fretta e l'agitazione possono portare a sbagliare o addirittura a non trovare una soluzione. Se, per esempio, la vostra media è sulle 35 mosse, ma dopo 20 minuti avete trovato una soluzione da 40 e l'avete scritta bene in chiaro su un foglio<sup>82</sup>, sarete molto più tranquilli per i 40 minuti rimanenti. Ci sono vari approcci possibili al trovare una soluzione di backup:

- Fissarsi un tempo limite (es.: 35 minuti) entro il quale vi imponete di aver trovato e scritto una soluzione, anche brutta; personalmente non lo faccio, ma può essere utile se vi succede spesso che alla fine dell'ora non abbiate ancora scritto niente di definitivo.
- Se per caso vi imbattete in una soluzione (es.: avete trovato un buon inizio, volete risolvere il cubo per ri-scramblarlo e nel farlo trovate un PLL skip), segnatela su un qualche foglio e tenetela da parte. Per i disordinati come me: non perdetela!
- Il procedimento che uso io: più che una soluzione di backup trovo vari *skeleton* di backup. Per esempio, io punto a trovare una soluzione da meno di 30 mosse; nel mio caso, uno skeleton che risolve tutto tranne 3 angoli in 26 mosse non è molto buono, ma se ne trovo uno me lo tengo da parte. Quando arrivo, ad esempio, a 10 minuti dalla fine, cerco un

---

80 Ad esempio, se scrivete cose come ... R z' U' ... è giusto che gli altri vi deridano.

81 Per aiutarvi a memorizzare lo schema (non che sia difficile), ricordate che il nome dei colori Blu e Rosso inizia con la stessa lettera della faccia corrispondente.

82 Anche il foglio ufficiale da consegnare: se volete cambiare la vostra soluzione potete benissimo cancellare le mosse fatte e scrivere quelle nuove.

insertion per quello skeleton.<sup>83</sup>

Quale può essere una buona soluzione di backup? Qualunque! Qualunque cosa è meglio di un DNF, specialmente ora che il formato preferibile per il Fewest Moves (nelle gare ufficiali) è la “Mean of 3”, ovvero la media secca di 3 tentativi: un solo DNF porta ad una media DNF!

## Gestione del Tempo

Il “come gestire il tempo a disposizione” è un argomento molto complesso e non me la sento di dire che i miei consigli siano completamente giusti: prendeteli con le pinze! L'insegnante migliore, in questo caso, è l'esperienza personale.

### Non Bloccarti!

Può capitare a chiunque, durante una gara, di fissarsi su un particolare inizio e non riuscire, in nessun modo, a proseguire. L'abilità di capire al volo (o quasi) quale inizio porti ad una buona continuazione sarebbe utile quanto saper leggere nel pensiero. Il consiglio che posso darvi, anche se può sembrare banale è di non bloccarvi. Se avete provato tutte le tecniche che conoscete e non avete trovato nulla, non rimanete lì a fissare il cubo sperando che si risolva da solo: tornate indietro e cambiate strada.

### Esplorare le Possibilità

Se siete informatici, matematici o simili posso tranquillamente dirvi che l'esplorare le varie possibilità di soluzione di uno scramble equivale a *visitare un [albero](#)*: scegliete voi se usare un approccio DFS (ricerca in profondità, cercare di completare una soluzione prima di esplorare altre possibilità), BFS (ricerca in ampiezza, fate “avanzare” tutte le strade che trovate di pari passo) o misto. Tenete in considerazione che da una sola soluzione parziale possono partire moltissime diramazioni “appetibili”, così come nessuna; per questo motivo non uso un metodo fisso. E inoltre fondamentale imparare a potare i rami poco promettenti (cioè, scartare soluzioni parziali non molto buone).

Spero che anche chi non è del settore abbia colto l'analogia.

---

83 Per me, una sola insertion per 3 angoli richiede circa 5 minuti.

## 7. Altre Risorse

Per concludere, ecco come promesso molte delle fonti da cui ho tratto le informazioni (o meglio, ho imparato) per questa guida e altri link utili:

- Il [sito di David Adams](#) che ospita una gara settimanale, nella quale è possibile confrontarsi con avversari di vario livello. Molto utile per confrontare varie soluzioni per uno stesso scramble.
- Il sito [fmc.mustcube.net](#), al momento offline, ospitava una gara settimanale come il precedente. Gestito da Per Kristen Fredlund.
- [Insertion Finder](#), utile per trovare insertions.
- [JARCS](#), uno strumento utile per trovare (e studiare) soluzioni ottimali a vari substep. Al momento offline.
- [fmc solves.cubing.net](#), un Blog/Database per solve Fewest Moves, ufficiali e non.
- [menas.com.br](#), raccolta di tutte le solve ufficiali di cuber brasiliani, con video di spiegazione per ognuna (in portoghese).
- [Speedsolving.com](#), punto di riferimento per il mondo del cubo in generale; in particolare:
  - [The FMC thread](#), dedicato al Fewest Moves.
  - [Fewest Moves: Tips and Techniques](#), altro thread per il FMC.
  - [Guida ai commutatori di Brian Yu](#), con traduzione in italiano di [Matteo Colombo](#).
- [Fewest Moves Thread](#) su [speedcubing.it](#).
- Il [sito di Ryan Heise](#), ottimo non solo per imparare il suo metodo, ma anche per i vari esempi per quanto riguarda le “[tecniche fondamentali](#)”.
- Il [sito di Lars Petrus](#), che contiene anche alcuni esempi di blockbuilding.
- [Video](#) di Daniel Sheppard su come migliorare nel FMC.
- Video tutorial in 5 parti di Ranzha: [\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#) [\[5\]](#)
- [Presentazione](#) di Pranav Maneriker.