

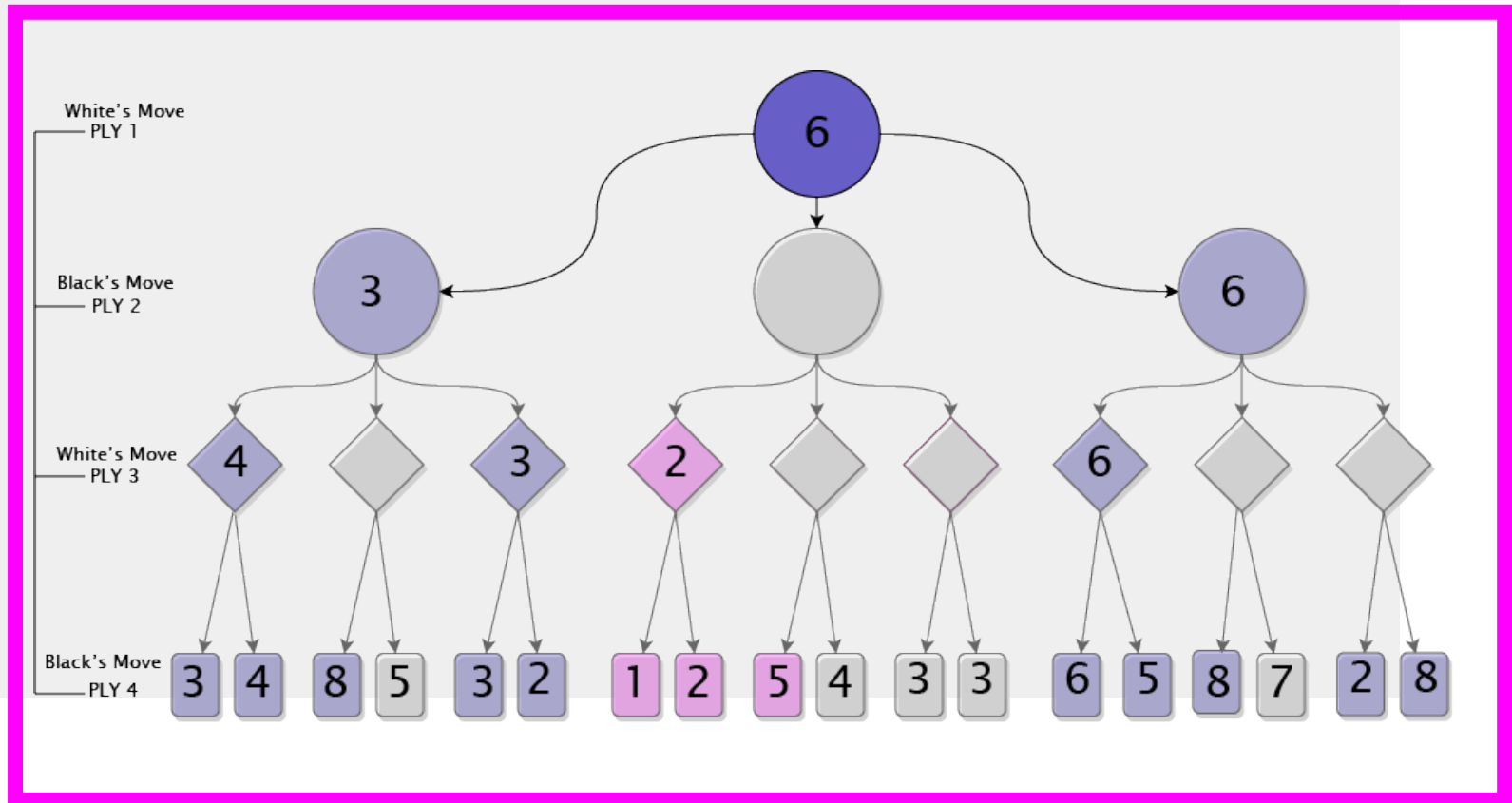
First Blood: Enhancing the Chess AI

by Lina Pulgarin-Duque

1. Alpha-beta & MTD(f)

A quick summary

Alpha-beta



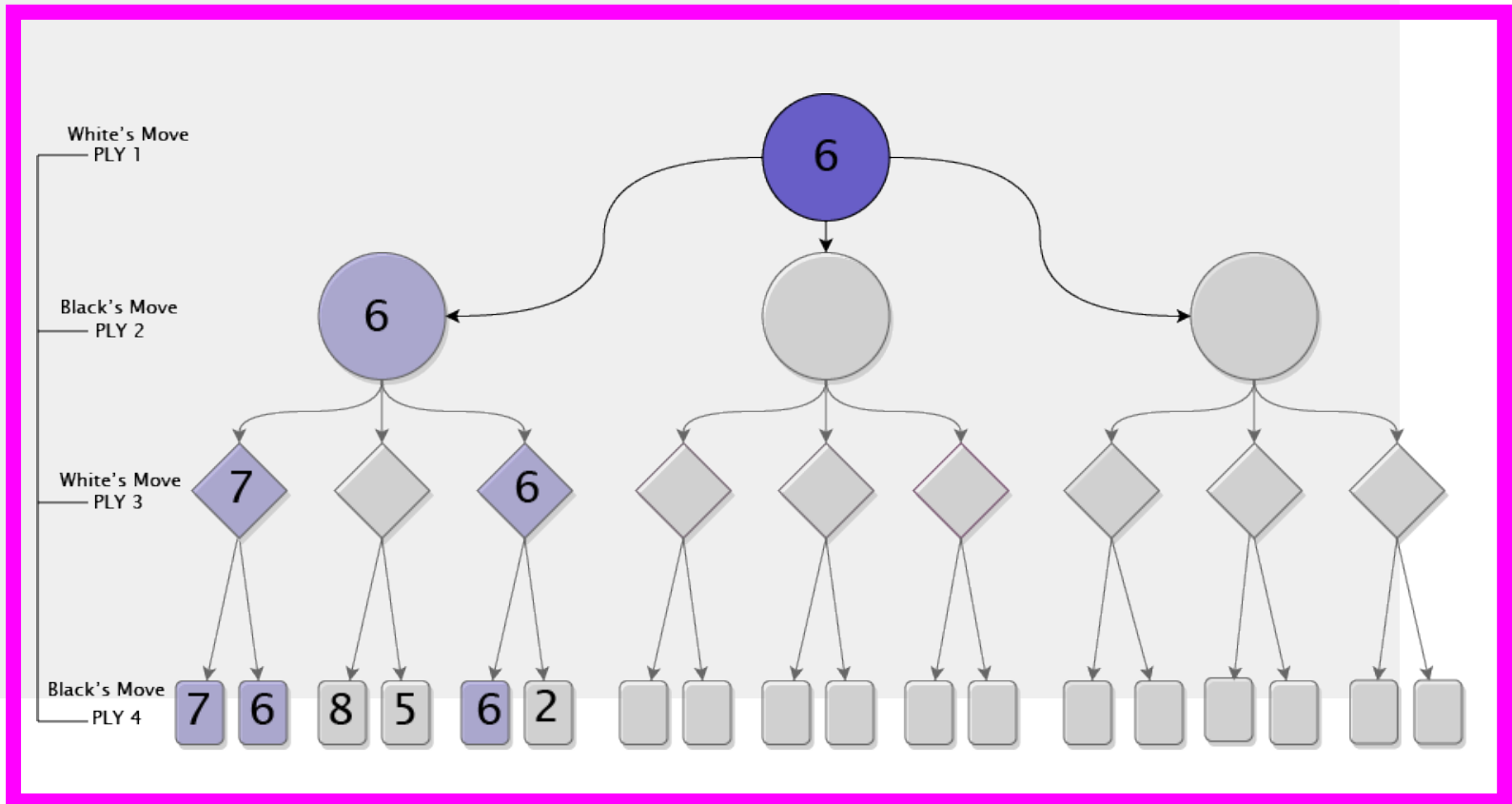
MTD(f)

- Set upper bound to **+Infinity** and lower bound to **-Infinity**
- Set alpha to target minus width
- Set beta to moving target plus width
- Call alpha-beta
- Use the return score to set new upper and lower bounds
- Repeat until lower bound > upper bound
- Return that score

2. First Blood

Method and Results

First-Blood



First Blood

MTD(f)

- Set upper bound to **+Infinity** and lower bound to **-Infinity**
- Set alpha to target minus width
- Set beta to moving target plus width
- Call alpha-beta
- Use the return score to set new upper and lower bounds
- Repeat until lower bound > upper bound
- Return that score

First Blood

- Return whenever our **lower bound > target score**
- We return something at a greater depth thanks to our database

Database

Methods for gathering moves

- ❑ Calculate each time
- ❑ Transposition Table
- ❑ Database using sqlite3

Database

Methods for gathering moves

- Calculate each time
- Transposition Table
- Database

```
iter(1)
iter(2)
Excess: 3 Reach: 5
  set best info 0
iter(3)
Excess: 2 Reach: 5
iter(4)
Excess: 1 Reach: 5
iter(5)
best: 5
p_board = 1
Greater score
Follow 5 Follow capt 1
Follow score: 48
Computer's chess_move:
e2e4 time=1.48 sec
(avg = 1.2)
Ply number: 0

 8  r n b q k b n r
 7  p p p p p p p p
 6  . . . . . . . .
 5  . . . . . . . .
 4  . . . . P . . .
 3  . . . . . . . .
 2  P P P P . P P P
 1  R N B Q K B N R

  a b c d e f g h
```

Found results
in the database
at each depth

Follow that result
for 5 plies or until
something better
comes up

Opening Book

1

Computer's chess_move:
f2f4 time=1.71 sec
(avg = 1.71)
Ply number: 0

```
8 r n b q k b n r
7 p p p p p p p p
6 . . . . . . . .
5 . . . . . . . .
4 . . . . . P . .
3 . . . . . . . .
2 P P P P P . P P
1 R N B Q K B N R

a b c d e f g h
```

Computer's chess_move:
d2d4 time=8.69 sec
(avg = 3.48)
Ply number: 2

```
8 r n b q k b n r
7 p p p . p p p p
6 . . . . . . . .
5 . . . p . . . .
4 . . . P : P . .
3 . . . . . . . .
2 P P P . P . P P
1 R N B Q K B N R

a b c d e f g h
```

Computer's chess_
move: c2c4 time=13.8
sec (avg = 13.8)
Ply number: 0

```
8 r n b q k b n r
7 p p p p p p p p
6 . . . . . . . .
5 . . . . . . . .
4 . . P . . . . .
3 . . . . . . . .
2 P P . P P P P P
1 R N B Q K B N R

a b c d e f g h
```

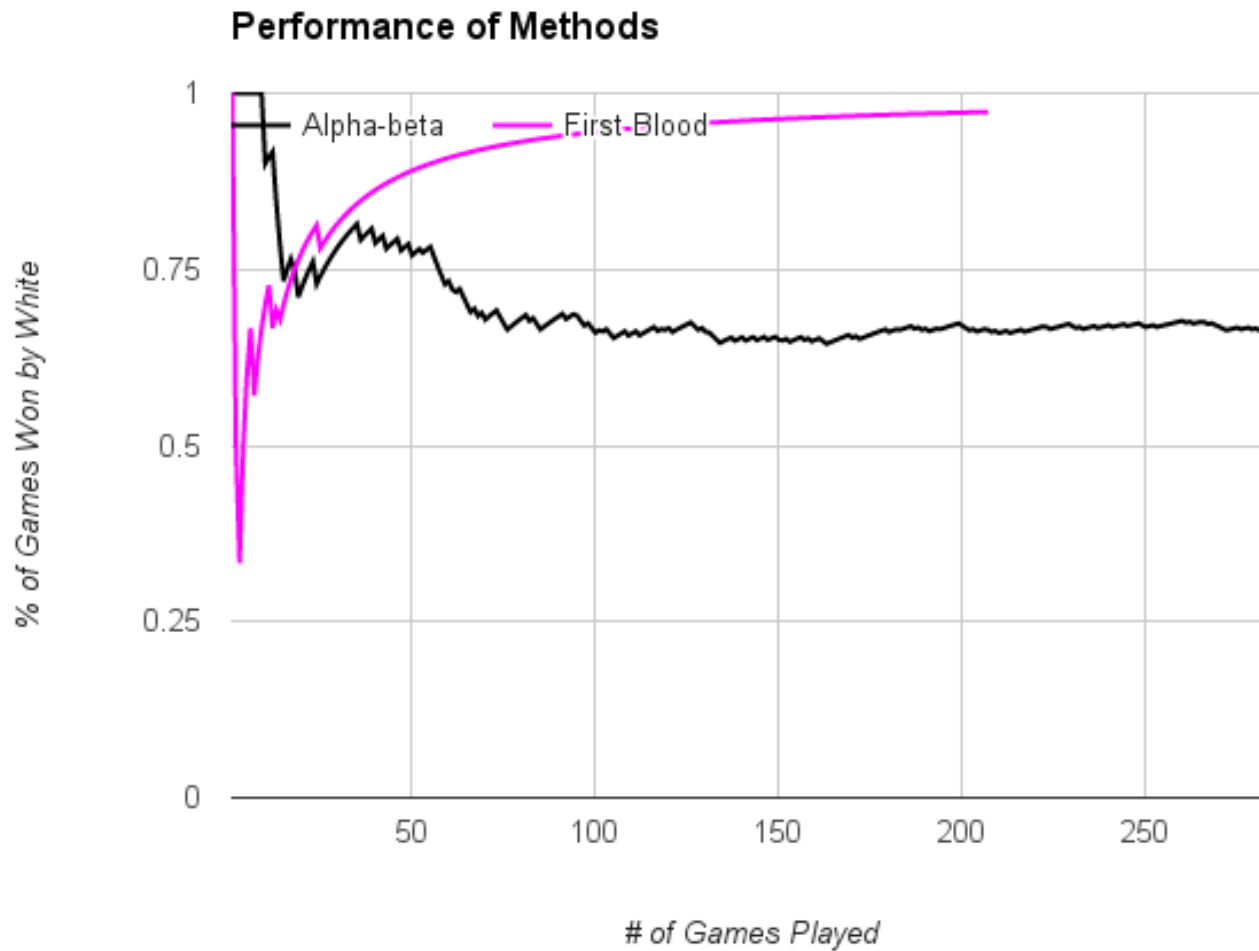
Computer's chess_move:
e2e4 time=65.2 sec
(avg = 26.4)
Ply number: 2

```
8 r n b q k b n r
7 p p p p . p p p
6 . . . . . . . .
5 . . . . p . . .
4 . . P . P . . .
3 . . . . . . . .
2 P P . P . P P P
1 R N B Q K B N R

a b c d e f g h
```

2

Results



thanks!

Any questions?

You can find me at
lpulgarinduc@gmail.com



Credits

Dr. Steven Brandt and Patrick Mancuso were both collaborators on this project.

Special thanks to Frank Loeffler for his time advising and contributing to the code.

This material is based upon work supported by the National Science Foundation under award OCI-1263236 with additional support from the Center for Computation & Technology at Louisiana State University.