

Time-Limited Iterative Deepening

Tuesday, April 21, 2020 8:58 PM

• Time Limiting Issues:

```
PROCEDURE iterativeDeepening( s0, maxdepth
FOR d := 0 TO maxdepth DO
  besta := actionSearch( s0, d )
RETURN besta
```

Heuristic
Minimax w/ $\alpha\beta$ pruning.

maxdepth ?



15:00
time-limit

- Timeleft — the amount of time left for our program to play
- getTime() — returns the current time.

given time left, how much to allocate to the current turn?

≈ 40 moves.

2% of timeleft.

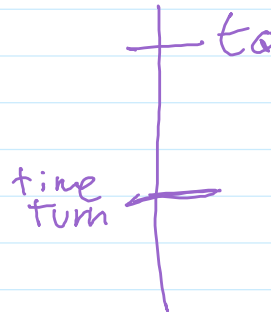
• Naïve Time Limit

```
PROCEDURE timeLimitedID( s0, timeleft ) (* over-time *)
```

```
  t0 := getTime()
  timeTurn := t0 + (timeleft * 0.022)
```

```
  d := 0
  REPEAT
    besta := actionSearch( s0, d )
```

```
    d++
    t1 := getTime()
  UNTIL t1 >= timeTurn
  RETURN besta
```



• Another option

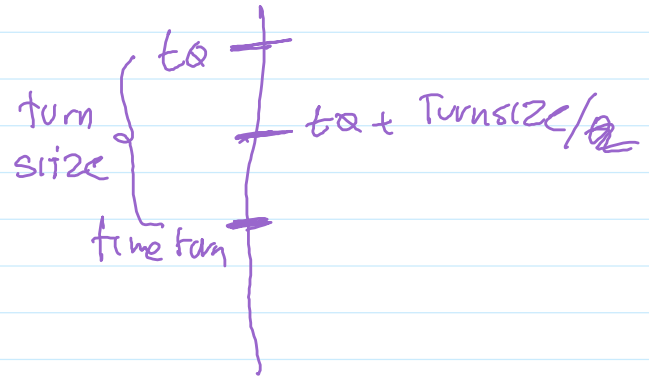
```
PROCEDURE timeLimitedID( s0, timeleft ) (* under-time *)
```

```
  t0 := getTime()
```

PROCEDURE timeLimitedID(s0, timeleft) (* under-time *)

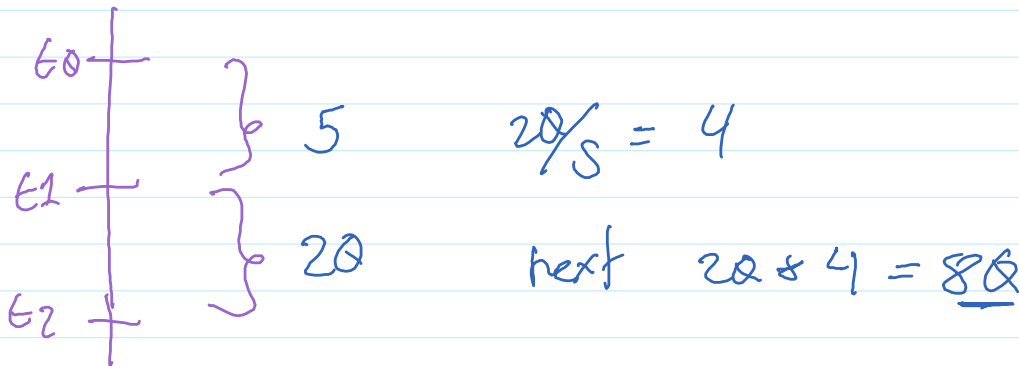
```
t0 := getTime()
turnSize := timeleft * 0.022
timeTurn := t0 + turnSize
```

```
d := 0
REPEAT
  besta := actionSearch( s0, d )
  d++
  t1 := getTime()
  UNTIL t1 > t0 + ( turnSize * 0.50 )
RETURN besta;
```



• Predictive time limit

- try to predict how much time the next iteration will take and return if not enough time left.



PROCEDURE timeLimitedID(s0, timeleft) (* predictive *)

```
t0 := getTime()
timelimit := t0 + timeleft * 0.022
```

```
besta := actionSearch( s0, 0 )
t1 := getTime()
```

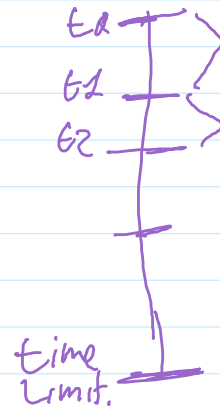
```
d := 1
REPEAT
  besta := actionSearch( s0, d )
  t2 := getTime()
```

```
timeratio := ( t2 - t1 ) / ( t1 - t0 )
timepredict := ( t2 - t1 ) * timeratio
```

```
d++
t0 := t1; t1 := t2
```

← book keeping time stamps.

```
UNTIL t2 + timepredict > timelimit
RETURN besta;
```



A small, handwritten green mark, possibly a checkmark or a stylized letter, located at the top left of the page.