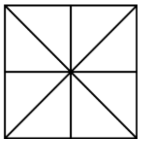
5CCS2INT Introduction to Artificial Intelligence

Coursework 2: Bagh–Chal - “The Tiger Game”

Name: Mateusz Nowak

Student Number: 1747415/K1763930

3x3 Board - Test Suite Description



The domain was tested on fifty-six problems on 3x3 Board – twenty-eight problems where goats won and twenty-eight where tigers won. The tests were separated into four subgroups – ranging from one to four tigers on board. The goat player has from five to eleven pawns to place on board in each subgroup. Each problem was run three times with no other user processes running on the computer. The shortest time was used in the analysis. *(On the right: 3x3 Bagh-Chal board)*

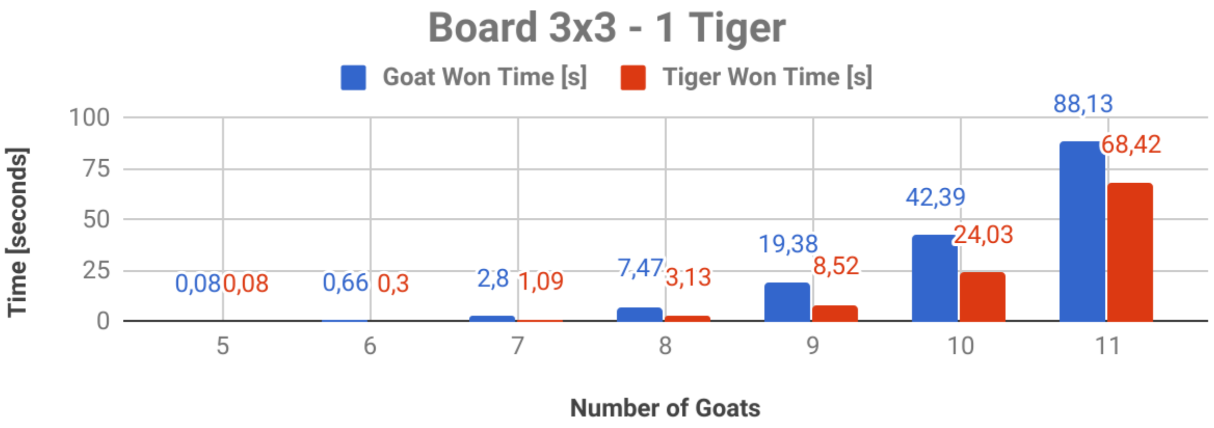
Planner used in tests: Metric-FF

CPU used in tests: Intel® Core™ i3-7100U CPU (2.40GHz × 4)

RAM used in tests: 8GB

First Subgroup – One Tiger

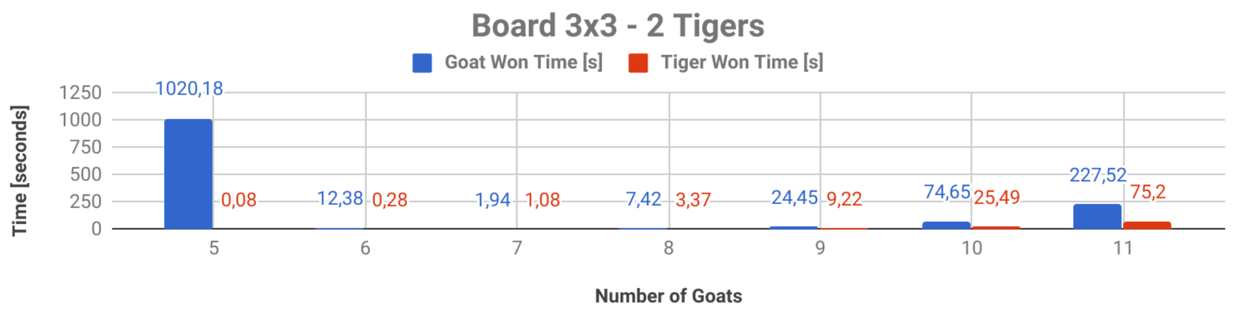
The subgroup represents the board states, where one tiger is placed in the bottom right corner of the board. The times for the planner to find the solution to each problem are shown below:



We can see that the time grows exponentially with the number of pawns in game. The planner takes more time to find a game scenario where goats win.

Second Subgroup – Two Tigers

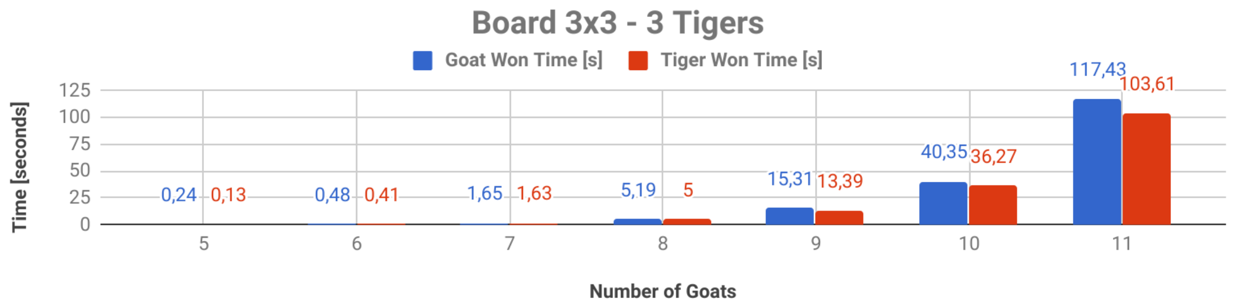
The subgroup represents the board states, where one tiger is placed in the top left corner of the board and one is placed in the bottom right corner of the board. In this suite, we can see an anomalous case. When the goat player has only five pawns to use and there are two tigers on the board, Enforced-Hill Climbing algorithm fails and the planner uses Best-First Search Algorithm. In this case, the planner takes around 17 minutes to find the game scenario where goat player wins. The times for the planner to find the solution to each problem are shown below:

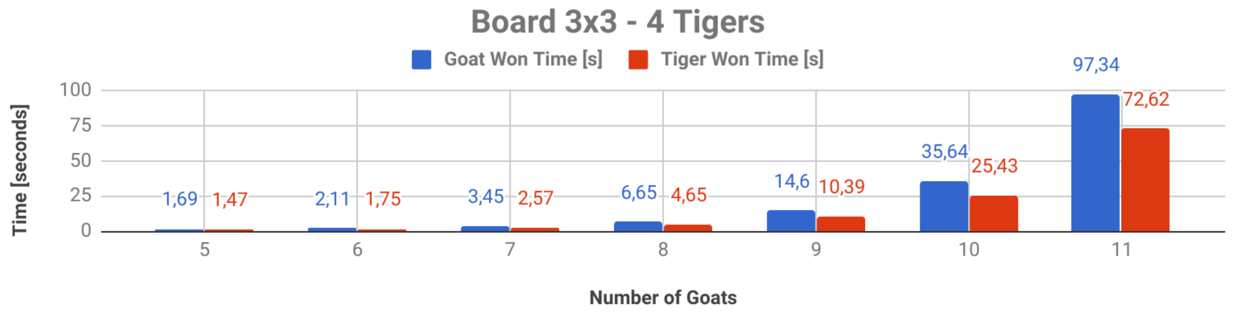


In this subgroup, we can see larger disproportion between goat player's times and tiger player's times compared to the previous test suite.

Third and Fourth Subgroup – Three and Four Tigers

The third subgroup represents the board states, where one tiger is placed in the top left corner of the board, one is place in top right corner and one is placed in the bottom right corner of the board. The fourth test suite represents the board states, where each tiger is in each corner. The times for the planner to find the solutions are shown below:

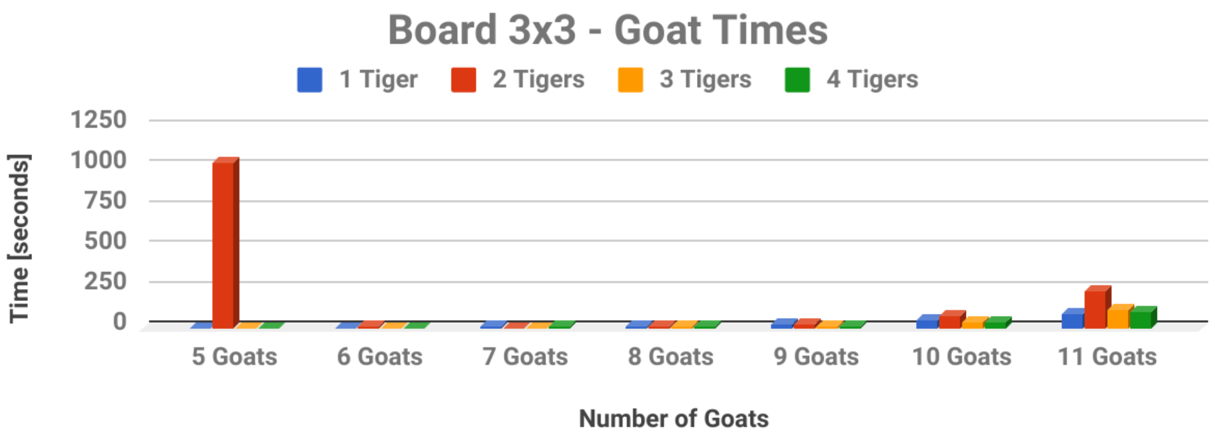


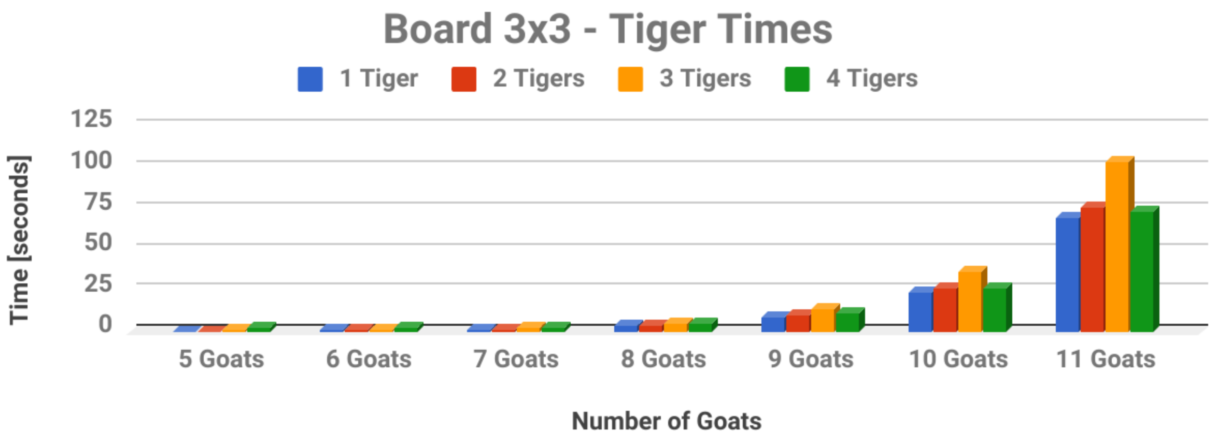


The times in subgroup three are higher for more complex problems (more than eight goats). The times follow the same pattern from the first subgroup.

3x3 Board Test Suite – Final Comparisons

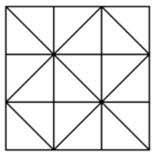
The winning condition is much more complex for the goat player compared to tiger player, which can be seen in the times taken by the planner to solve the problem. Despite the anomalous case in second test suite, the times in each subgroup follow an exponential pattern. Overall, the problems in the second and third test suite take much more time compared to the first and fourth subgroup.





4x4 Board Test Suite

Due to extensive amount of time taken by the planner, only some problems were possible to run for the goat player on 4x4 board. These problems took more than an hour to solve with the highest time measured equal to twelve hours and thirty minutes and one problem killed after forty-two hours of run time. The test suite follows the same pattern as the 3x3 board test suite. The results are shown in the appendix of this report. *(On the right: 4x4 Bagh-Chal Board)*



Conclusion

The planner fails to outperform the human player – it takes too much time to play the game on bigger boards and, due to the planner limitations and game complexity, the planner rejects any games on 5x5 board – normal Bagh-Chal board. Nevertheless, the planner can be used to balance the game rules. The most balanced scenario would be the one with the equal time for goat player and tiger player to win (3 tigers and 7 goats in test suite three).

Metric-FF was also able to show the hardest game scenarios on 3x3 board for both players:

- Goat player – 5 Goats and 2 Tigers
- Tiger player – 11 Goats and 3 Tigers

APPENDIX

Contents:

Notice: If there are any problems with running the problem files – please close all programs before running the planner!

DOMAIN

```
(define (domain bagh-chal)

  (:requirements :adl)
  (:types pawn - object
        goat tiger - pawn
        location)
  (:predicates

    (occupied ?l - location)
    (connected ?l - location ?n - location)
    (jump ?from - location ?through - location ?to - location)

    (atlocation ?p - pawn ?l - location)
    (taken ?g - goat)
    (placed ?g - goat)

    (goatWon)
    (tigerWon)
    (goatMove)

  )

  (:functions
 (numberOfTakenGoats)
)

(:action place
  :parameters (?g - goat ?l - location)
  :precondition (and
    (not (placed ?g))
    (not (occupied ?l))
    (not (taken ?g))
    (goatMove)
  )
  :effect (and
    (atlocation ?g ?l)
    (not (goatMove))
    (placed ?g)
    (occupied ?l)
  )
)

(:action move-Goat
  :parameters (?g - goat ?from - location ?to - location)
  :precondition (and
    (forall (?goat - goat)
        (or (placed ?goat)
          (taken ?goat)
        )
    )
    (atlocation ?g ?from)
    (not (taken ?g))
    (or
      (connected ?from ?to)
      (connected ?to ?from)
    )
    (not (occupied ?to))
    (goatMove)
  )
  :effect (and
    (atlocation ?g ?to)
    (occupied ?to)
    (not (atlocation ?g ?from))
    (not (occupied ?from))
    (not (goatMove))
  ))

(:action move-Tiger
  :parameters (?t - tiger ?from - location ?to - location)
  :precondition (and
    (atlocation ?t ?from)
    (or
      (connected ?from ?to)
      (connected ?to ?from)
    )
    (not (occupied ?to))
    (not (goatMove))
  )
  :effect (and
    (not (atlocation ?t ?from))
    (not (occupied ?from))
    (atlocation ?t ?to)
    (occupied ?to)
    (goatMove)
  )
)

(:action jump
  :parameters (?t - tiger ?from - location ?to - location)
  :vars (?goat - goat ?middle - location)
  :precondition (and
    (atlocation ?t ?from)
    (not (occupied ?to))
    (atlocation ?goat ?middle)
    (or (jump ?from ?middle ?to)
      (jump ?to ?middle ?from))
    (not (goatMove))
  )
  :effect (and
      (atlocation ?t ?to)
      (occupied ?to)
      (not (atlocation ?t ?from))
      (not (occupied ?from))
      (not (atlocation ?goat ?middle))
      (not (occupied ?middle))
      (taken ?goat)
      (not (placed ?goat))
      (goatMove)
      (increase (numberOfTakenGoats) 1)
   )
)

(:action win-Goats
   :precondition
   (and
    (not (tigerWon))
    (forall (?tiger - tiger)
     (exists (?blocked - location)
       (and  (atlocation ?tiger ?blocked)
          (forall (?dest - location ?middle - location)
               (imply (or (connected ?blocked ?dest)
                    (connected ?dest ?blocked)
                    (jump ?blocked ?middle ?dest)
                    (jump ?dest ?middle ?blocked)
                    )
                  (occupied ?dest))
          )
       )
    )
    )
   )
   :effect (goatWon)
  )

  (:action win-Tigers
   :precondition
   (and
    (not (goatWon))
    (>= (numberOfTakenGoats) 5)
   )
   :effect (tigerWon)
  )
)
```

PROBLEM – BOARD 3x3; 6 GOATS; 2 TIGERS; GOATWON

```
(define (problem game3x3-G-2T6G)
 (:domain bagh-chal)
 (:objects
     tiger1 tiger2 - tiger
     location1 location2 location3 location4 location5 location6 location7 location8 location9 - location
     goat1 goat2 goat3 goat4 goat5 goat6 - goat
 )

 (:init
  (goatMove)
  (= (numberOfTakenGoats) 0)

  (occupied location9)
  (occupied location1)
  (atlocation tiger1 location9)
  (atlocation tiger2 location1)

  (jump location1 location2 location3)
  (jump location1 location4 location7)
  (jump location1 location5 location9)
  (jump location2 location5 location8)
  (jump location3 location5 location7)
  (jump location4 location5 location6)
  (jump location3 location6 location9)
  (jump location7 location8 location9)

  (connected location1 location4)
  (connected location4 location7)
  (connected location7 location8)
  (connected location8 location9)
  (connected location9 location6)
  (connected location6 location3)
  (connected location3 location2)
  (connected location2 location1)

  (connected location1 location5)
  (connected location3 location5)
  (connected location2 location5)
  (connected location4 location5)
  (connected location6 location5)
  (connected location7 location5)
  (connected location8 location5)
  (connected location9 location5)
 )
 (:goal
   (goatWon)
 )
 (:metric minimize (numberOfTakenGoats))
)
```

PROBLEM – BOARD 3x3; 10 GOATS; 3 TIGERS; TIGERWON

```
(define (problem game3x3-T-T3G10)
 (:domain bagh-chal)
 (:objects
     tiger1 tiger2 tiger3 - tiger
     location1 location2 location3 location4 location5 location6 location7 location8 location9 - location
     goat1 goat2 goat3 goat4 goat5 goat6 goat7 goat8 goat9 goat10 - goat
 )
 (:init
  (goatMove)
  (= (numberOfTakenGoats) 0)

  (occupied location9)
  (occupied location1)
  (occupied location3)
  (atlocation tiger1 location9)
  (atlocation tiger3 location3)
  (atlocation tiger2 location1)

  (jump location1 location2 location3)
  (jump location1 location4 location7)
  (jump location1 location5 location9)
  (jump location2 location5 location8)
  (jump location3 location5 location7)
  (jump location4 location5 location6)
  (jump location3 location6 location9)
  (jump location7 location8 location9)

  (connected location1 location4)
  (connected location4 location7)
  (connected location7 location8)
  (connected location8 location9)
  (connected location9 location6)
  (connected location6 location3)
  (connected location3 location2)
  (connected location2 location1)

  (connected location1 location5)
  (connected location3 location5)
  (connected location2 location5)
  (connected location4 location5)
  (connected location6 location5)
  (connected location7 location5)
  (connected location8 location5)
  (connected location9 location5)
 )
 (:goal
   (tigerWon)
 )
 (:metric minimize (numberOfTakenGoats))
)
```

PROBLEM – BOARD 3x3; 11 GOATS; 4 TIGERS; GOATWON

```
(define (problem game3x3-G-T4G11)
(:domain bagh-chal)
(:objects
    tiger1 tiger2 tiger3 tiger4 - tiger
    location1 location2 location3 location4 location5 location6 location7 location8 location9 - location
    goat1 goat2 goat3 goat4 goat5 goat6 goat7 goat8 goat9 goat10 goat11 - goat
)

(:init
  (goatMove)
  (= (numberOfTakenGoats) 0)

  (occupied location9)
  (occupied location1)
  (occupied location3)
  (occupied location7)
  (atlocation tiger1 location9)
  (atlocation tiger2 location1)
  (atlocation tiger3 location3)
  (atlocation tiger4 location7)

  (jump location1 location2 location3)
  (jump location1 location4 location7)
  (jump location1 location5 location9)
  (jump location2 location5 location8)
  (jump location3 location5 location7)
  (jump location4 location5 location6)
  (jump location3 location6 location9)
  (jump location7 location8 location9)

  (connected location1 location4)
  (connected location4 location7)
  (connected location7 location8)
  (connected location8 location9)
  (connected location9 location6)
  (connected location6 location3)
  (connected location3 location2)
  (connected location2 location1)

  (connected location1 location5)
  (connected location3 location5)
  (connected location2 location5)
  (connected location4 location5)
  (connected location6 location5)
  (connected location7 location5)
  (connected location8 location5)
  (connected location9 location5)
 )
 (:goal
   (goatWon)
 )
 (:metric minimize (numberOfTakenGoats))
)
```

PROBLEM – BOARD 4x4; 8 GOATS; 4 TIGERS; GOATWON (LONGEST COMPLETED PROBLEM)

```
(define (problem board4x4-G-4T8G)
 (:domain bagh-chal)
 (:objects
    tiger1 tiger2 tiger3 tiger4 - tiger
     l1 l2 l3 l4 l5 l5 l6 l7 l8 l9 l10 l11 l12 l13 l14 l15 l16 - location
     g1 g2 g3 g4 g5 g6 g7 g8 - goat
 )

 (:init
  (goatMove)
  (= (numberOfTakenGoats) 0)

  (occupied l1)
  (occupied l4)
  (occupied l13)
  (occupied l16)
  (atlocation tiger1 l1)
  (atlocation tiger2 l4)
  (atlocation tiger3 l13)
  (atlocation tiger4 l16)

  (connected l1 l2)
  (connected l1 l5)
  (connected l1 l6)
  (jump l1 l2 l3)
  (jump l1 l5 l9)
  (jump l1 l6 l11)

  (connected l2 l6)
  (connected l2 l3)
  (jump l2 l3 l4)
  (jump l2 l6 l10)

  (connected l3 l6)
  (connected l3 l7)
  (connected l3 l8)
```

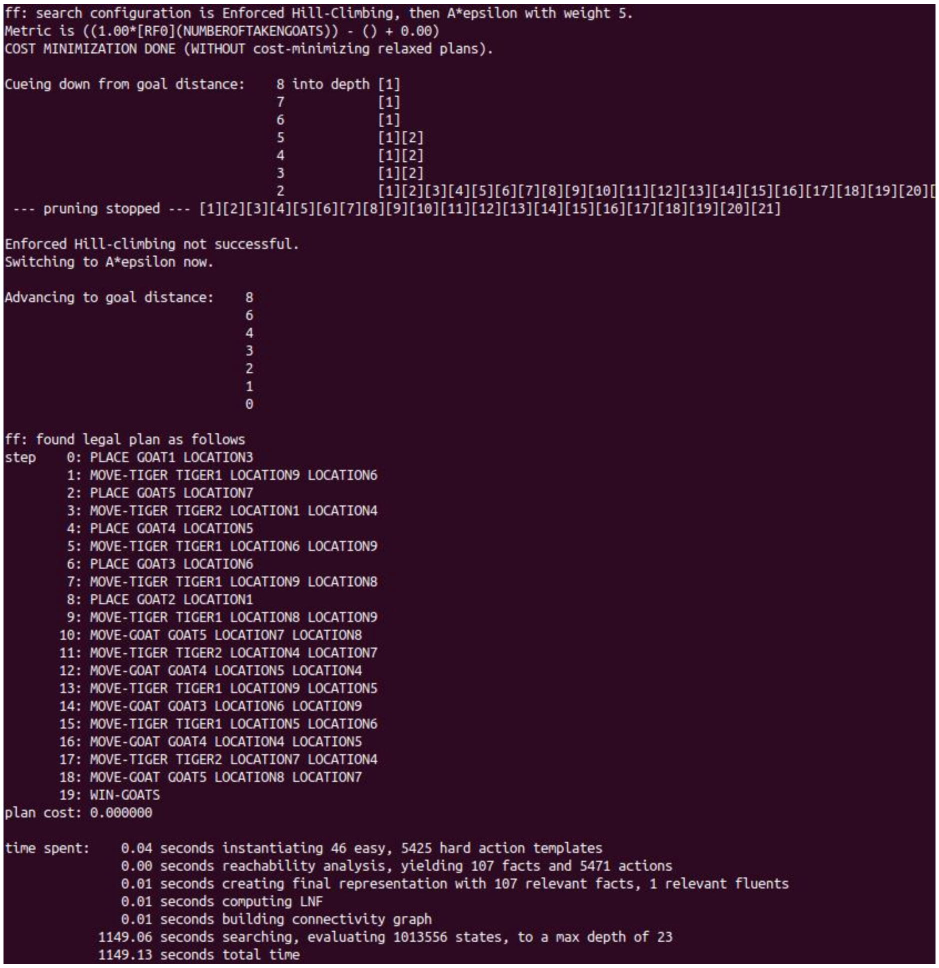
```
(connected l3 l4)
(jump l3 l6 l9)
(jump l3 l7 l11)

(connected l4 l8)
(jump l4 l8 l12)

(connected l5 l6)
(connected l5 l9)
(jump l5 l9 l13)
(jump l5 l6 l7)

(connected l6 l9)
(connected l6 l10)
(connected l6 l11)
(connected l6 l7)
(jump l6 l10 l14)
(jump l6 l11 l16)
(jump l6 l7 l8)

(connected l7 l8)
(connected l7 l11)
(jump l7 l11 l15)

(connected l8 l11)
(connected l8 l12)
(jump l8 l11 l14)
(jump l8 l12 l16)

(connected l9 l10)
(connected l9 l13)
(connected l9 l14)
(jump l9 l10 l11)

(connected l10 l11)
(connected l10 l14)
(jump l10 l11 l12)

(connected l11 l14)
(connected l11 l15)
(connected l11 l16)
(connected l11 l12)

(connected l12 l15)
```

```
  (connected l13 l14)
  (jump l13 l14 l15)
  (connected l14 l15)
  (jump l14 l15 l16)
  (connected l15 l16)
 )
 (:goal
   (goatWon)
 )
 (:metric minimize (numberOfTakenGoats))
)
```

LIMITED NUMBER OF RESULTS FOR BOARD 4x4 [Time in seconds]

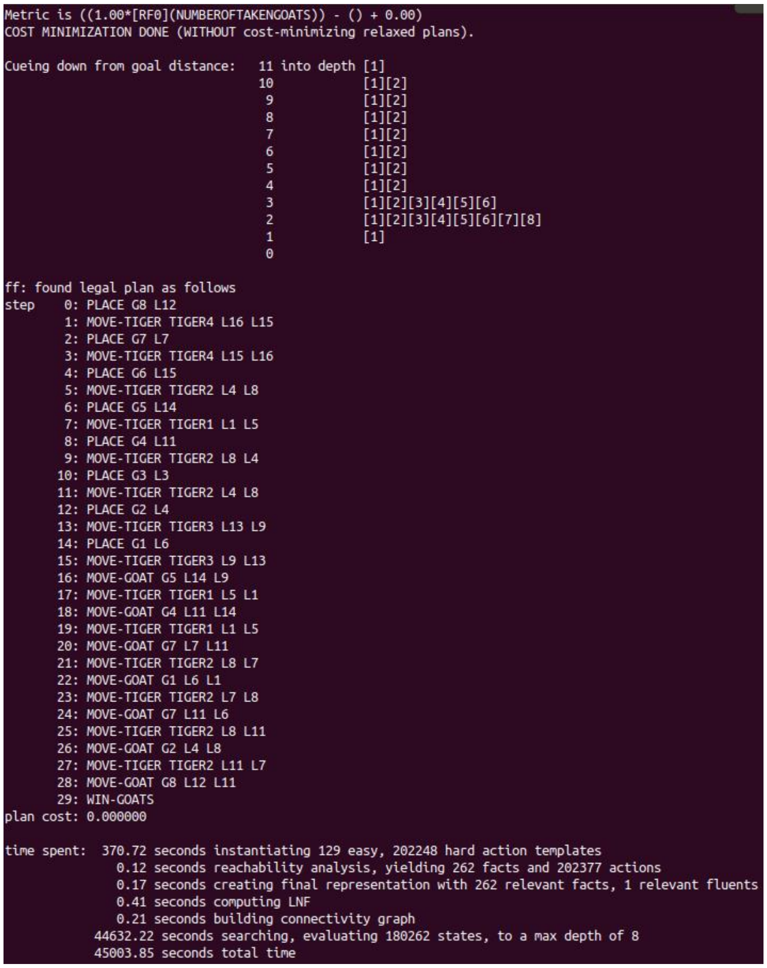
BOARD 4x4 - GOAT TIMES					
	Number of Tigers				
Number of Goats		1 Tiger	2 Tigers	3 Tigers	4 Tigers
	5 Goats	477,1			
	6 Goats	724,23			
	7 Goats	3,74	788,63		
	8 Goats	11,5	4304,11	>~153600	45003,85
	9 Goats	34,97	18335,78	>~86400	27380,55
	10 Goats	98,86			

DETAILED RESULT – BOARD 3x3; 5 GOATS; 2 TIGERS – GOATWON



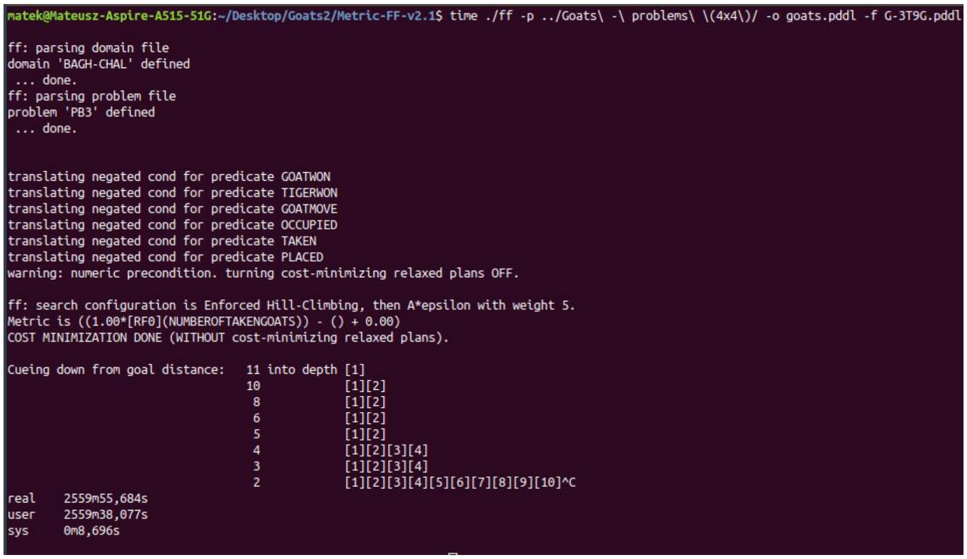
```
ff: search configuration is Enforced Hill-Climbing, then A*epsilon with weight 5.
Metric is ((1.00*[RF0](NUMBEROFTAKENGOATS)) - () + 0.00)
COST MINIMIZATION DONE (WITHOUT cost-minimizing relaxed plans).

Cueing down from goal distance:    8 into depth [1]
                                   7            [1]
                                   6            [1]
                                   5            [1][2]
                                   4            [1][2]
                                   3            [1][2]
                                   2            [1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][
 --- pruning stopped --- [1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21]

Enforced Hill-climbing not successful.
Switching to A*epsilon now.

Advancing to goal distance:    8
                               6
                               4
                               3
                               2
                               1
                               0

ff: found legal plan as follows
step    0: PLACE GOAT1 LOCATION3
        1: MOVE-TIGER TIGER1 LOCATION9 LOCATION6
        2: PLACE GOAT5 LOCATION7
        3: MOVE-TIGER TIGER2 LOCATION1 LOCATION4
        4: PLACE GOAT4 LOCATION5
        5: MOVE-TIGER TIGER1 LOCATION6 LOCATION9
        6: PLACE GOAT3 LOCATION6
        7: MOVE-TIGER TIGER1 LOCATION9 LOCATION8
        8: PLACE GOAT2 LOCATION1
        9: MOVE-TIGER TIGER1 LOCATION8 LOCATION9
       10: MOVE-GOAT GOAT5 LOCATION7 LOCATION8
       11: MOVE-TIGER TIGER2 LOCATION4 LOCATION7
       12: MOVE-GOAT GOAT4 LOCATION5 LOCATION4
       13: MOVE-TIGER TIGER1 LOCATION9 LOCATION5
       14: MOVE-GOAT GOAT3 LOCATION6 LOCATION9
       15: MOVE-TIGER TIGER1 LOCATION5 LOCATION6
       16: MOVE-GOAT GOAT4 LOCATION4 LOCATION5
       17: MOVE-TIGER TIGER2 LOCATION7 LOCATION4
       18: MOVE-GOAT GOAT5 LOCATION8 LOCATION7
       19: WIN-GOATS
plan cost: 0.000000

time spent:    0.04 seconds instantiating 46 easy, 5425 hard action templates
               0.00 seconds reachability analysis, yielding 107 facts and 5471 actions
               0.01 seconds creating final representation with 107 relevant facts, 1 relevant fluents
               0.01 seconds computing LNF
               0.01 seconds building connectivity graph
            1149.06 seconds searching, evaluating 1013556 states, to a max depth of 23
            1149.13 seconds total time
```

DETAILED RESULT – BOARD 4x4; 8 GOATS; 4 TIGERS – GOATWON



```
Metric is ((1.00*[RF0](NUMBEROFTAKENGOATS)) - () + 0.00)
COST MINIMIZATION DONE (WITHOUT cost-minimizing relaxed plans).

Cueing down from goal distance:   11 into depth [1]
                                  10            [1][2]
                                   9            [1][2]
                                   8            [1][2]
                                   7            [1][2]
                                   6            [1][2]
                                   5            [1][2]
                                   4            [1][2]
                                   3            [1][2][3][4][5][6]
                                   2            [1][2][3][4][5][6][7][8]
                                   1            [1]
                                   0

ff: found legal plan as follows
step    0: PLACE G8 L12
        1: MOVE-TIGER TIGER4 L16 L15
        2: PLACE G7 L7
        3: MOVE-TIGER TIGER4 L15 L16
        4: PLACE G6 L15
        5: MOVE-TIGER TIGER2 L4 L8
        6: PLACE G5 L14
        7: MOVE-TIGER TIGER1 L1 L5
        8: PLACE G4 L11
        9: MOVE-TIGER TIGER2 L8 L4
       10: PLACE G3 L3
       11: MOVE-TIGER TIGER2 L4 L8
       12: PLACE G2 L4
       13: MOVE-TIGER TIGER3 L13 L9
       14: PLACE G1 L6
       15: MOVE-TIGER TIGER3 L9 L13
       16: MOVE-GOAT G5 L14 L9
       17: MOVE-TIGER TIGER1 L5 L1
       18: MOVE-GOAT G4 L11 L14
       19: MOVE-TIGER TIGER1 L1 L5
       20: MOVE-GOAT G7 L7 L11
       21: MOVE-TIGER TIGER2 L8 L7
       22: MOVE-GOAT G1 L6 L1
       23: MOVE-TIGER TIGER2 L7 L8
       24: MOVE-GOAT G7 L11 L6
       25: MOVE-TIGER TIGER2 L8 L11
       26: MOVE-GOAT G2 L4 L8
       27: MOVE-TIGER TIGER2 L11 L7
       28: MOVE-GOAT G8 L12 L11
       29: WIN-GOATS
plan cost: 0.000000

time spent:  370.72 seconds instantiating 129 easy, 202248 hard action templates
               0.12 seconds reachability analysis, yielding 262 facts and 202377 actions
               0.17 seconds creating final representation with 262 relevant facts, 1 relevant fluents
               0.41 seconds computing LNF
               0.21 seconds building connectivity graph
           44632.22 seconds searching, evaluating 180262 states, to a max depth of 8
           45003.85 seconds total time
```

42 HOURS UNFINISHED PROBLEM CASE



```
matek@Mateusz-Aspire-A515-51G:~/Desktop/Goats2/Metric-FF-v2.1$ time ./ff -p ../Goats\ -\ problems\ \(4x4\)/ -o goats.pddl -f G-3T9G.pddl

ff: parsing domain file
domain 'BAGH-CHAL' defined
 ... done.
ff: parsing problem file
problem 'PB3' defined
 ... done.


translating negated cond for predicate GOATWON
translating negated cond for predicate TIGERWON
translating negated cond for predicate GOATMOVE
translating negated cond for predicate OCCUPIED
translating negated cond for predicate TAKEN
translating negated cond for predicate PLACED
warning: numeric precondition. turning cost-minimizing relaxed plans OFF.

ff: search configuration is Enforced Hill-Climbing, then A*epsilon with weight 5.
Metric is ((1.00*[RF0](NUMBEROFTAKENGOATS)) - () + 0.00)
COST MINIMIZATION DONE (WITHOUT cost-minimizing relaxed plans).

Cueing down from goal distance:   11 into depth [1]
                                  10            [1][2]
                                   8            [1][2]
                                   6            [1][2]
                                   5            [1][2]
                                   4            [1][2][3][4]
                                   3            [1][2][3][4]
                                   2            [1][2][3][4][5][6][7][8][9][10]^C
real    2559m55,684s
user    2559m38,077s
sys     0m8,696s
```