Iterative Budgeted Exponential Search

Malte Helmert[1], Tor Lattimore[2], Levi H.S. Lelis[3],
Laurent Orseau[2], Nathan Sturtevant[4]

[1]University of Basel, Switzerland
[2]DeepMind, UK
[3]Federal de Viçosa, Brazil
[4]University of Alberta, Canada

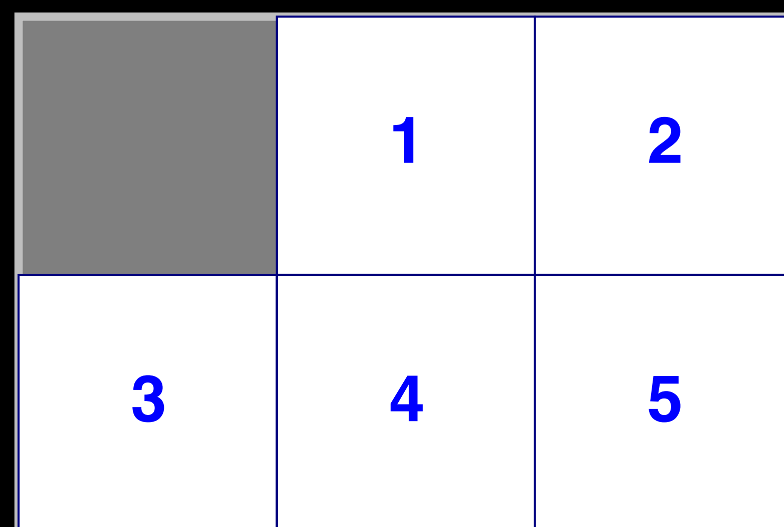
IJCAI 2019 - Macau, China

Talk jointly presented by Laurent Orseau and Nathan Sturtevant

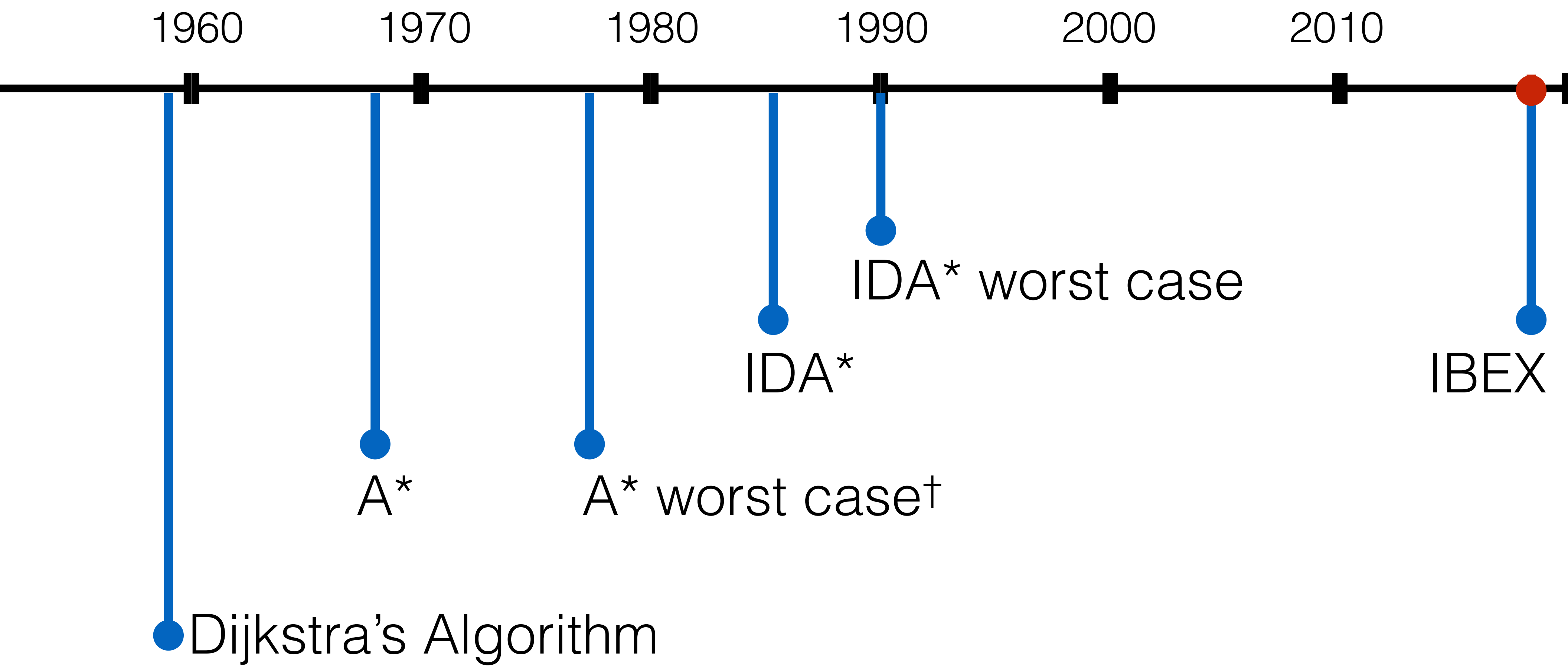
Start

h = 11

Goal



Shortest Path Algorithm History



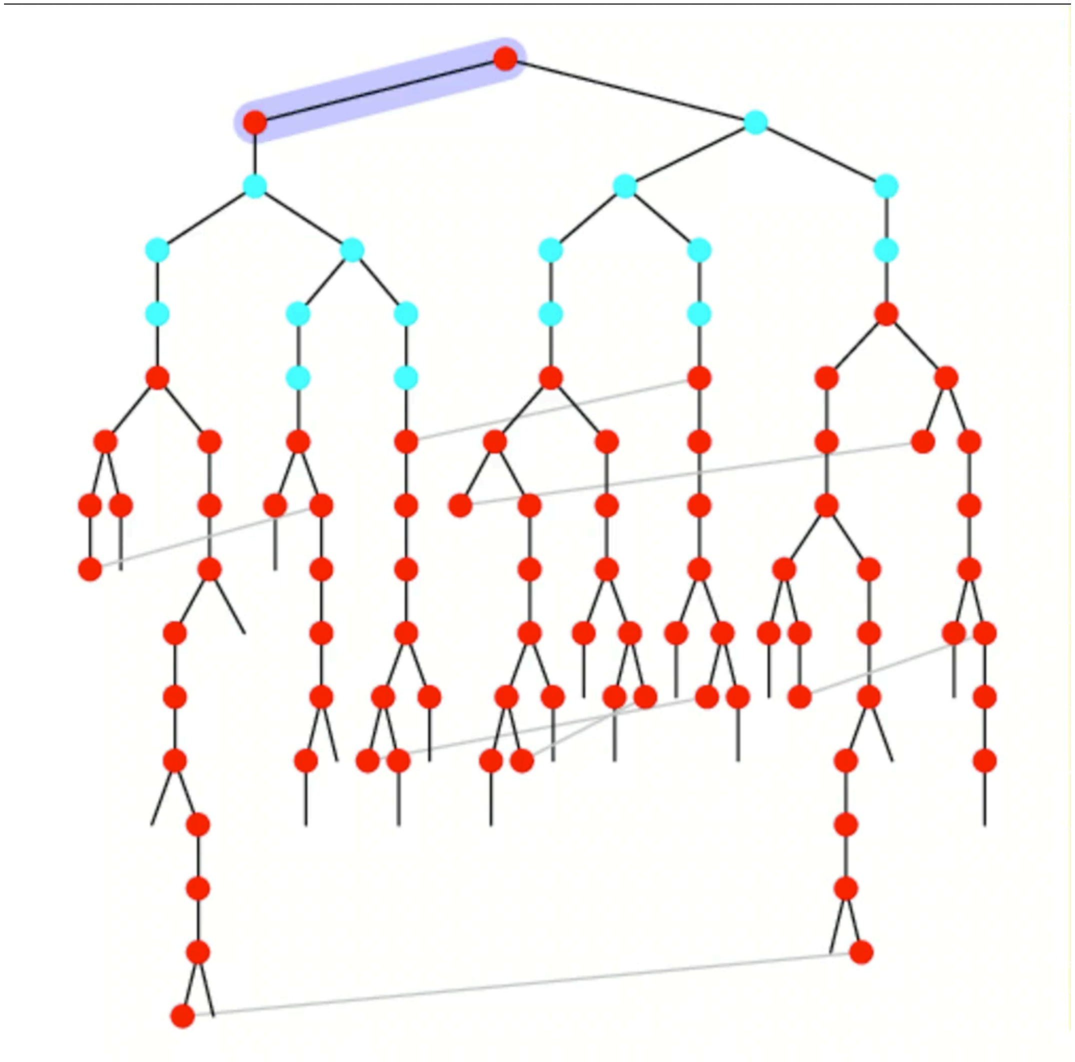
The N^2 problem

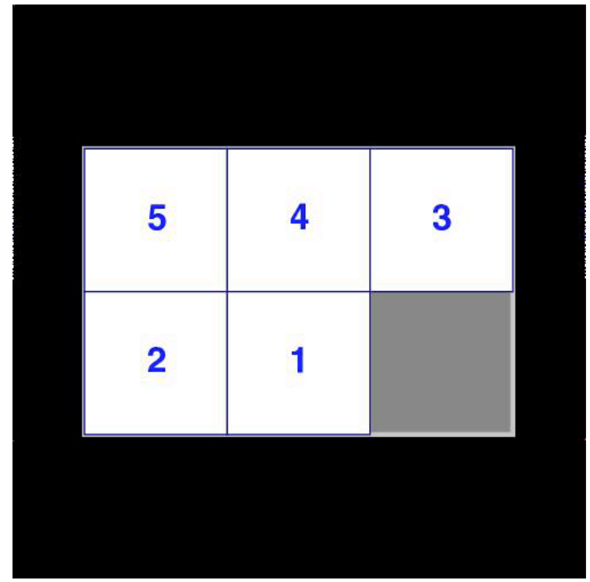
- Algorithms perform N^2 re-expansions of N states
- IDA*: N^2 expansions of $N = b^d$ states
- A*: 2^N expansions of N states
  - B & B': N^2 (Martelli 1977; Mero, 1984)
- Weighted A*: N^2 (Chen et al, 2019)

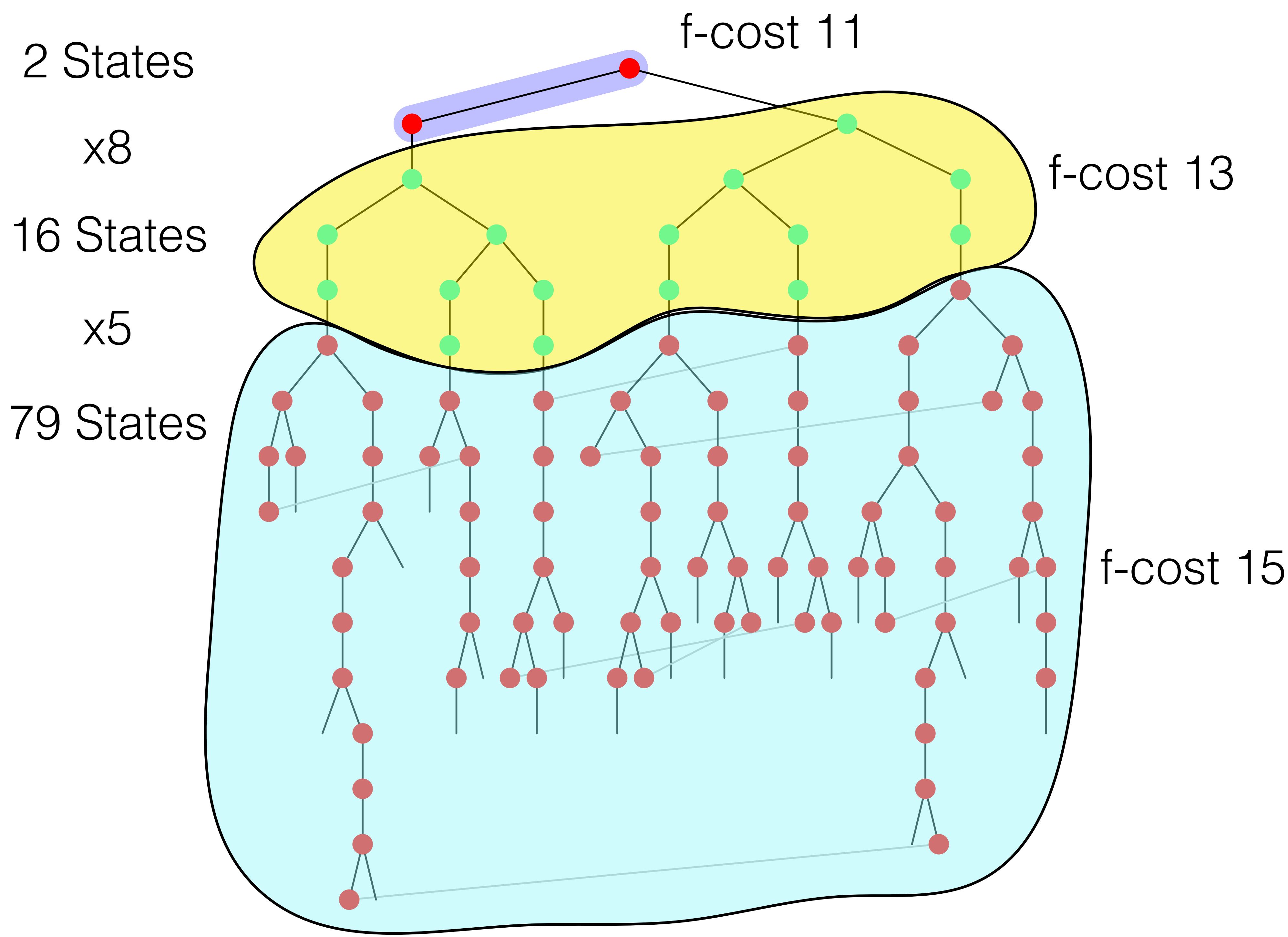
IDA* Refresher

- IDA* does *iterative deepening* search on *f*-costs
  - $f(n) = g(n) + h(n)$
- Next iteration *f*-cost:
  - Smallest unexplored from previous iteration
- If *f*-cost layers grow exponentially
  - IDA* is asymptotically optimal in node expansions

IDA* - Unit Costs



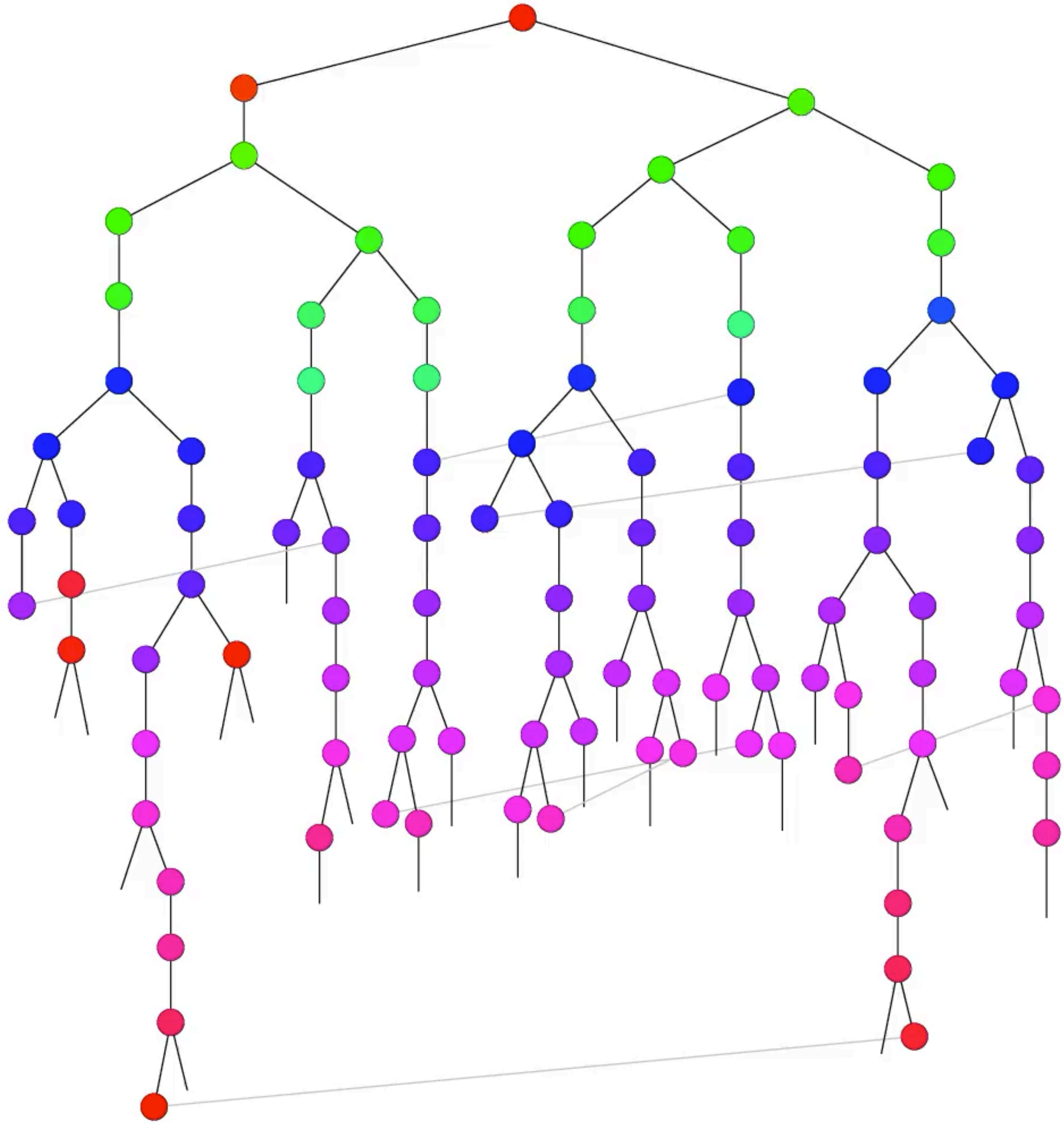


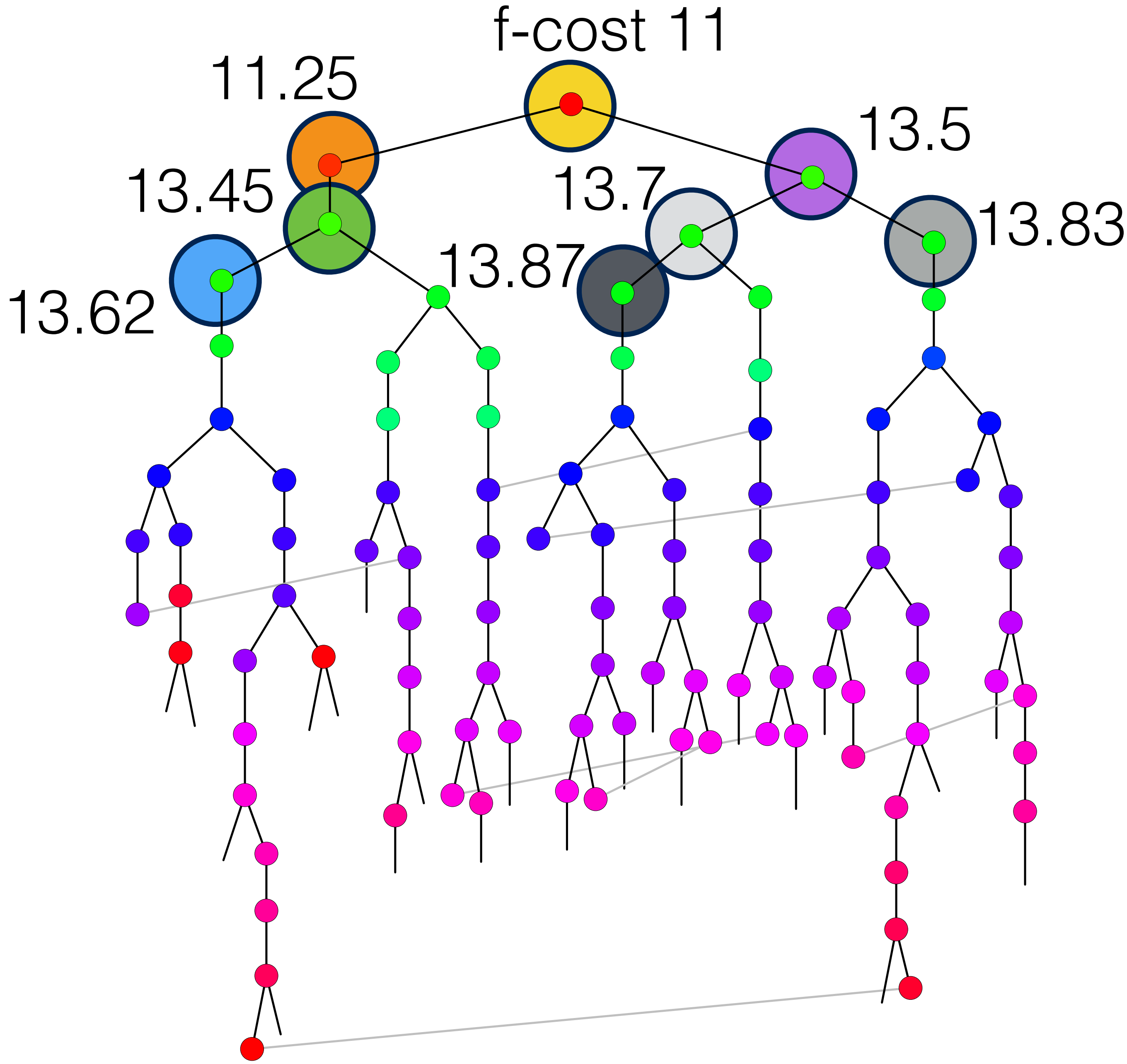


2 States
x8
16 States
x5
79 States
f-cost 11
f-cost 13
f-cost 15

IDA* Worst Case

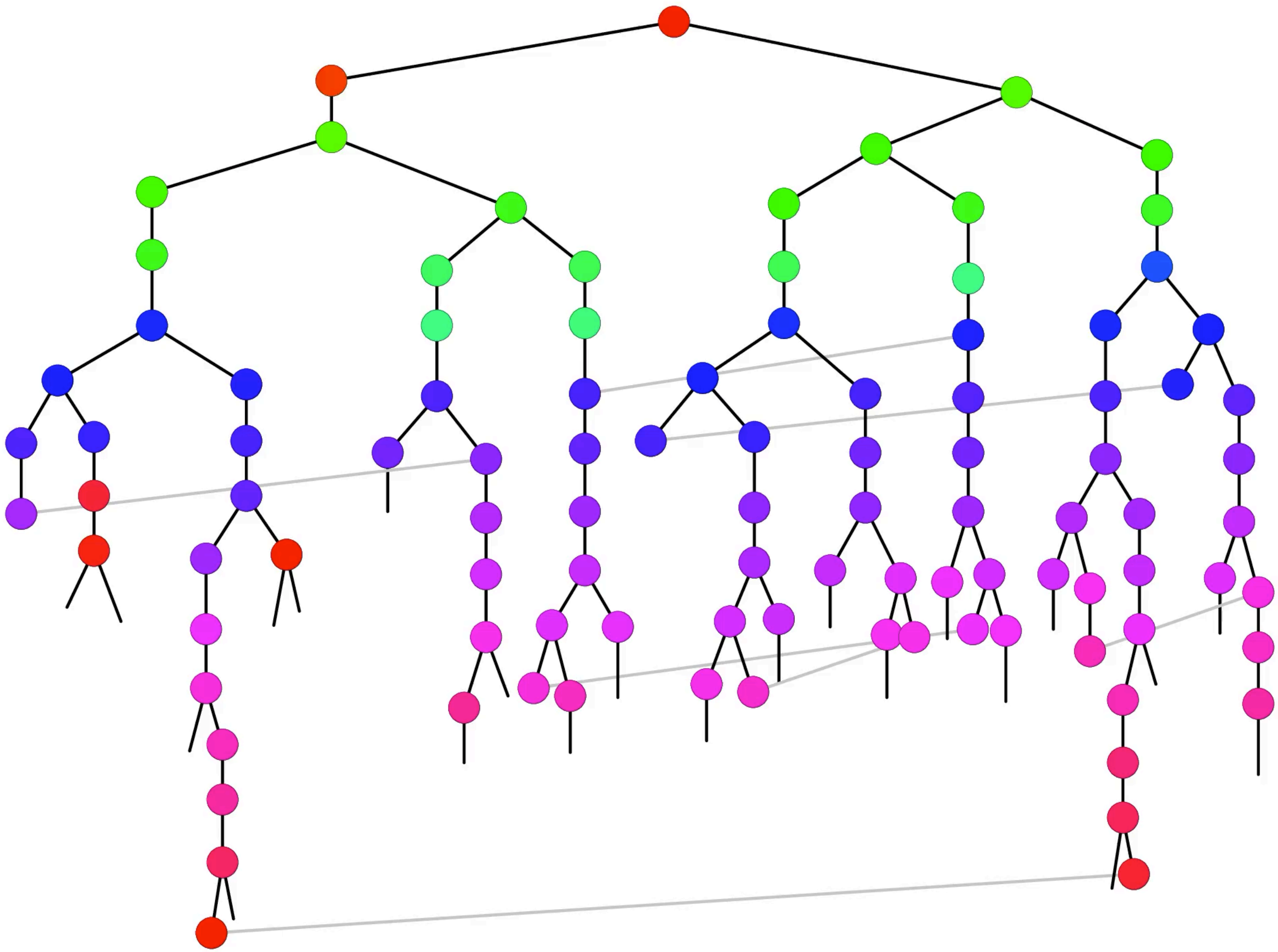
- *f*-cost layers grow exponentially
  - $1 + b + b^2 + b^3 + \ldots + b^d \approx b^d$
- What if *f*-cost layers grew linearly?
  - $1 + 2 + 3 + 4 + \ldots + b^d \approx (b^d)^2$
- Happens with non-unit edge costs:
  - STP: Cost of moving tile *t*: $\dfrac{t+2}{t+1}$




f-cost 11
11.25
13.5
13.45
13.7
13.83
13.87
13.62

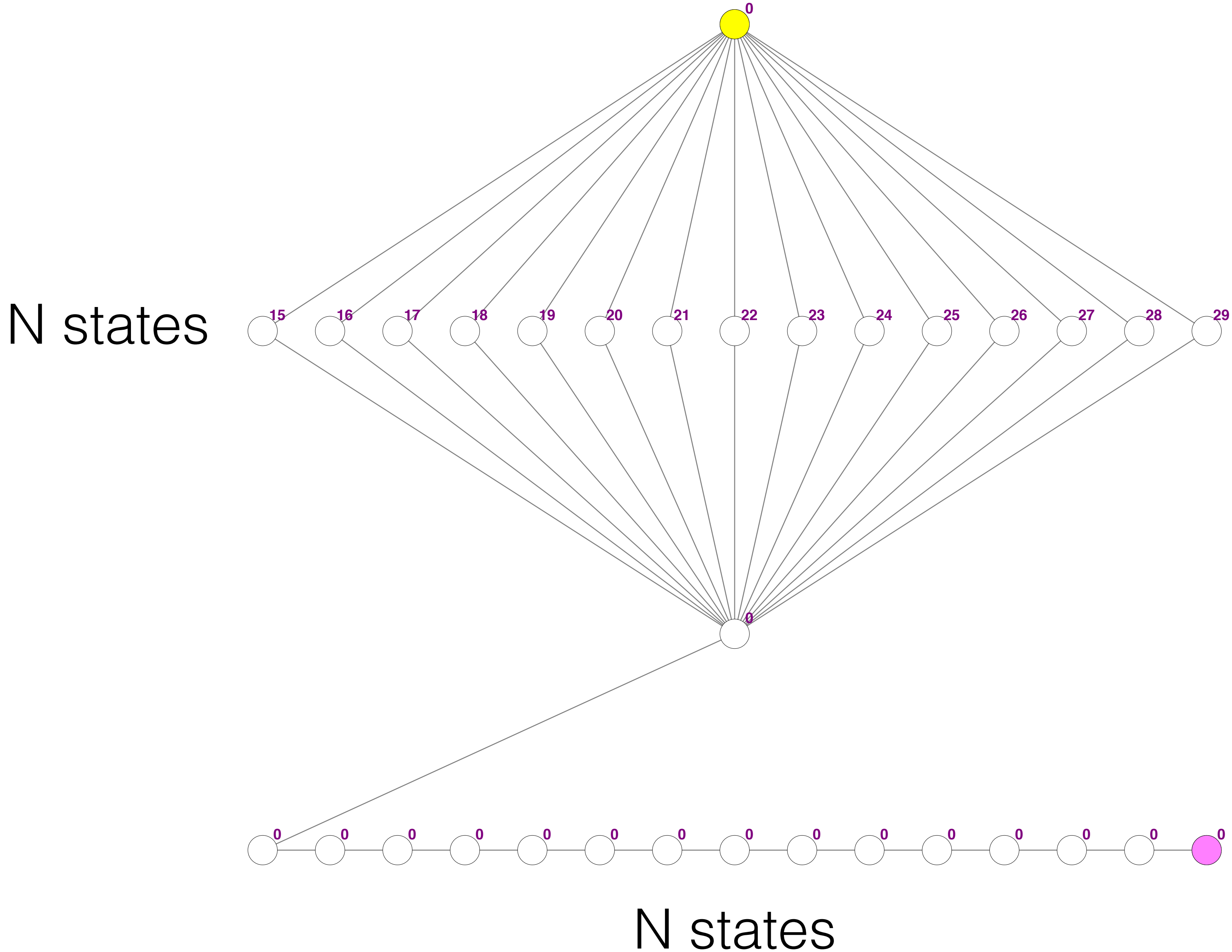
Previous Work

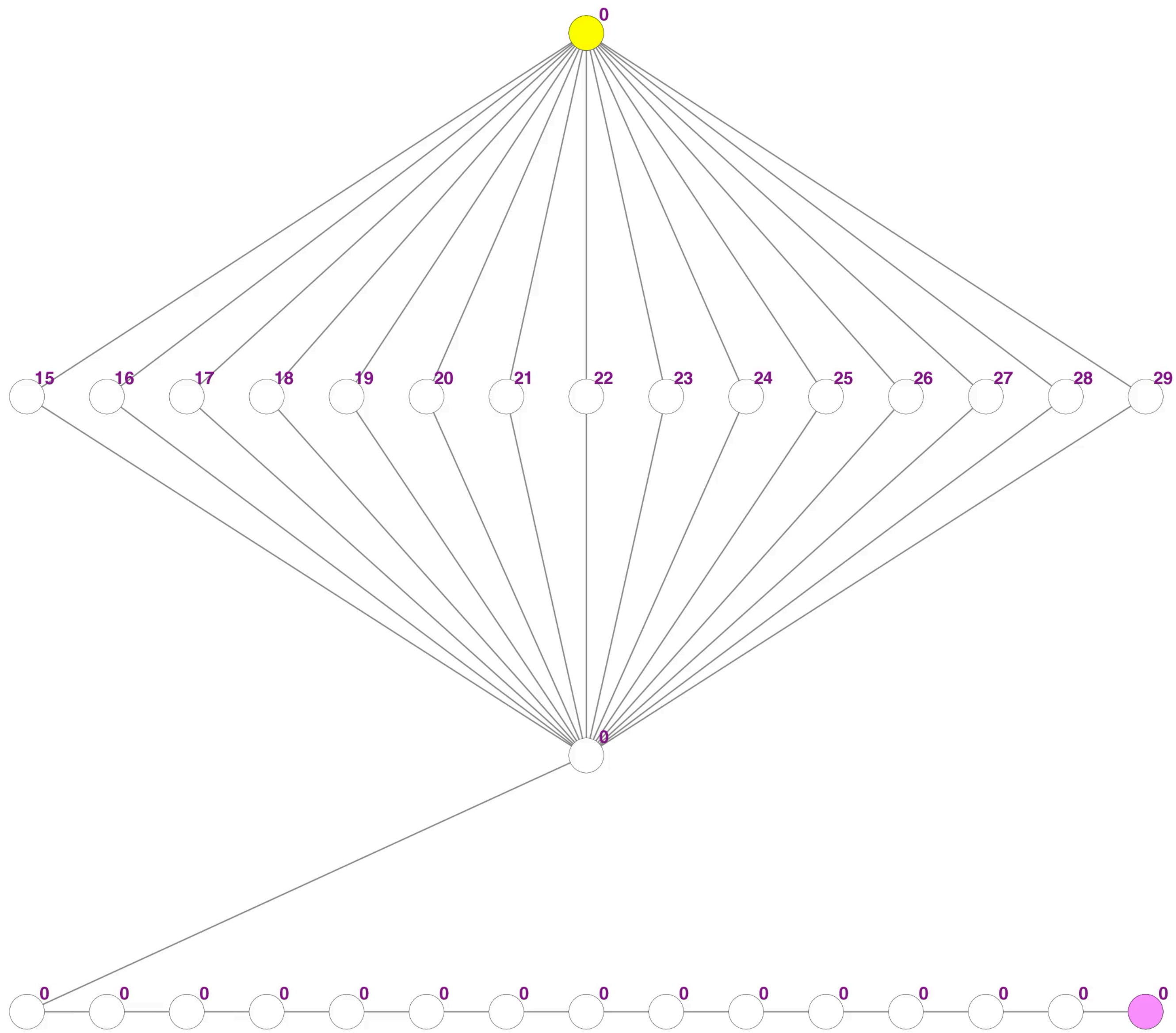
- Iterative Deepening
  - IDA* (Korf, 1985)
  - IDA^*_{CR} (Sarkar et al, 1990)
  - IDA^*_{IM} (Burns & Ruml, 2013)
  - EDA* (Sharon et al, 2014)



A* (B/B')

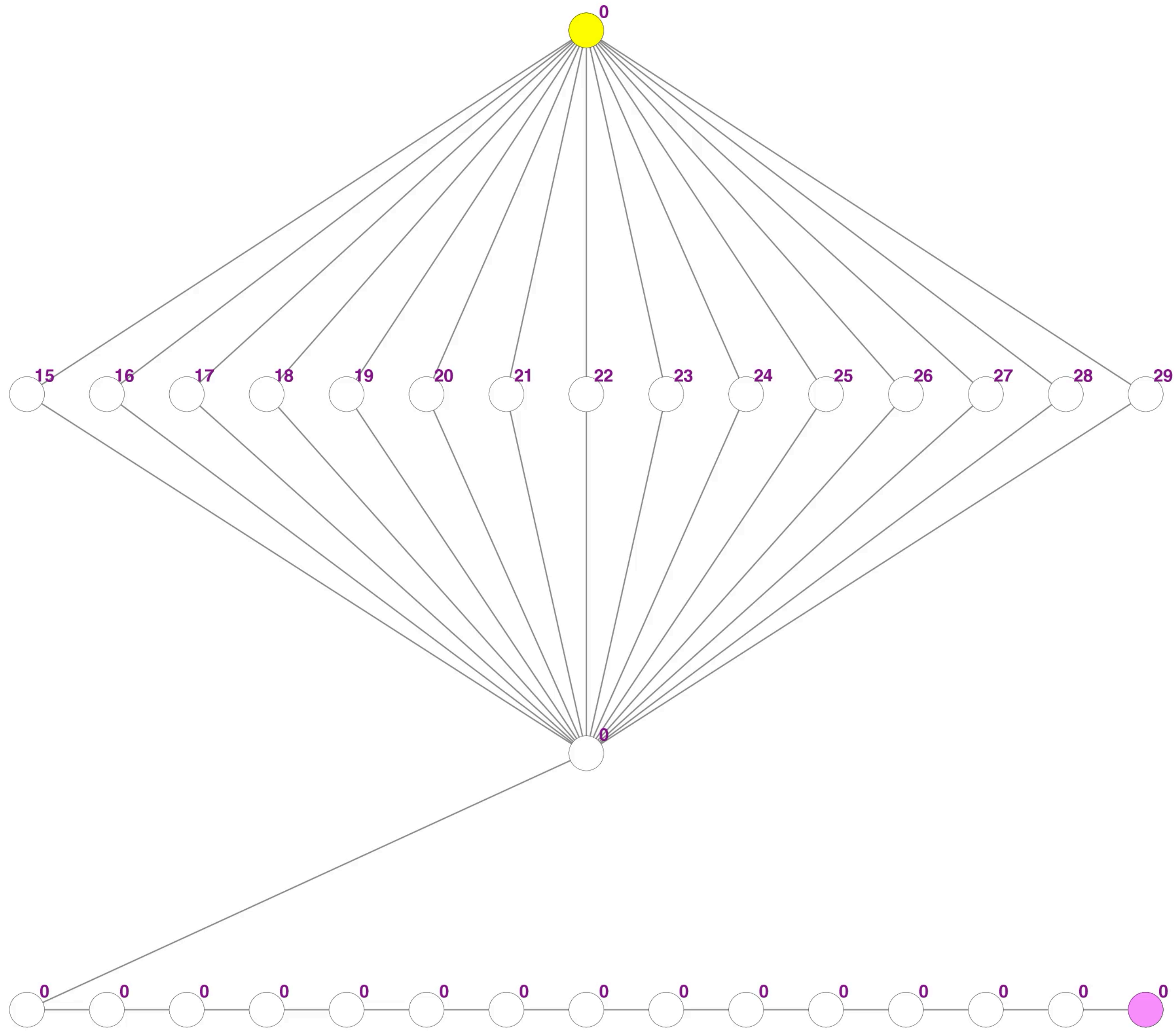
- A* with a consistent heuristic is optimal*
- With an inconsistent heuristic A* might do 2^N expansions of N states
- B and B' immediately propagate shorter paths
  - Will do at most N^2 expansions of N states


0
N states
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
N states


0
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0

Previous Work

- Inconsistent Heuristics
  - B/B’ (Martelli 1977; Mero, 1984)
  - Delay (Sturtevant et al, 2008)
  - BPMX (Felner et al, 2011)

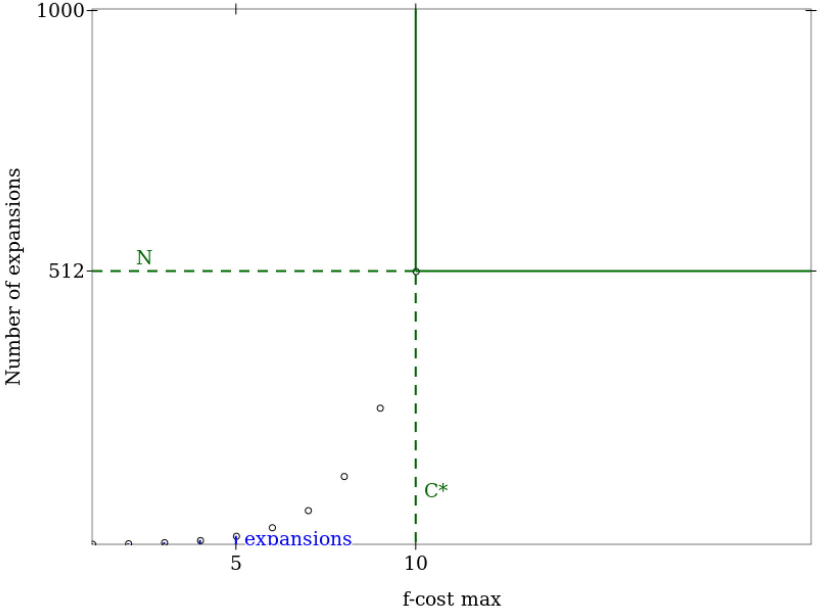

0
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0

Demos

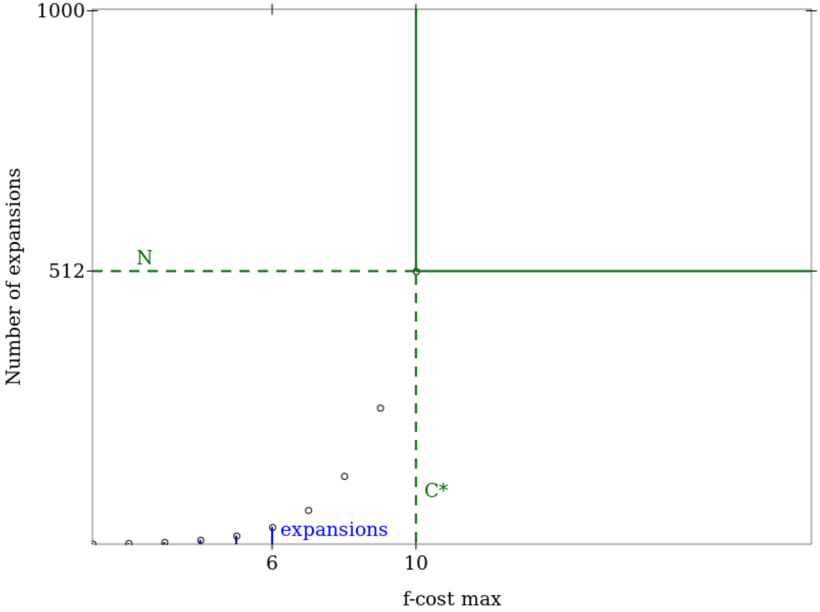
- All demos shown here runnable online:
  - https://www.movingai.com/SAS/BTS/

- Also running on my iPad; can give detailed demo

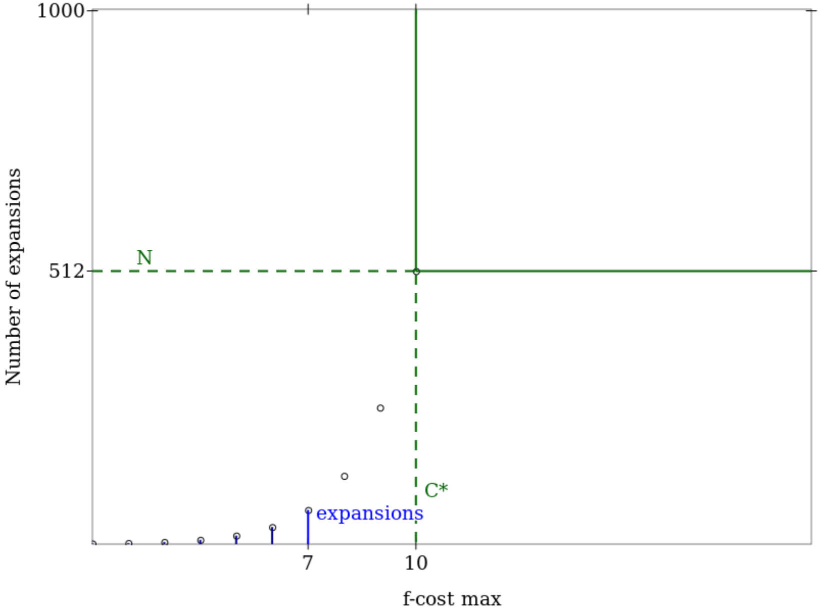
IDA* — Good Case



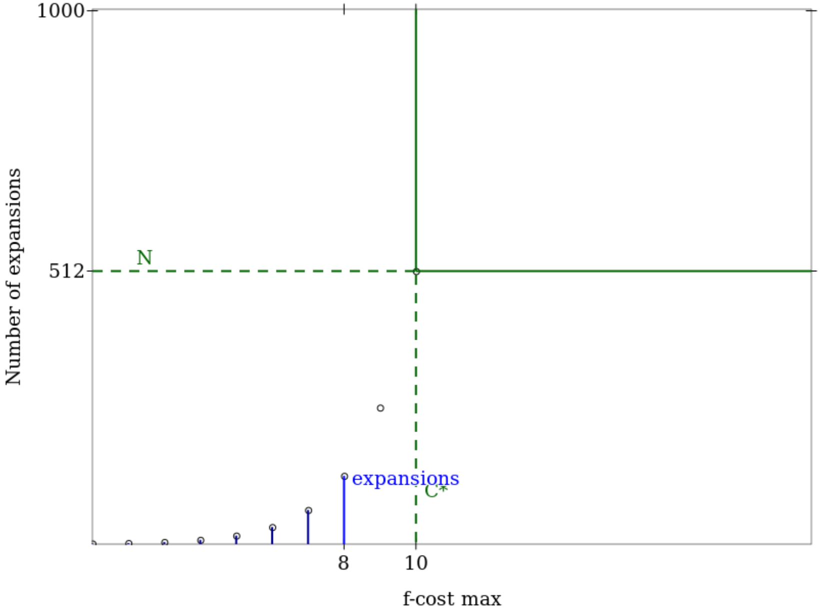
IDA* — Good Case



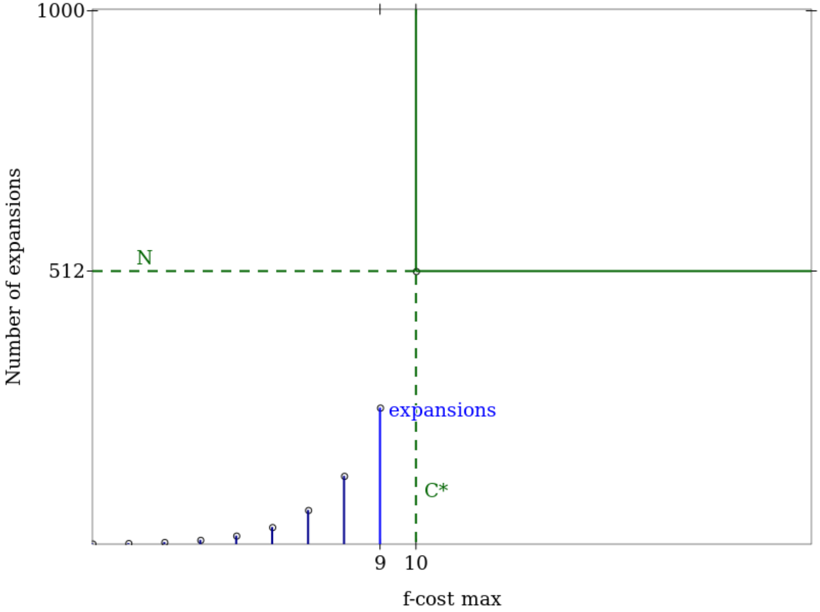
IDA* — Good Case



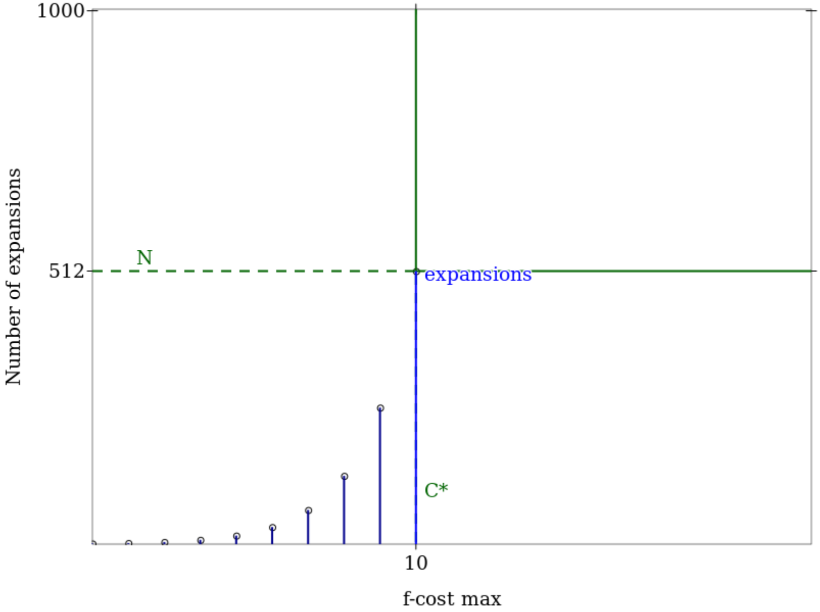
IDA* — Good Case



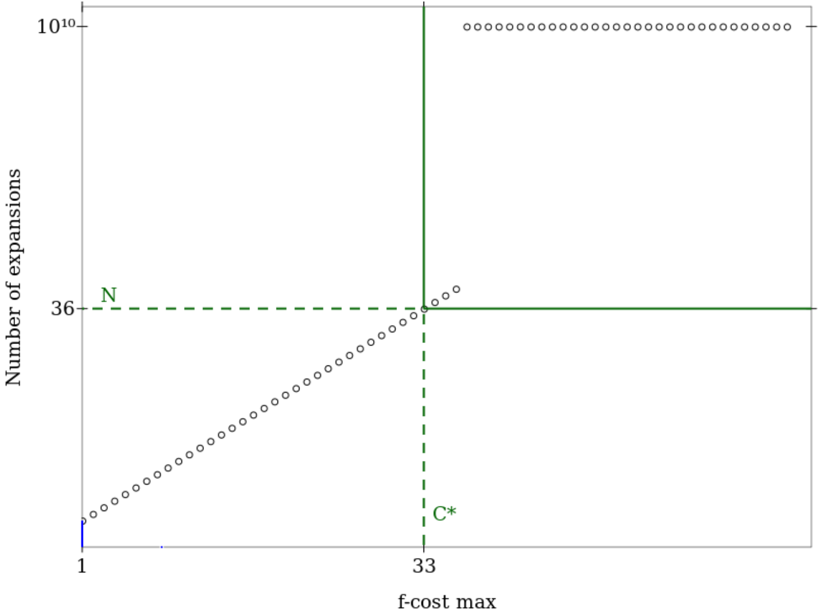
IDA* — Good Case



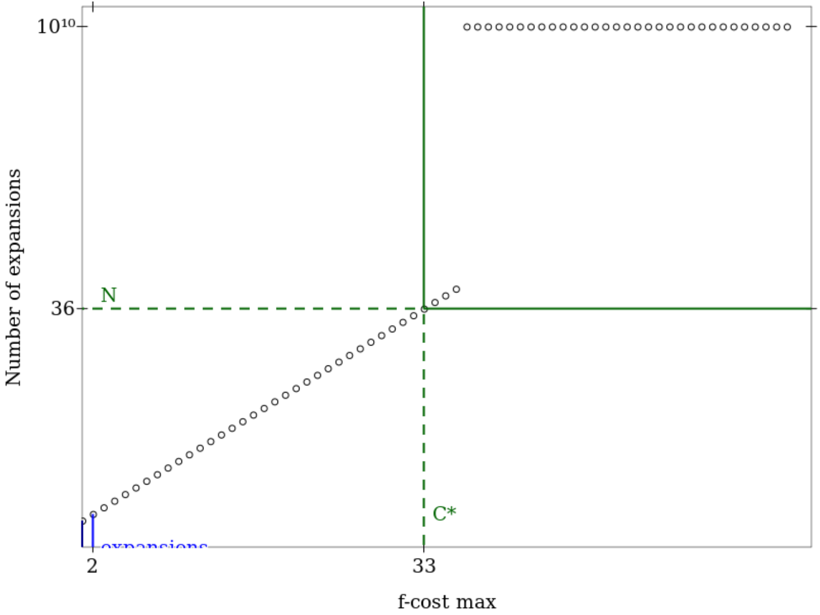
IDA* — Good Case



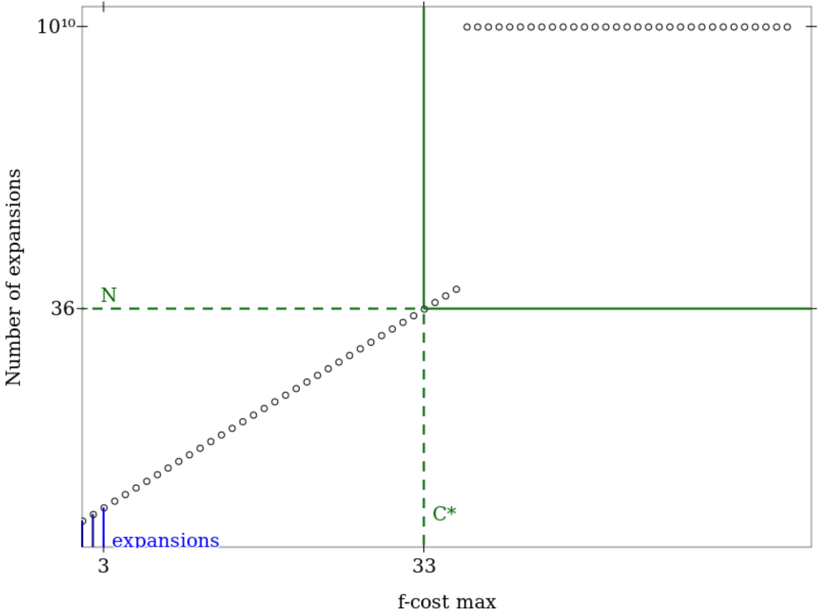
IDA* — Bad Case



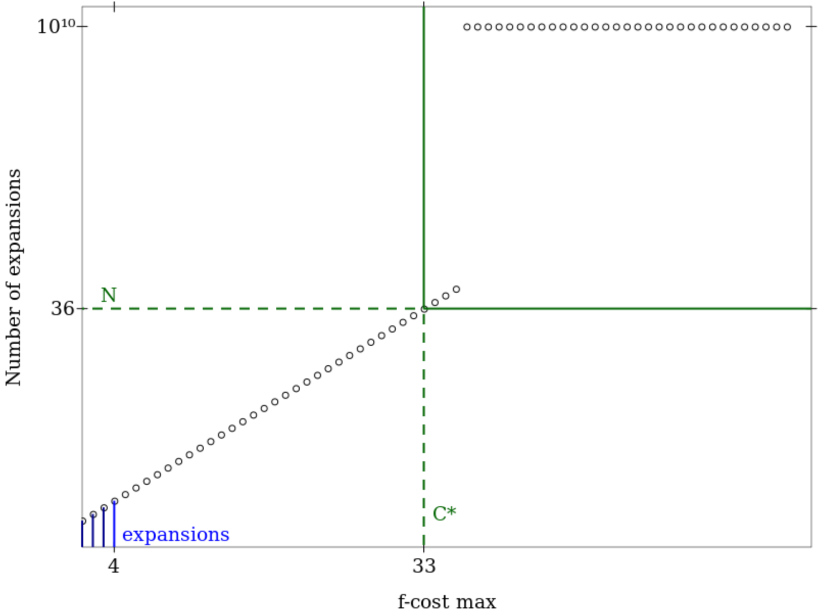
IDA* — Bad Case



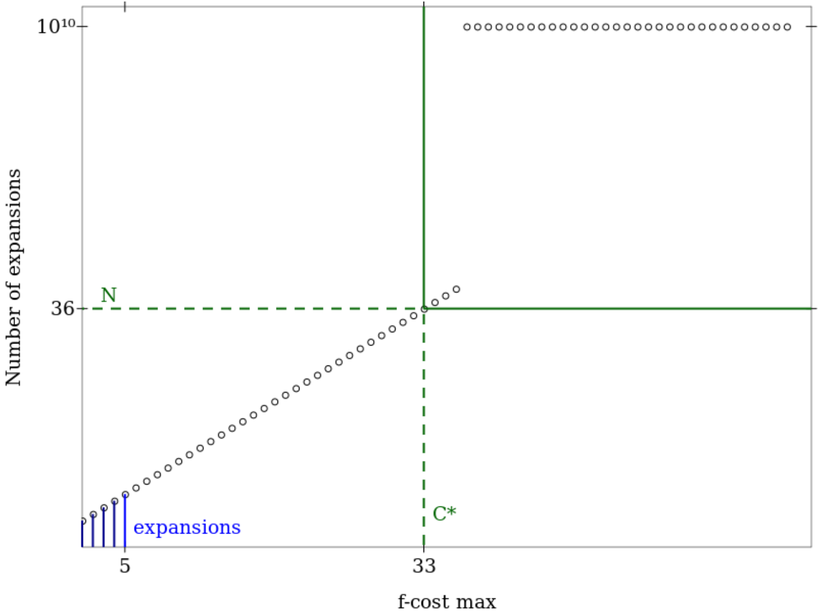
IDA* — Bad Case



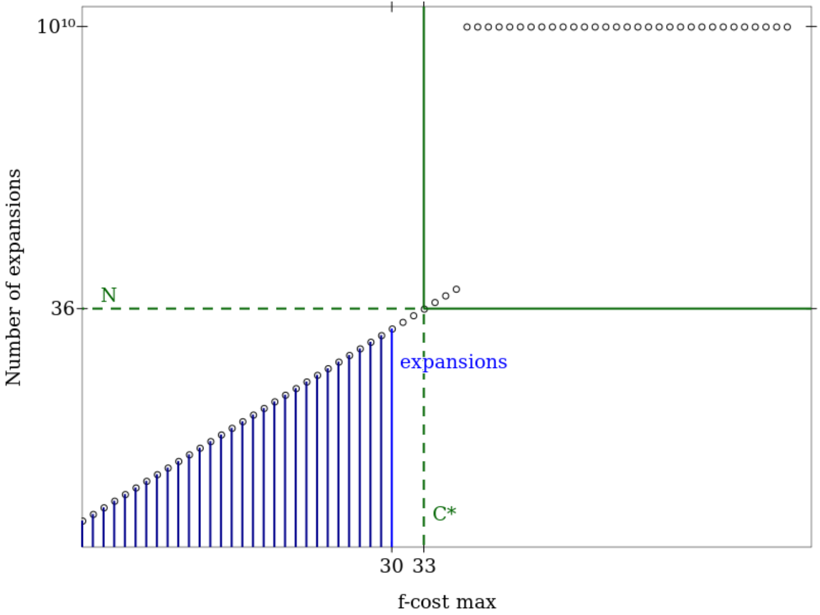
IDA* — Bad Case



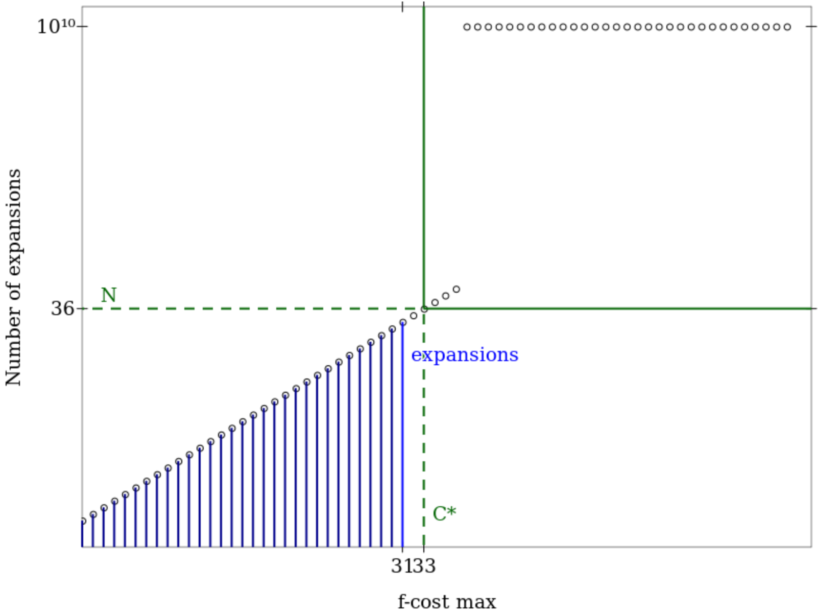
IDA* — Bad Case



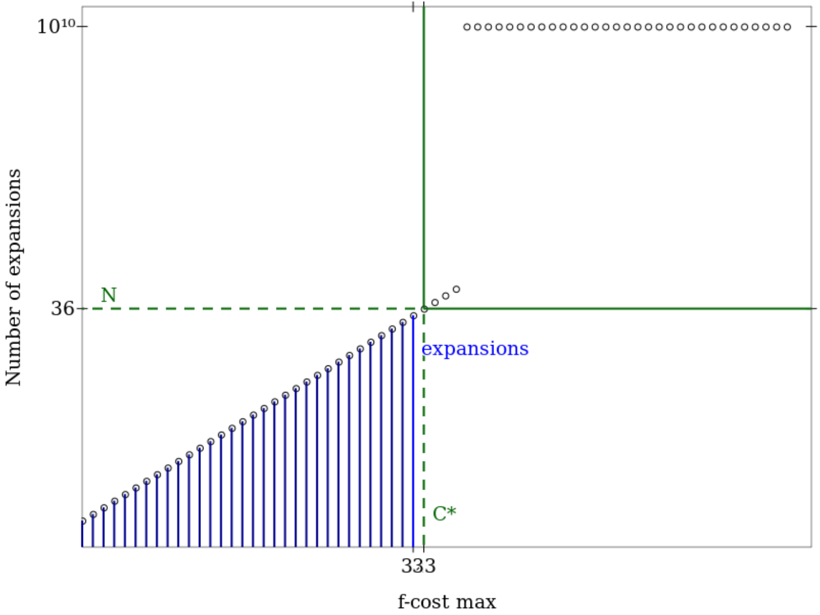
IDA* — Bad Case



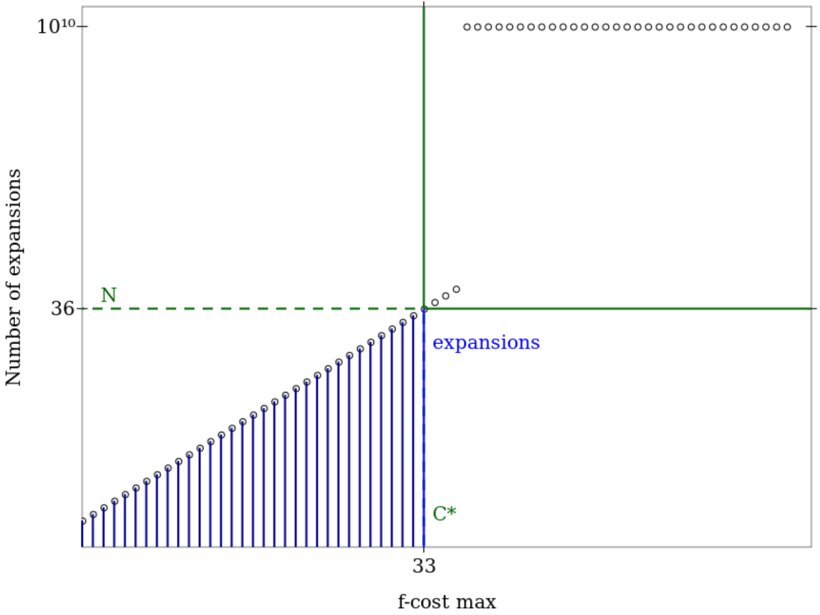
IDA* — Bad Case



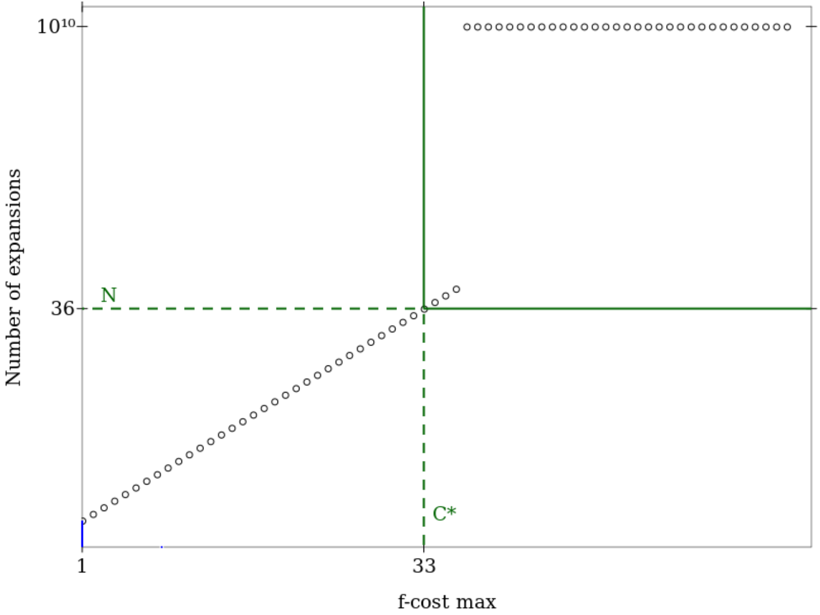
IDA* — Bad Case



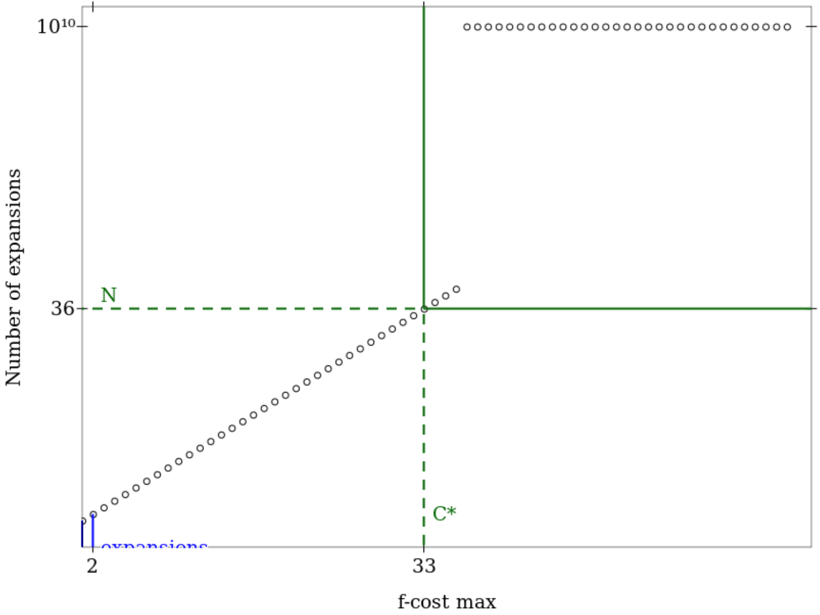
IDA* — Bad Case



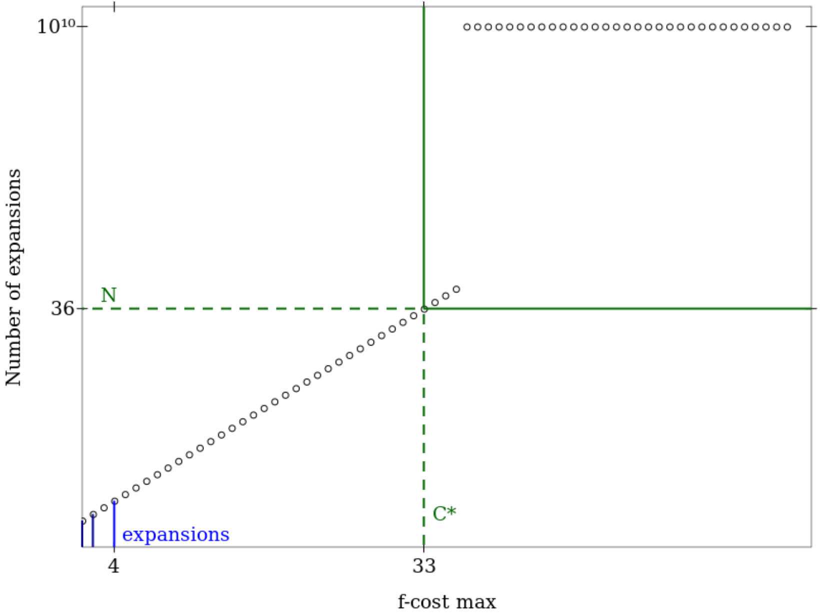
EDA*



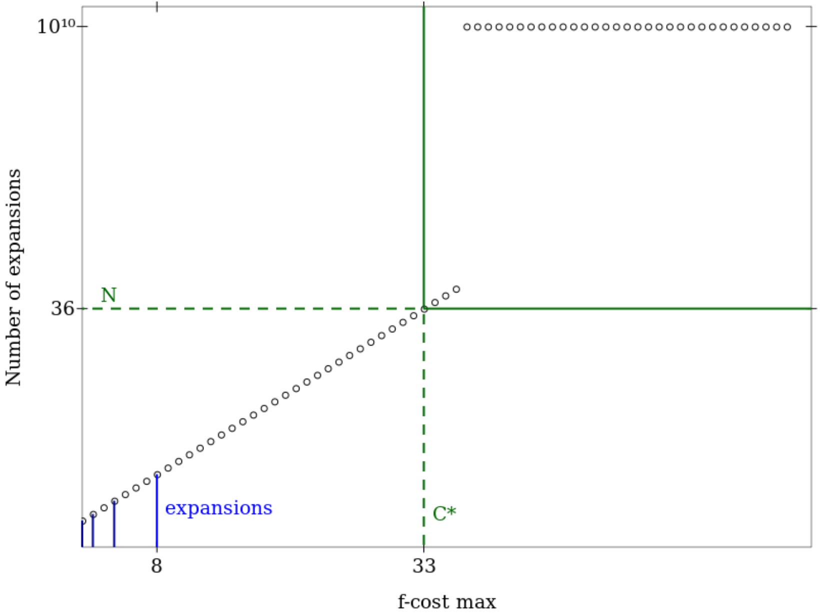
EDA*



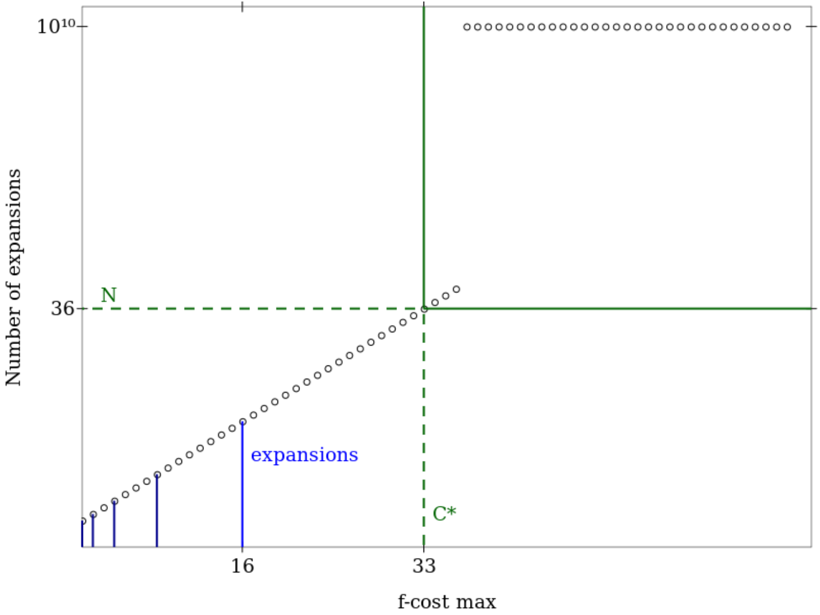
EDA*



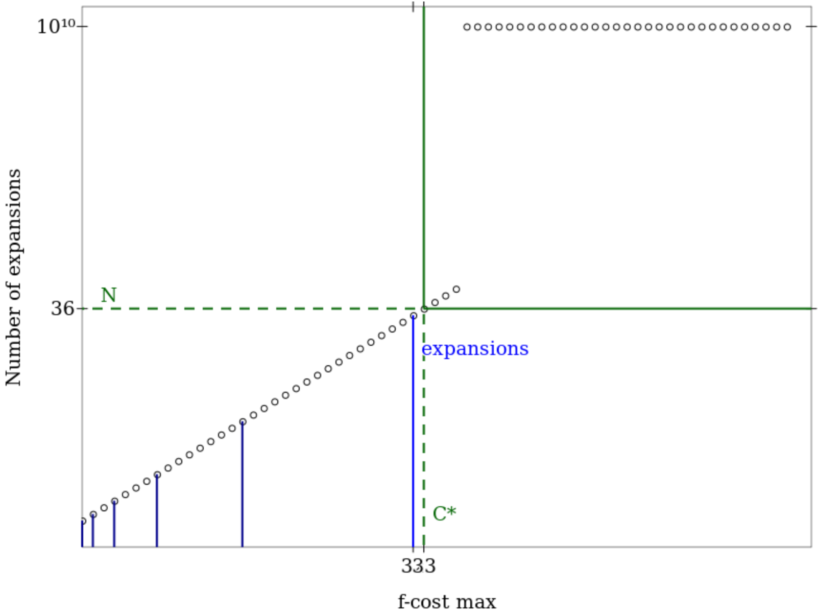
EDA*



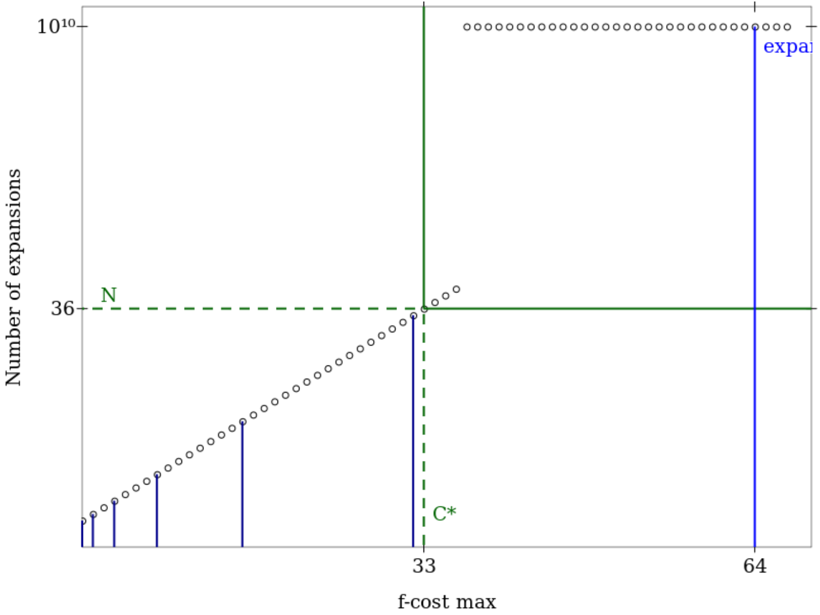
EDA*



EDA*



EDA*



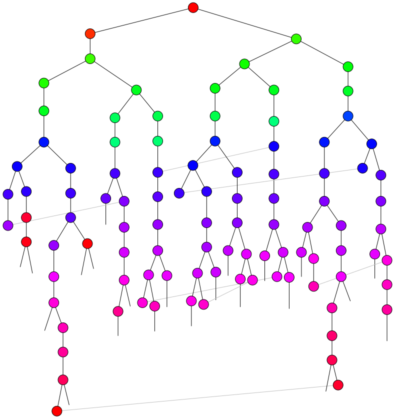
IBEX (main idea)

```
For budget = 2, 4, 8, 16, ...:
    cost_bound = Oracle(budget)
    search within cost_bound and within budget
```

Oracle(budget) returns largest cost for which budget is sufficient.

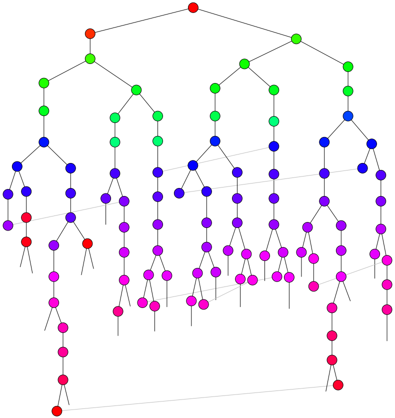
IBEX (main idea): With an oracle

budget	f-cost	expansions
2		



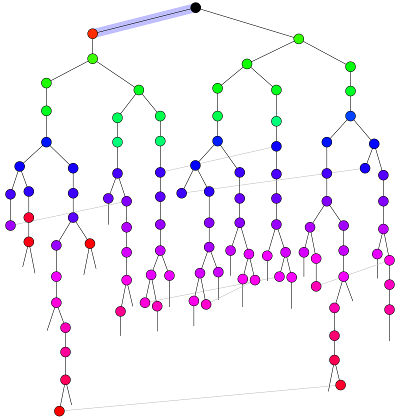
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	



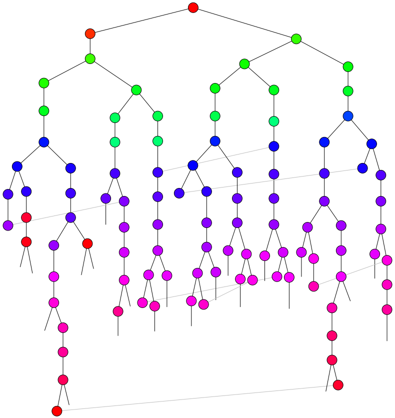
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2



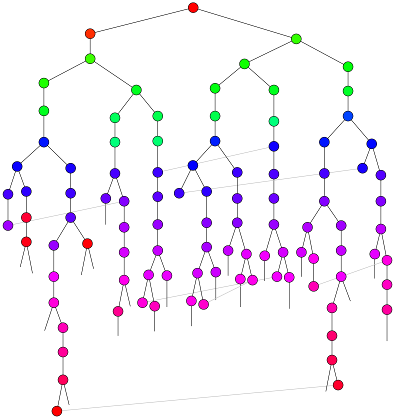
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2
4		



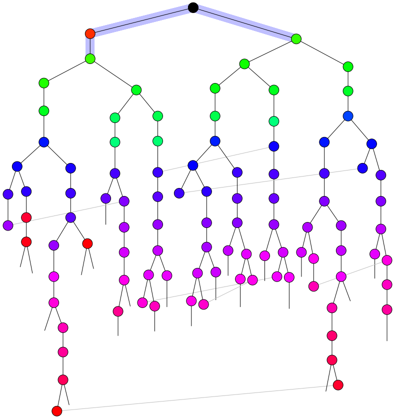
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2
4	13.5	



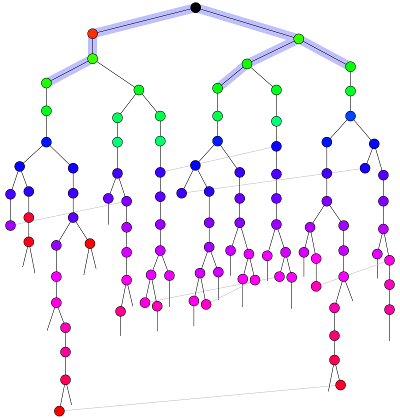
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2
4	13.5	4



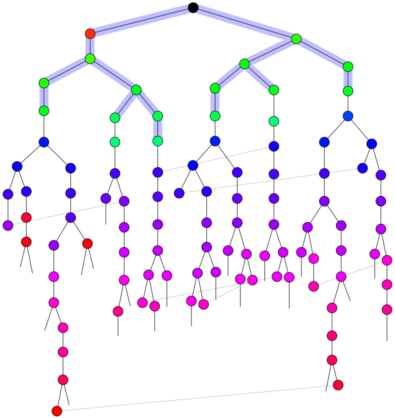
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2
4	13.5	4
8	13.9	8



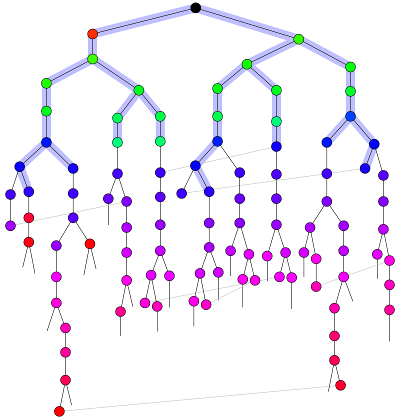
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2
4	13.5	4
8	13.9	8
16	14.4	16



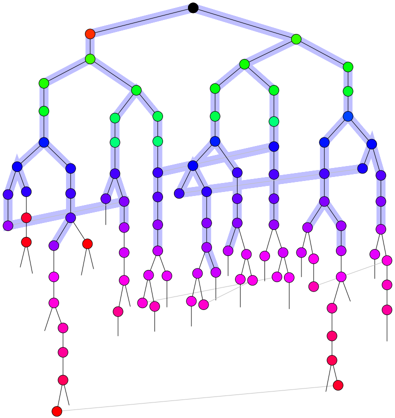
IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2
4	13.5	4
8	13.9	8
16	14.4	16
32	16.8	30



IBEX (main idea): With an oracle

budget	f-cost	expansions
2	11.2	2
4	13.5	4
8	13.9	8
16	14.4	16
32	16.8	30
64	17.7	62



IBEX (main idea)

```
For budget = 2, 4, 8, 16, ...:
    cost_bound = Oracle(budget)
    search within cost_bound and within budget
```

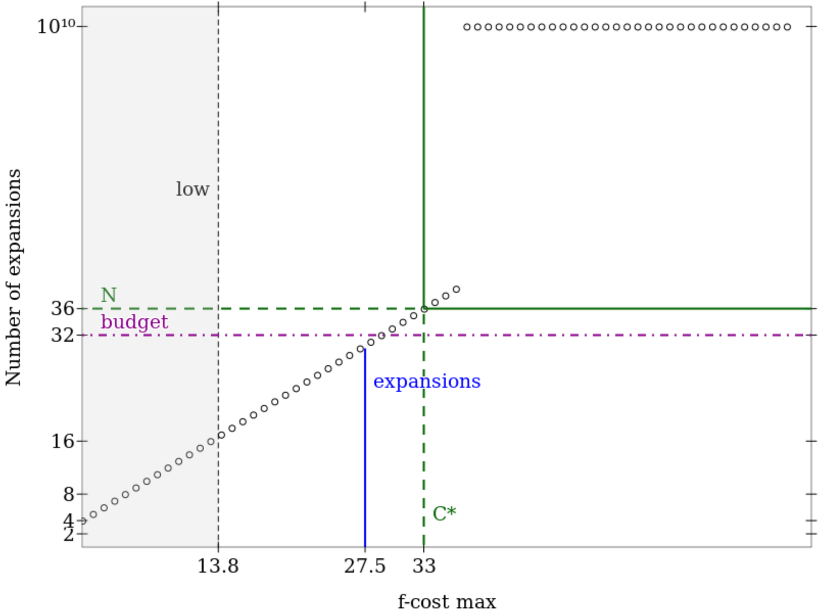
IBEX (main idea)

```
For budget = 2, 4, 8, 16, ...:
    cost_bound = Oracle(budget)
    search within cost_bound and within budget
```

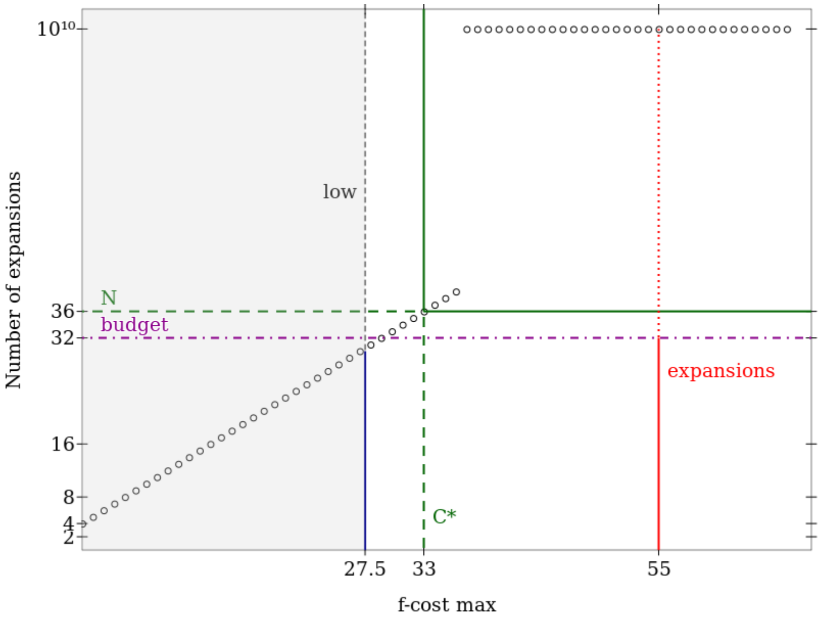
Oracle(budget) returns largest cost for which budget is sufficient.

Oracle(budget) $=$ (budgeted) exponential search

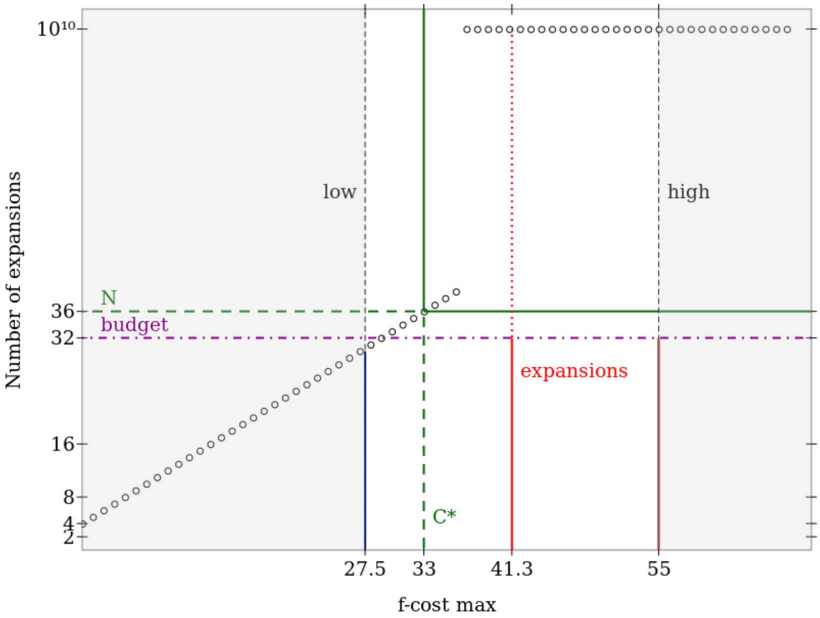
IBEX (main idea)



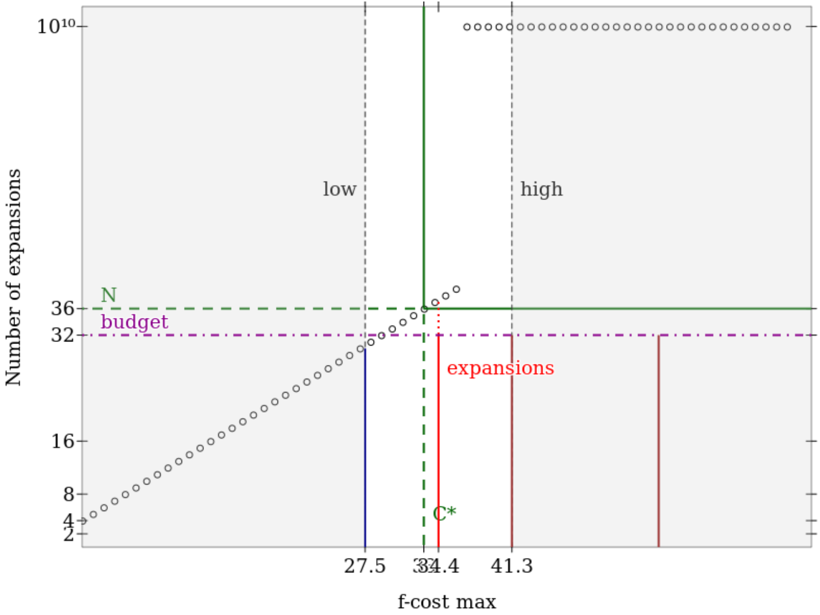
IBEX (main idea)



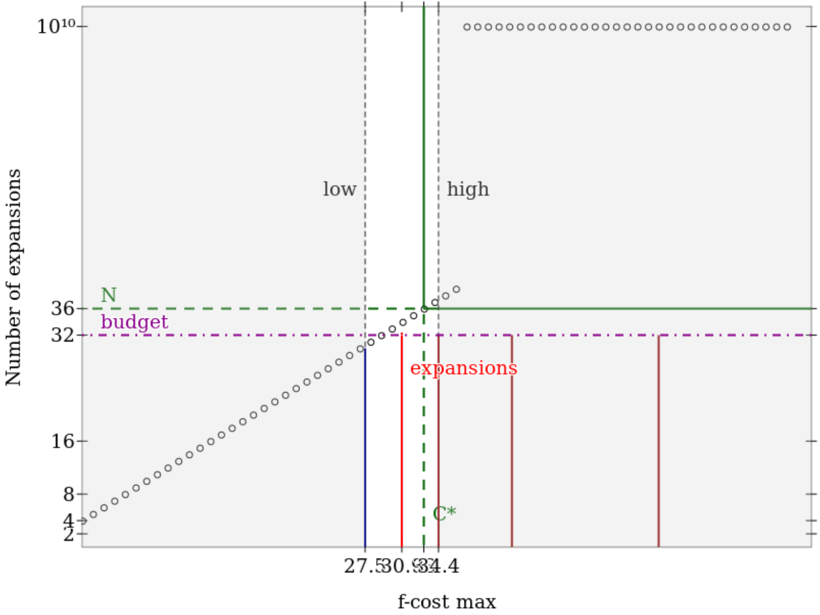
IBEX (main idea)



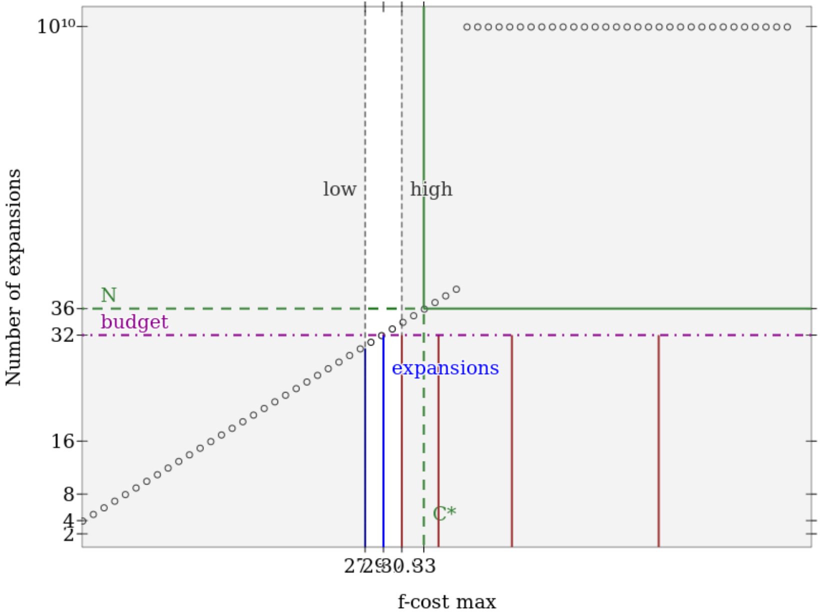
IBEX (main idea)



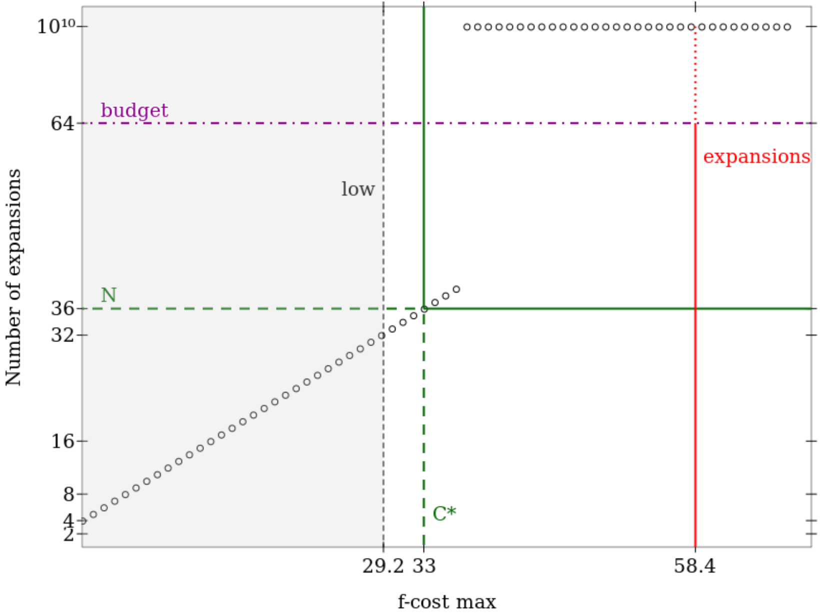
IBEX (main idea)



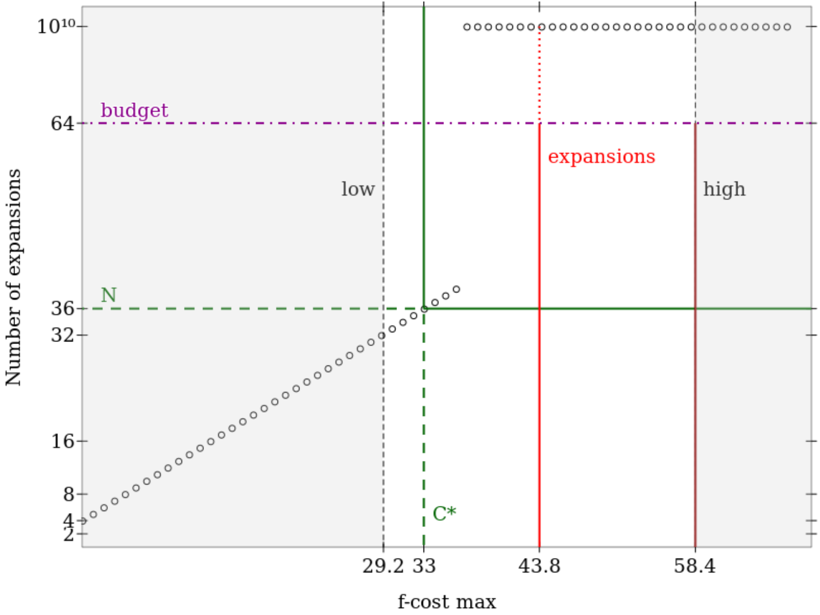
IBEX (main idea)



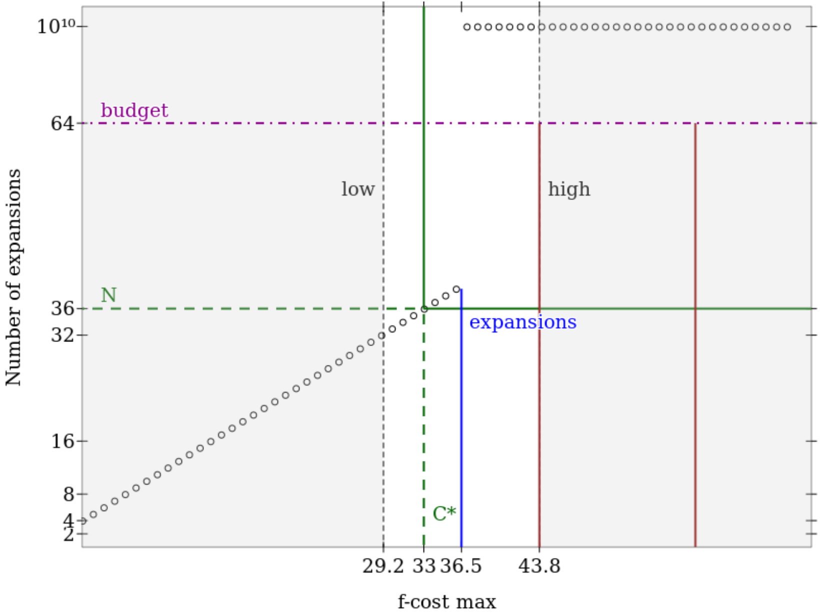
IBEX (main idea)



IBEX (main idea)



IBEX (main idea)



IBEX (main idea)

N

$\downarrow$

For budget = 2, 4, 8, 16, 32, 64 : $\leq 4N$

With **exponential search**:

Refine lower and upper bound on cost

Proves no solution when lower = upper

$\Big\}\ O(\log C^*)$

Application: Tree Search

IBEX + Depth-First Search = *Budgeted Tree Search (BTS)*

Alg.	**Worst case**
IDA*$_{(1985)}$	$\Theta(N^2)$
EDA*$_{(2014)}$	unbounded
BTS	$O(N \log C^*)$

C^*: cost of the optimal solution
N: number of nodes of cost at most C^*

All algorithms return an optimal solution (under usual assumptions).

Application: Graph Search

IBEX + Uniform-Cost Search = *Budgeted Graph Search (BGS)*

Alg.	Admissible Heuristic	
	Consistent	Inconsistent
A*$_{(1968)}$	$O(N)$	$\Omega(2^N)$
B$_{(1977)}$	$O(N)$	$\Theta(N^2)$
BGS	$O(N)^\dagger$	$O(N \log C^*)$

C^*: cost of the optimal solution
N: number of nodes of cost at most C^*

All algorithms return an optimal solution (under usual assumptions).

Enhancements: Tree search

- Recover IDA* when exponential unit-cost domains
  1) Infinite iteration at next f-cost
  2) If expansions grow exponentially, go to 1) else exponential search

Enhancements: Tree search

- Recover IDA* when exponential unit-cost domains
  1) Infinite iteration at next f-cost
  2) If expansions grow exponentially, go to 1) else exponential search
- Budget window $[2, 8]$ to be more aggressive

Experiments

Algorithm	15-Puzzle (unit)		15-Puzzle (real)		Coconut	
	Solved	Exp. $\times 10^6$	Solved	Exp. $\times 10^6$	Solved	Exp. $\times 10^4$
BTS	100	**242.5**	100	**673.1**	100	84.7
EDA*	99	5 586.0	100	2 882.3	3	$\geq$ 554 249.0
IDA*_CR	100	868.4	100	700.6	3	$\geq$ 516 937.7
IDA*	100	**242.5**	57	62 044.3	100	5 484.2
N	100	100.8	100	258.1	100	2.7

BTS *(Budgeted Tree Search)* = IBEX + Depth-First Search

Experiments

Algorithm	15-Puzzle (unit)		15-Puzzle (real)		Coconut	
	Solved	Exp. $\times 10^6$	Solved	Exp. $\times 10^6$	Solved	Exp. $\times 10^4$
BTS	100	**242.5**	100	**673.1**	100	84.7
EDA*	99	5 586.0	100	2 882.3	3	$\geq$ 554 249.0
IDA*_CR	100	868.4	100	700.6	3	$\geq$ 516 937.7
IDA*	100	**242.5**	57	62 044.3	100	5 484.2
N	100	100.8	100	258.1	100	2.7

BTS *(Budgeted Tree Search)* = IBEX + Depth-First Search

Experiments

Algorithm	15-Puzzle (unit)		15-Puzzle (real)		Coconut	
	Solved	Exp. $\times 10^6$	Solved	Exp. $\times 10^6$	Solved	Exp. $\times 10^4$
BTS	100	**242.5**	100	**673.1**	100	84.7
EDA*	99	5 586.0	100	2 882.3	3	$\geq$ 554 249.0
IDA*_CR	100	868.4	100	700.6	3	$\geq$ 516 937.7
IDA*	100	**242.5**	57	62 044.3	100	5 484.2
N	100	100.8	100	258.1	100	2.7

BTS *(Budgeted Tree Search)* = IBEX + Depth-First Search

Conclusion

- IBEX: Budgeting $\rightarrow O(N \log C)$
- Tree search: drop-in replacement for IDA*
- DovIBEX: For applications where oracle returns only if a solution is found
  - Interleaves budget iterations

Do you have a problem where the current worst case is $O(N^2)$, due to being cautious?

Maybe (Dov)IBEX can help!

Come and see our poster!