SOS: Safe, Optimal* and Small Strategies for Hybrid Markov Decision Processes

Pranav Ashok[1], Jan Kretinsky[1], Kim Guldstrand Larsen[2], Adrien Le Coënt[2], Jakob Haahr Taankvist[2] and Maximilian Weininger[1]

[1]Technical University of Munich, Munich, Germany
[2]Aalborg University, Aalborg, Denmark

Outline

1. UPPAAL Stratego
2. Safe strategies
3. Compression of strategies
4. Our proposal: Stratego+
5. Results
6. Future directions

What is Stratego?

Generate, optimize, evaluate, compare strategies

Hybrid Markov Decision Process
costs, time, data variables ...

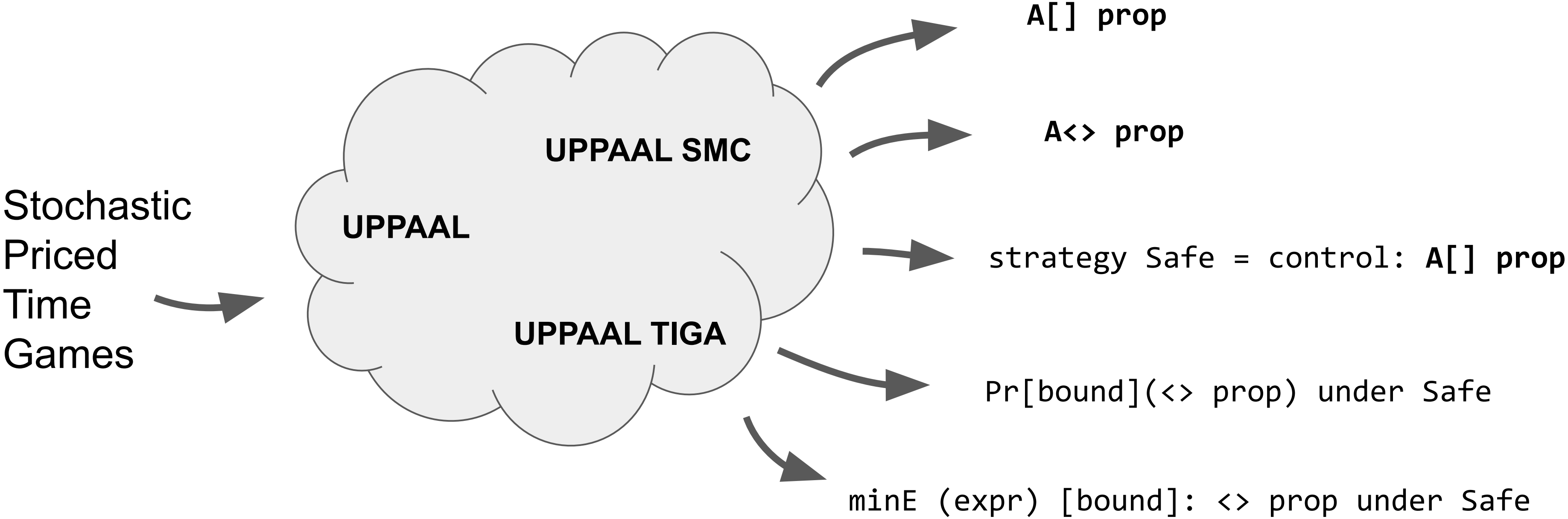
Accessed from: http://people.cs.aau.dk/~marius/stratego/

What is Stratego?

The tool allows for efficient and flexible “strategy-space” exploration before adaptation in a final implementation by maintaining strategies as first class objects in the model-checking query language.

Accessed from: http://people.cs.aau.dk/~marius/stratego/

What is Stratego?



Stochastic
Priced
Time
Games

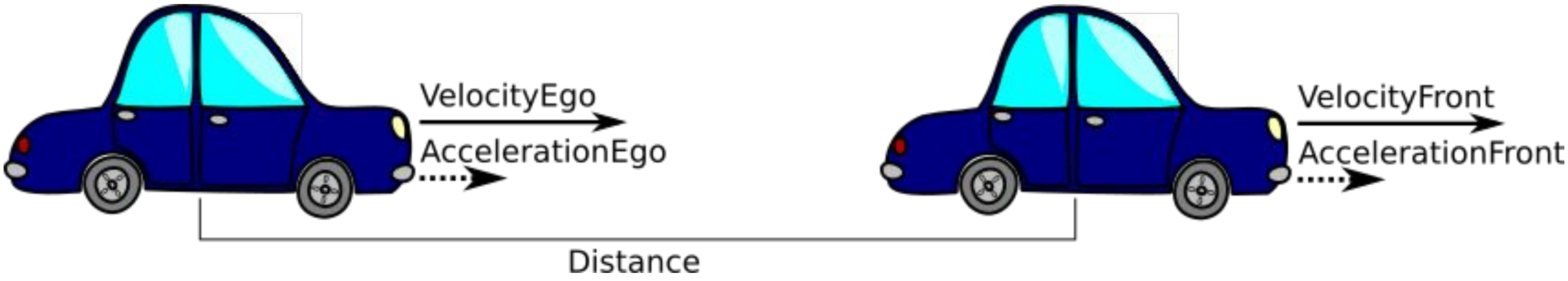
UPPAAL

UPPAAL SMC

UPPAAL TIGA

```
A[] prop

A<> prop

strategy Safe = control: A[] prop

Pr[bound](<> prop) under Safe

minE (expr) [bound]: <> prop under Safe
```

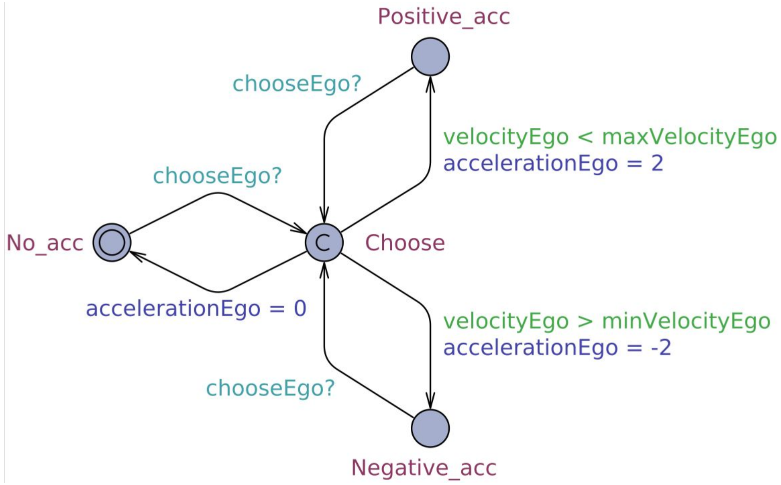
Strategies from UPPAAL Stratego
are **huge** and **incomprehensible**



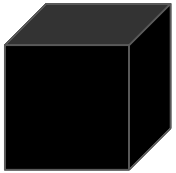
safety: A[] distance ≥ 5

optimality: minimize aggregate distance

Ego



Front



safety: A[] distance ≥ 5

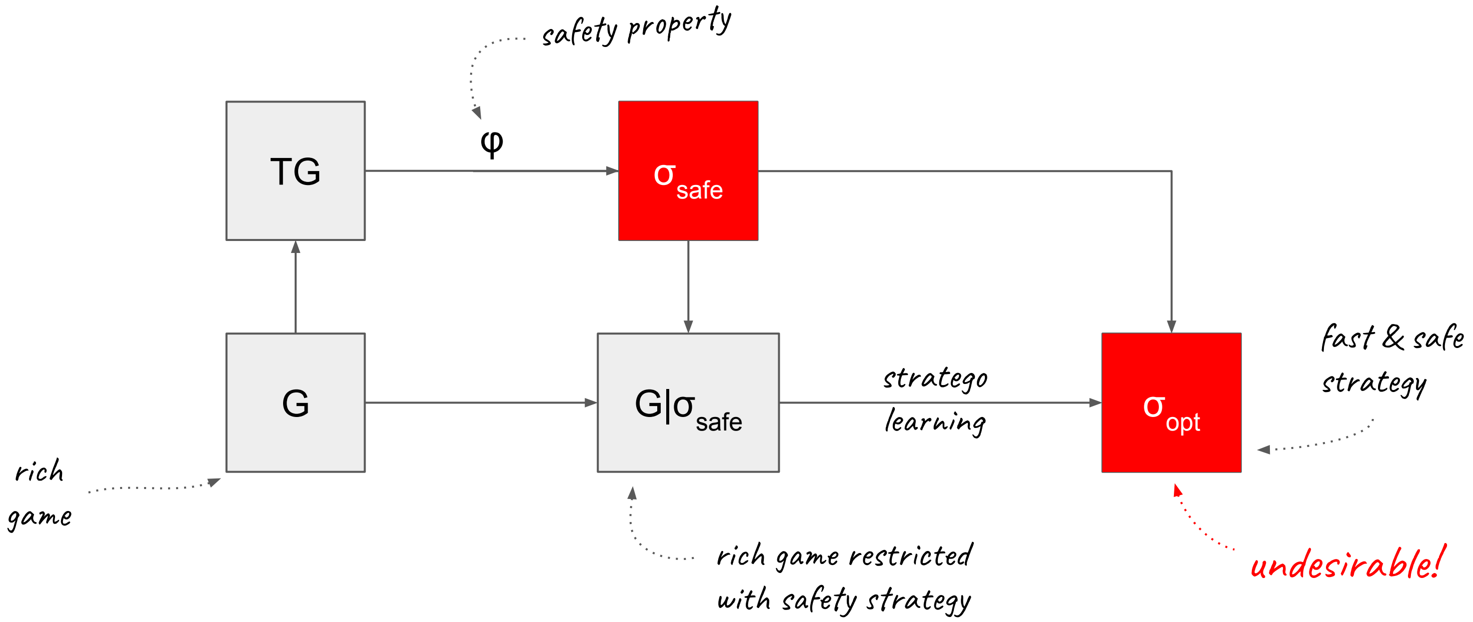
optimality: minimize aggregate distance

Sample safe strategy

```
State: ( Ego.No_acc Front.No_acceleration ) distance=200 velocityEgo=16 accelerationEgo=0 velocityFront=12
accelerationFront=0
Wait.

State: ( Ego.Choose Front.Negative_acc ) distance=101 velocityEgo=-6 accelerationEgo=0 velocityFront=-4
accelerationFront=-2
Take transition Ego.Choose->Ego.No_acc
Take transition Ego.Choose->Ego.Positive_acc
Take transition Ego.Choose->Ego.Negative_acc

State: ( Ego.No_acc Front.Positive_acc ) distance=82 velocityEgo=2 accelerationEgo=0 velocityFront=10
accelerationFront=2
Wait.

State: ( Ego.No_acc Front.Positive_acc System.FrontNext Monitor._id12 ) distance=85
```

Stratego Internals

Generating a safe and optimal controller for cruise control



Problems

- 900k state action pairs, 300k controllable
- Incomprehensible for humans
- Lookup table too big for microcontrollers
- Executing inside Stratego takes too long

Would there exist a **small and safe** strategy?

Can we learn a smaller representation?

Can we learn a smaller representation?

Binary Decision Diagrams?

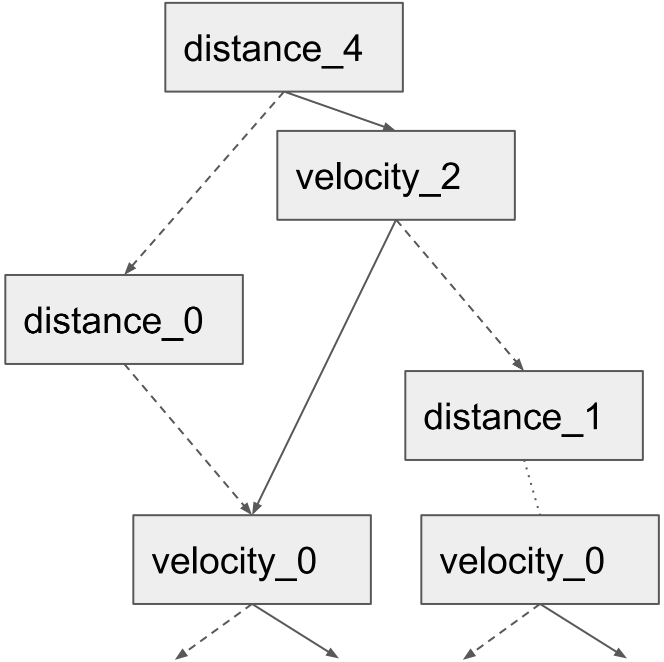
Can we learn a smaller representation?

Each state is composed of variables with integer domains

```
State: ( Ego.No_acc Front.No_acceleration ) distance=200 velocityEgo=16 accelerationEgo=0 velocityFront=12
accelerationFront=0
Wait.

State: ( Ego.Choose Front.Negative_acc ) distance=101 velocityEgo=-6 accelerationEgo=0 velocityFront=-4
accelerationFront=-2
Take transition Ego.Choose->Ego.No_acc
Take transition Ego.Choose->Ego.Positive_acc
Take transition Ego.Choose->Ego.Negative_acc

State: ( Ego.No_acc Front.Positive_acc ) distance=82 velocityEgo=2 accelerationEgo=0 velocityFront=10
accelerationFront=2
Wait.

State: ( Ego.No_acc Front.Positive_acc System.FrontNext Monitor._id12 ) distance=85
```

Can we learn a smaller representation?

Each state is composed of variables with integer domains

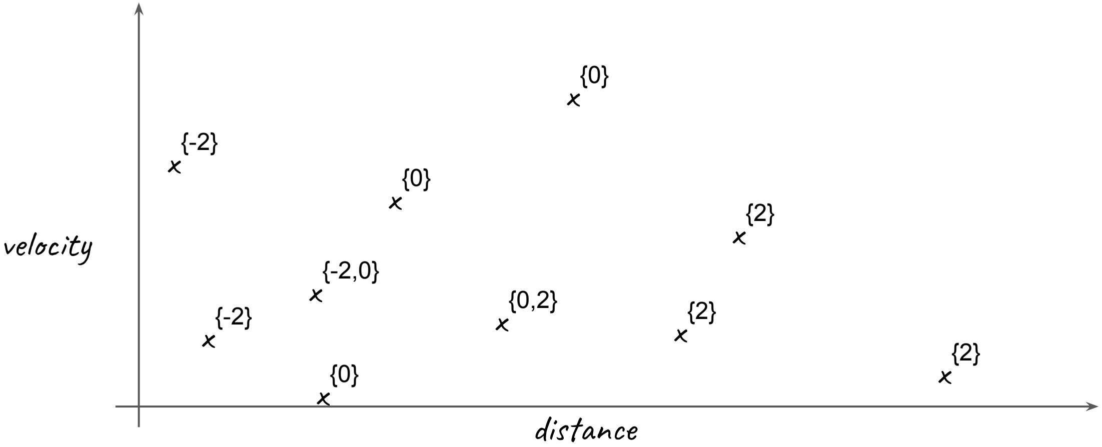


Can we learn a smaller representation?

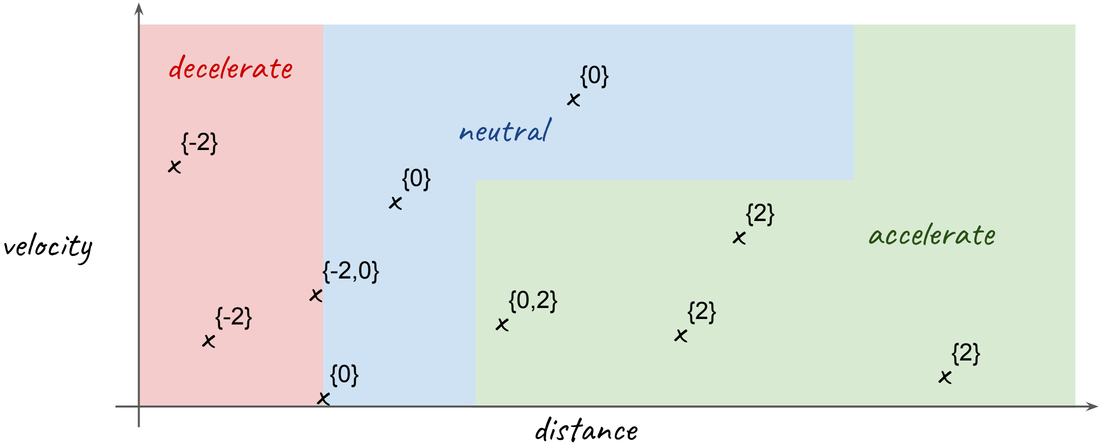
Nearby points behave same?

```
State: ( Ego.No_acc Front.No_acceleration ) distance=200 velocityEgo=16 accelerationEgo=0 velocityFront=12
accelerationFront=0
Wait.

State: ( Ego.Choose Front.Negative_acc ) distance=101 velocityEgo=-6 accelerationEgo=0 velocityFront=-4
accelerationFront=-2
Take transition Ego.Choose->Ego.No_acc
Take transition Ego.Choose->Ego.Positive_acc
Take transition Ego.Choose->Ego.Negative_acc

State: ( Ego.No_acc Front.Positive_acc ) distance=82 velocityEgo=2 accelerationEgo=0 velocityFront=10
accelerationFront=2
Wait.

State: ( Ego.No_acc Front.Positive_acc System.FrontNext Monitor._id12 ) distance=85
```

Can we learn a smaller representation?

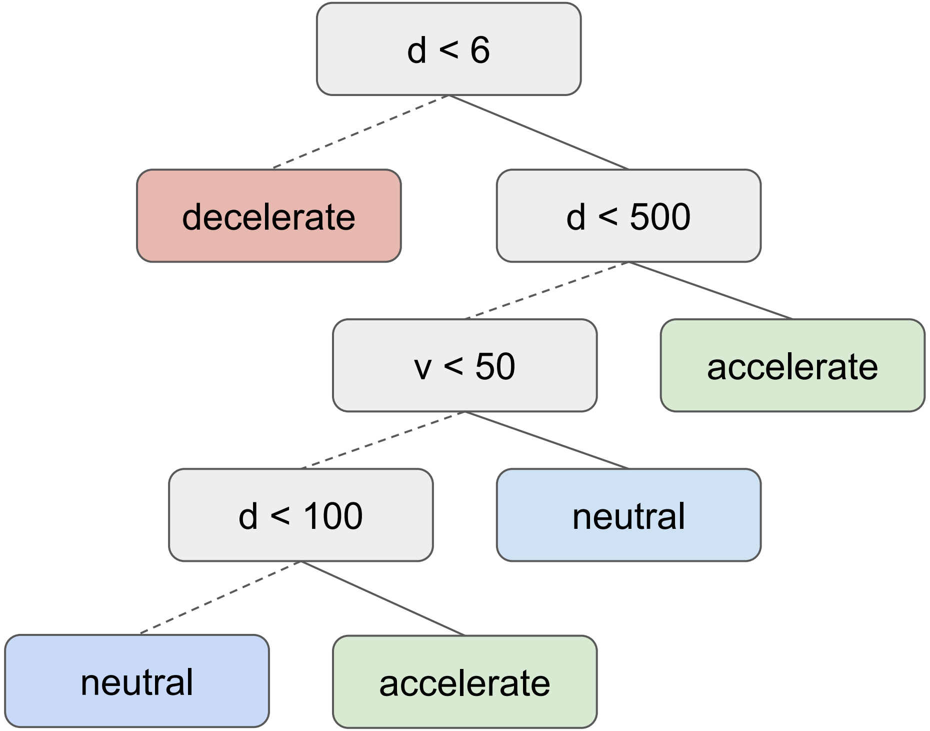


Can we learn a smaller representation?



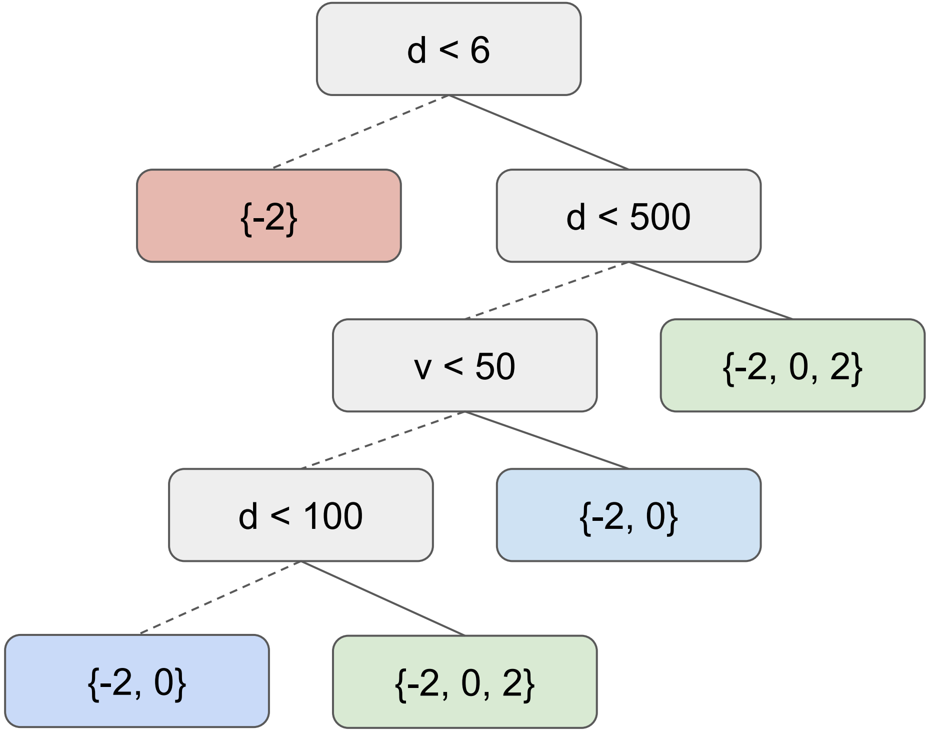
Decision Trees

- No magic inside, unlike many other ML techniques
- Simple to interpret
- Good with features which have inherent ordering
- Can be converted into executable code easily

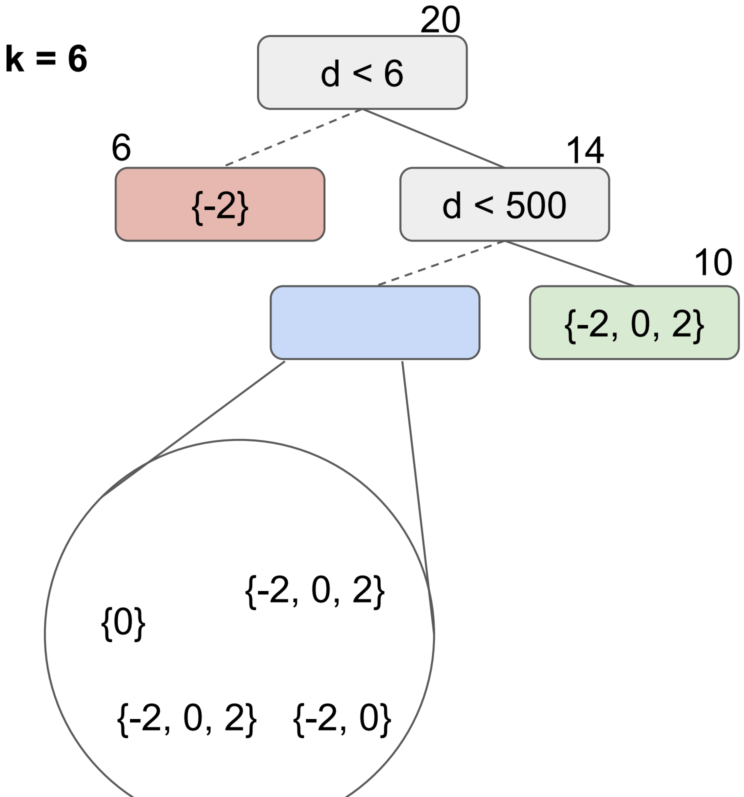


Construction of DTs

1. Multi-label classification
2. All leaves homogeneous
3. Size-permissiveness tradeoff



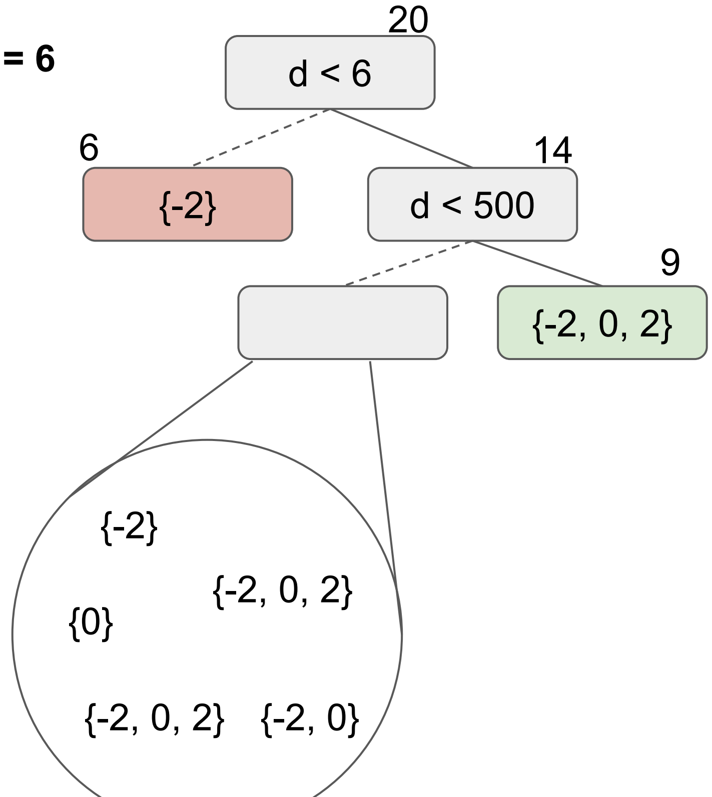
Minimum split size



Consider splitting node only if # data points > k

Minimum split size

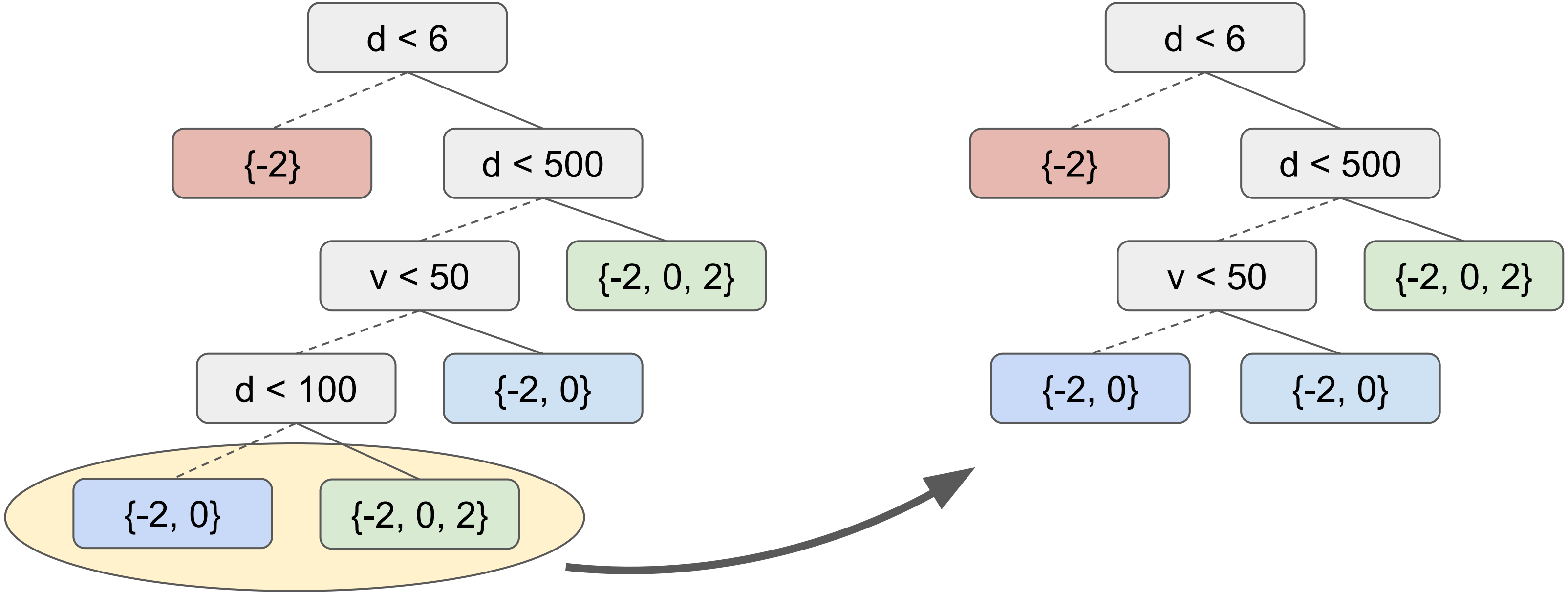
k = 6



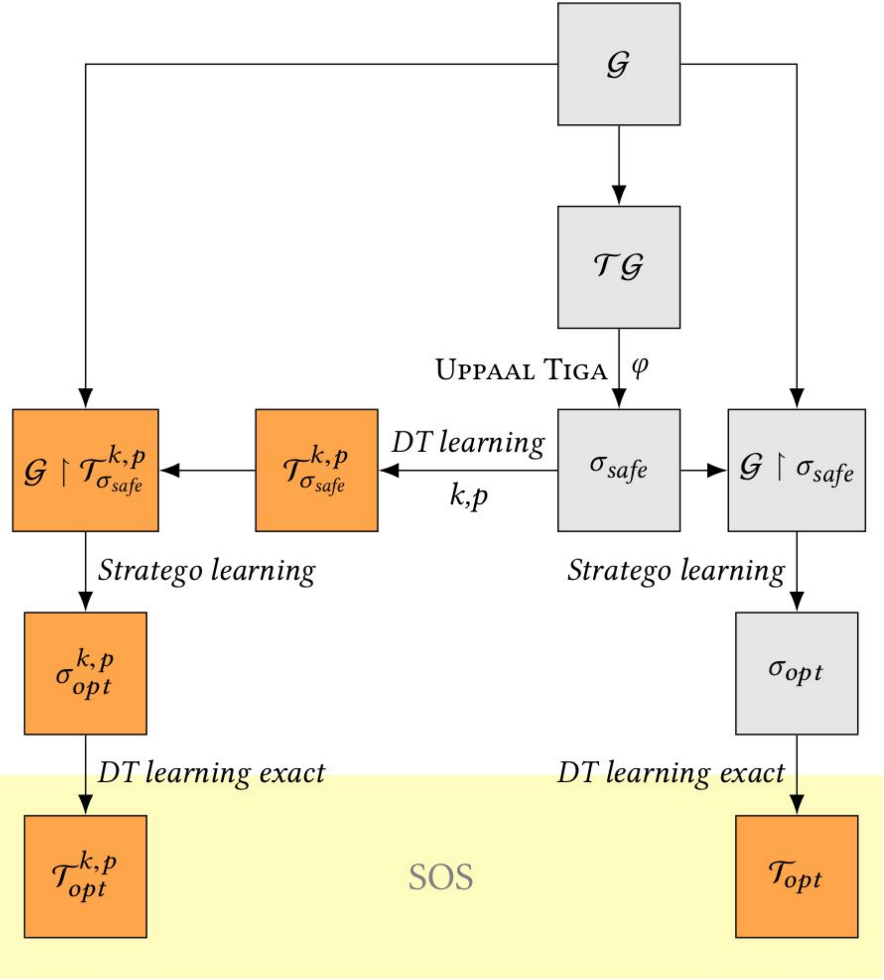
Consider splitting node only if # data points > k

Safe Pruning

Merge leaves if intersection non-empty



Stratego+ Framework



Experiments

cruise

strategy guarantees safety only at integer points

cruise-euler

enriched cruise

strategy guarantees safety at all points

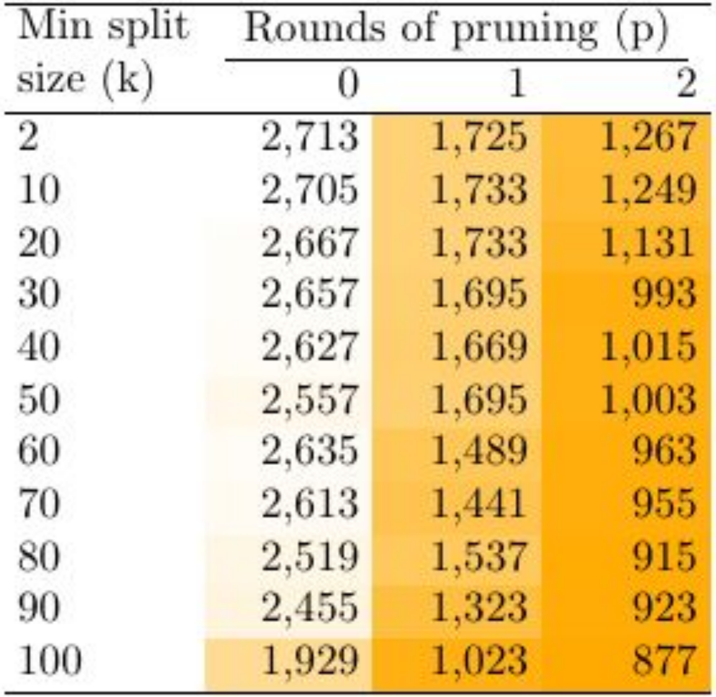
tworooms-euler

automatic climate control

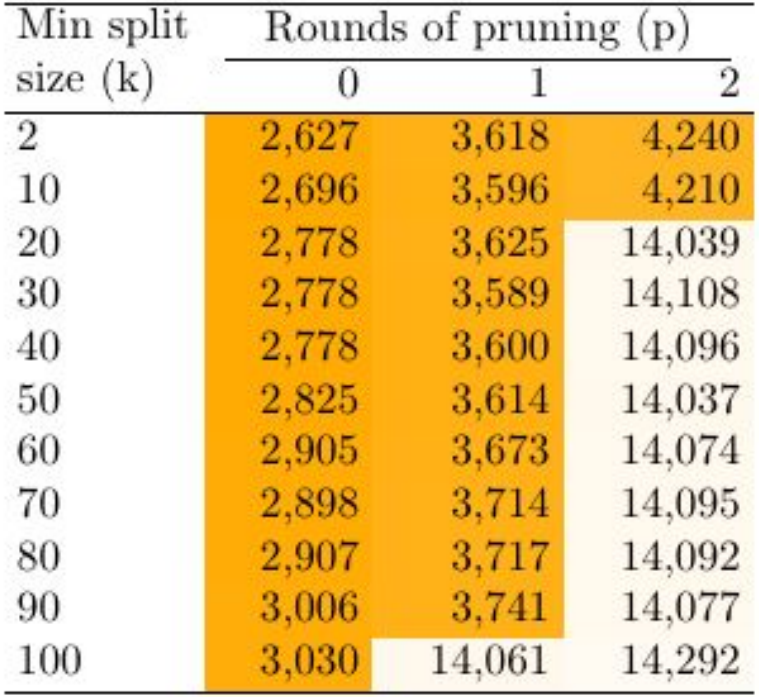
interaction between two rooms, env. and heaters

Experimental Results

Model	State-action pairs	Controllable	BDD (reordered)	DT T_{opt}
~~cruise~~	~~1,790,034~~	~~308,216~~	~~5066~~	~~2,899~~
~~cruise-euler~~	~~5,931,154~~	~~304,752~~	~~4728~~	~~2,713~~
cruise	817,278	295,970	2,730	1,005
cruise-euler	1,140,756	414,899	2,667	1,025
two-rooms	1,924,708	509,715	20,214	487



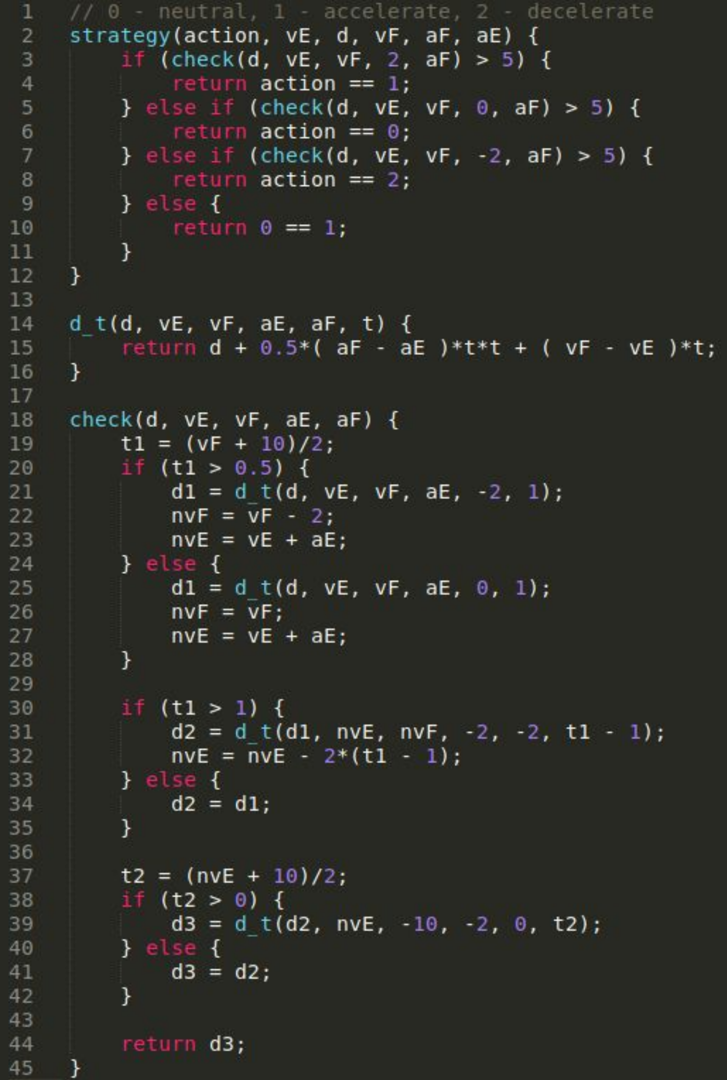
Min split	Rounds of pruning (p)		
size (k)	0	1	2
2	2,713	1,725	1,267
10	2,705	1,733	1,249
20	2,667	1,733	1,131
30	2,657	1,695	993
40	2,627	1,669	1,015
50	2,557	1,695	1,003
60	2,635	1,489	963
70	2,613	1,441	955
80	2,519	1,537	915
90	2,455	1,323	923
100	1,929	1,023	877



Min split	Rounds of pruning (p)		
size (k)	0	1	2
2	2,627	3,618	4,240
10	2,696	3,596	4,210
20	2,778	3,625	14,039
30	2,778	3,589	14,108
40	2,778	3,600	14,096
50	2,825	3,614	14,037
60	2,905	3,673	14,074
70	2,898	3,714	14,095
80	2,907	3,717	14,092
90	3,006	3,741	14,077
100	3,030	14,061	14,292

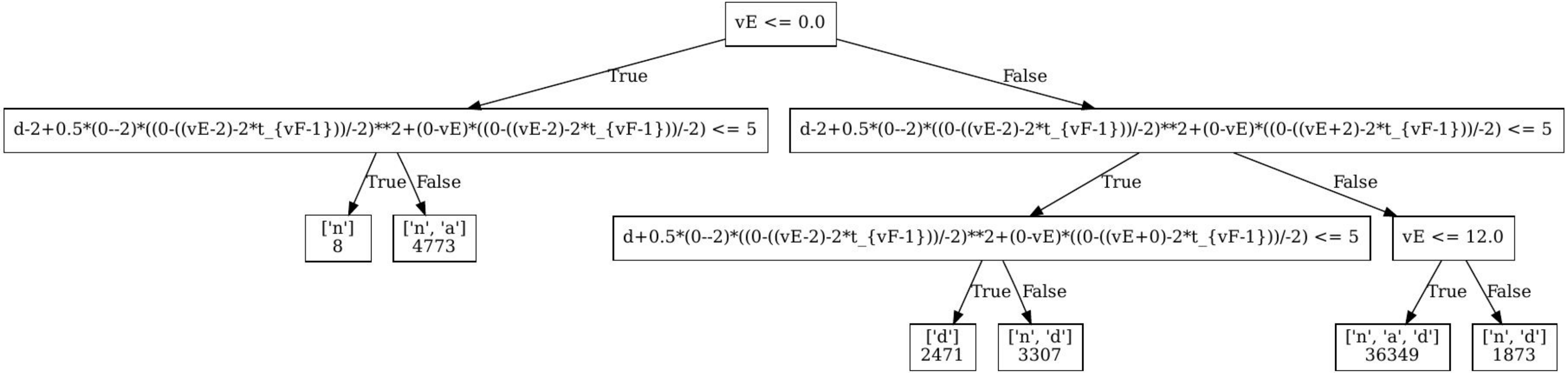
Experimental Results: `cruise-euler` size-optimality tradeoff

Handcrafted Strategy



```
// 0 - neutral, 1 - accelerate, 2 - decelerate
strategy(action, vE, d, vF, aF, aE) {
    if (check(d, vE, vF, 2, aF) > 5) {
        return action == 1;
    } else if (check(d, vE, vF, 0, aF) > 5) {
        return action == 0;
    } else if (check(d, vE, vF, -2, aF) > 5) {
        return action == 2;
    } else {
        return 0 == 1;
    }
}

d_t(d, vE, vF, aE, aF, t) {
    return d + 0.5*( aF - aE )*t*t + ( vF - vE )*t;
}

check(d, vE, vF, aE, aF) {
    t1 = (vF + 10)/2;
    if (t1 > 0.5) {
        d1 = d_t(d, vE, vF, aE, -2, 1);
        nvF = vF - 2;
        nvE = vE + aE;
    } else {
        d1 = d_t(d, vE, vF, aE, 0, 1);
        nvF = vF;
        nvE = vE + aE;
    }

    if (t1 > 1) {
        d2 = d_t(d1, nvE, nvF, -2, -2, t1 - 1);
        nvE = nvE - 2*(t1 - 1);
    } else {
        d2 = d1;
    }

    t2 = (nvE + 10)/2;
    if (t2 > 0) {
        d3 = d_t(d2, nvE, -10, -2, 0, t2);
    } else {
        d3 = d2;
    }

    return d3;
}
```

Strategy Preview: cruise



vE <= 0.0

True

False

d-2+0.5*(0--2)*((0-((vE-2)-2*t_{vF-1}))/-2)**2+(0-vE)*((0-((vE-2)-2*t_{vF-1}))/-2) <= 5

d-2+0.5*(0--2)*((0-((vE-2)-2*t_{vF-1}))/-2)**2+(0-vE)*((0-((vE+2)-2*t_{vF-1}))/-2) <= 5

True

False

True

False

['n']
8

['n', 'a']
4773

d+0.5*(0--2)*((0-((vE-2)-2*t_{vF-1}))/-2)**2+(0-vE)*((0-((vE+0)-2*t_{vF-1}))/-2) <= 5

vE <= 12.0

True

False

True

False

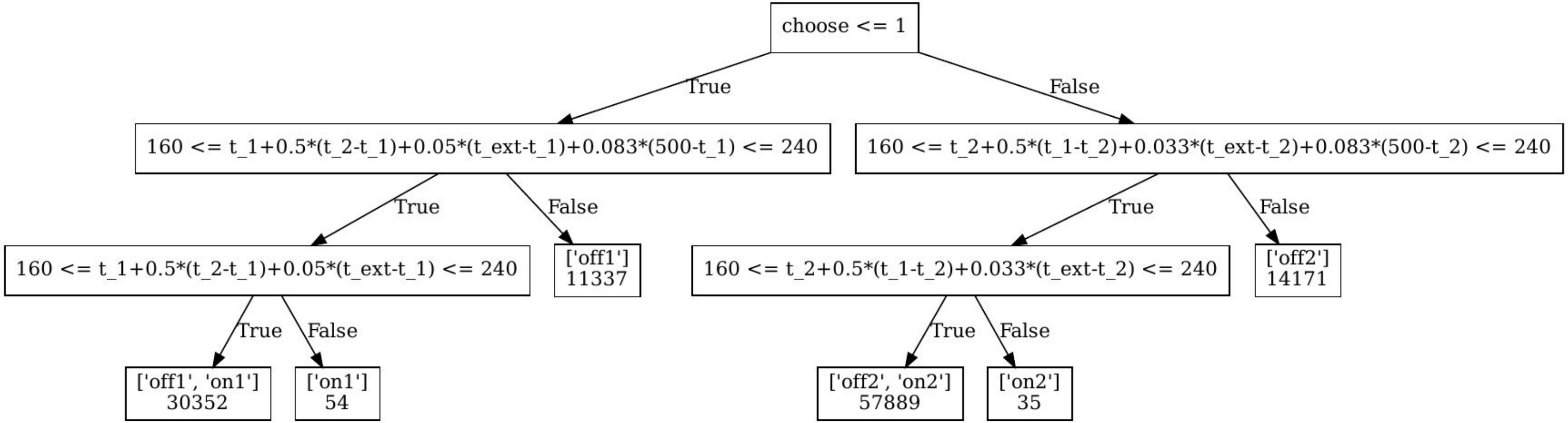
['d']
2471

['n', 'd']
3307

['n', 'a', 'd']
36349

['n', 'd']
1873

Strategy Preview: tworooms



choose <= 1

True

False

160 <= t_1+0.5*(t_2-t_1)+0.05*(t_ext-t_1)+0.083*(500-t_1) <= 240

160 <= t_2+0.5*(t_1-t_2)+0.033*(t_ext-t_2)+0.083*(500-t_2) <= 240

True

False

160 <= t_1+0.5*(t_2-t_1)+0.05*(t_ext-t_1) <= 240

['off1']
11337

True

False

160 <= t_2+0.5*(t_1-t_2)+0.033*(t_ext-t_2) <= 240

['off2']
14171

True

False

['off1', 'on1']
30352

['on1']
54

True

False

['off2', 'on2']
57889

['on2']
35

Concluding Remarks

Problem: Strategies from UPPAAL Stratego are large and incomprehensible

Solution: Stratego+ framework representing safe, small, optimal strategies as DTs

Takeaways

- BDDs are insufficient (uninterpretable, not that small either)
- Great prospects from decision trees

Future work

- Linear/algebraic predicates + domain knowledge

Backup

Experimental Results

Model	State-action pairs	Controllable	BDD (median)	DT T_{safe}	DT T_{opt}
cruise	817,278	295,970	2,730	1,017	1,005
cruise-euler	1,140,756	414,899	4,728	1,045	1,025
two-rooms	1,924,708	509,715	20,214	543	487