Optimal routing
of
autonomous vehicles
in stochastic environments

George V. Moustakides
University of Patras, Greece
& Columbia University, USA

Outline

- Routing and type of environments
- Routing of "unintelligent" vehicles
- Routing of "intelligent" vehicles
  - A cost based optimization problem
  - Optimal solution for "impatient" vehicles
  - Optimal solution for "patient" vehicles
- Extensions

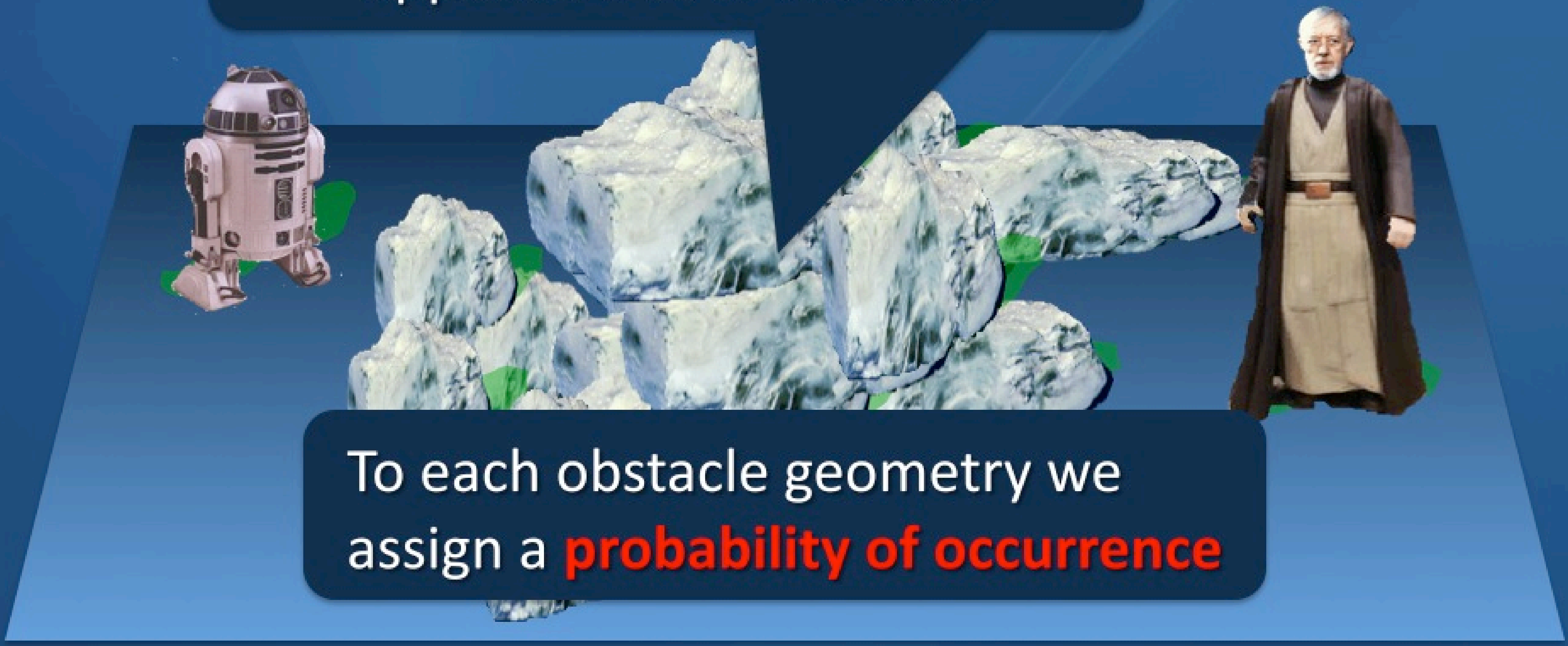
Routing and type of environments

Deterministic / Static

Deterministic / Dynamic

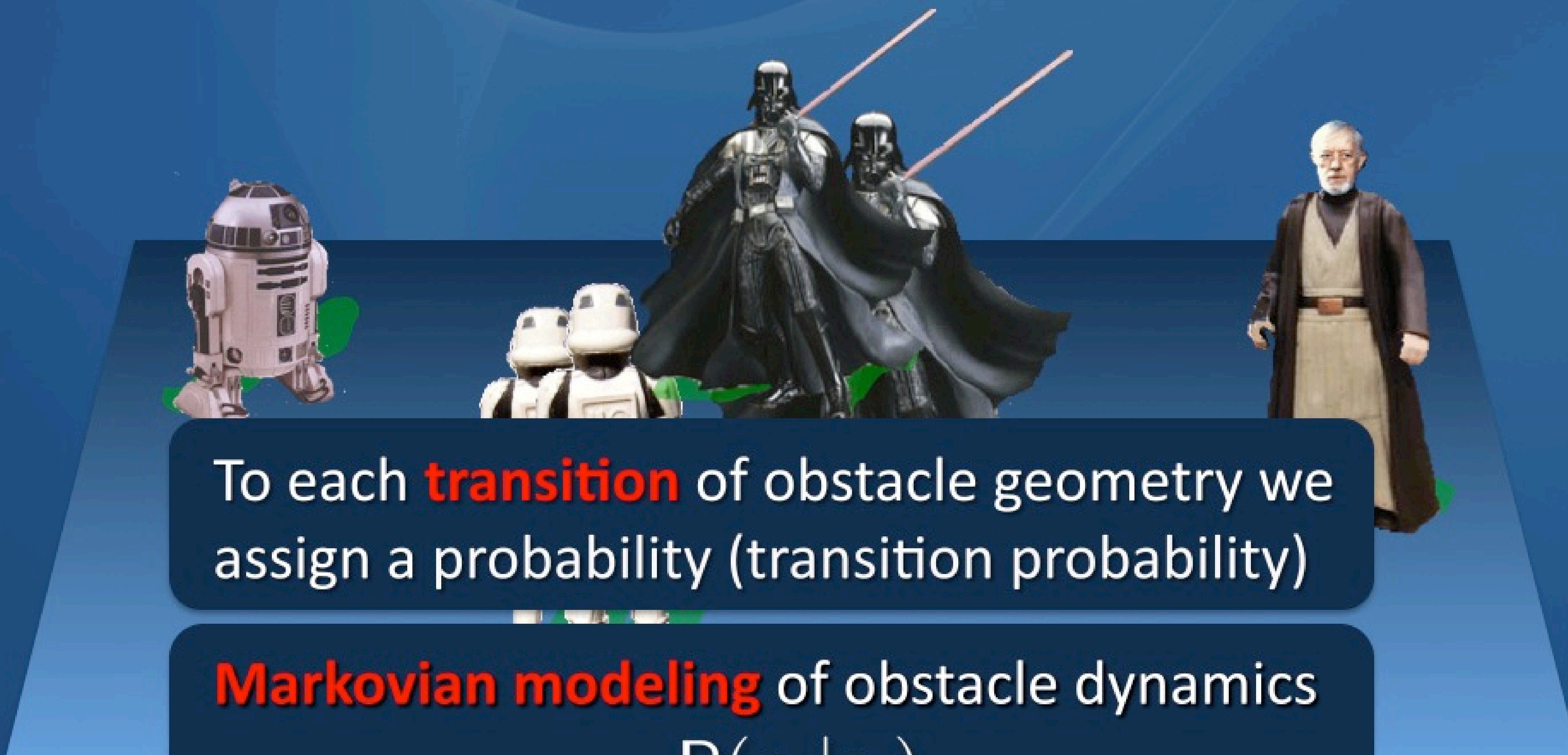
Stochastic / Static

This specific obstacle geometry appears 17% of the time



To each obstacle geometry we assign a **probability of occurrence**

Stochastic / Dynamic



To each **transition** of obstacle geometry we assign a probability (transition probability)

Markovian modeling of obstacle dynamics

$$\mathsf{P}(s_j|s_i)$$

Routing of "unintelligent" vehicles

If the only available information is the transition probabilities of the obstacle geometries, then there exists an **optimum route** that can be pre computed.

Optimality criteria

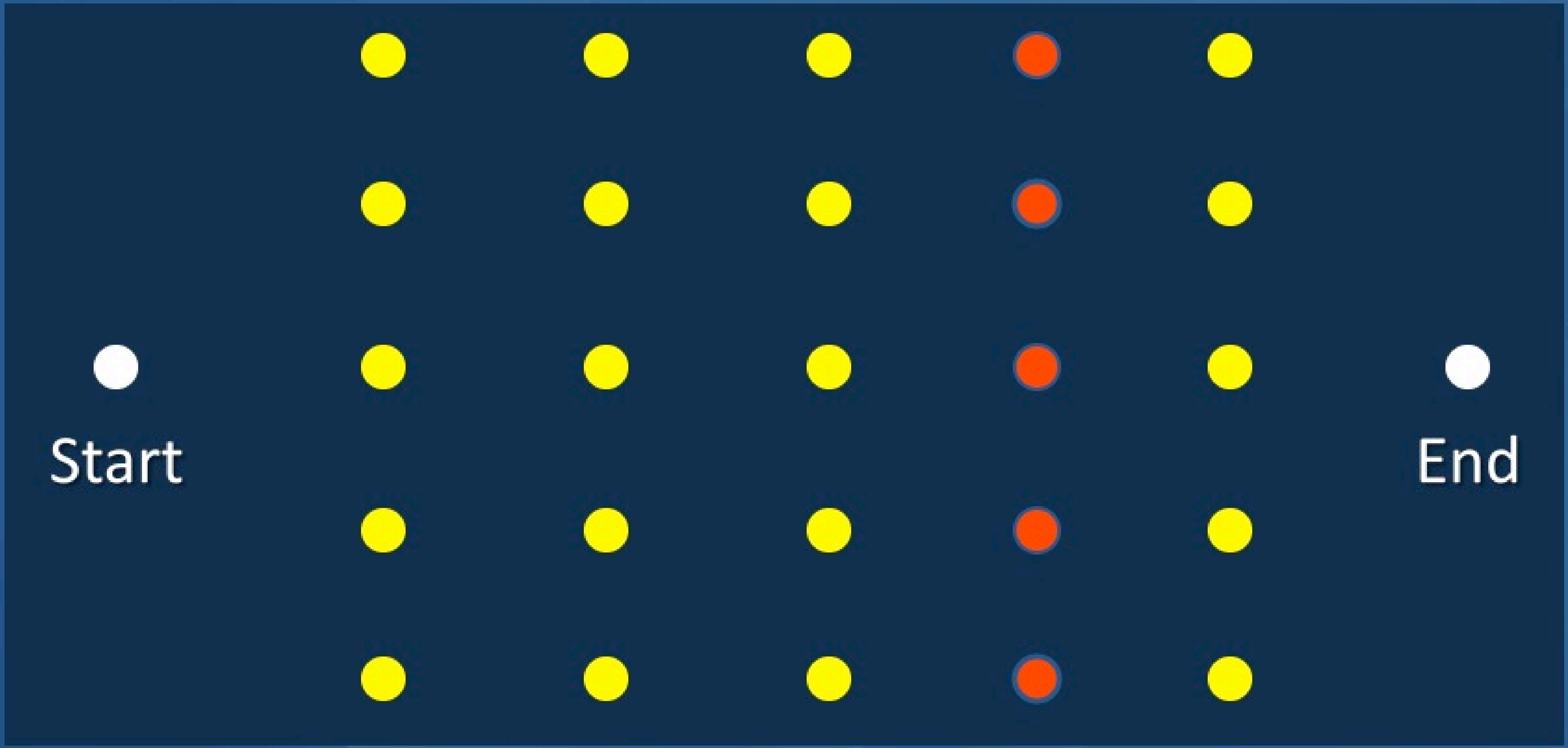
Cost based: every action, collision has a cost.

Minimize expected cost.

Routing of "intelligent" vehicles

R2D2 can identify obstacles inside a finite visual horizon.

Goal: Come up with an **optimum** route tracing scheme that combines prior and sequential information.



Discrete Space and Time Obstacles: ● Vehicle: ●

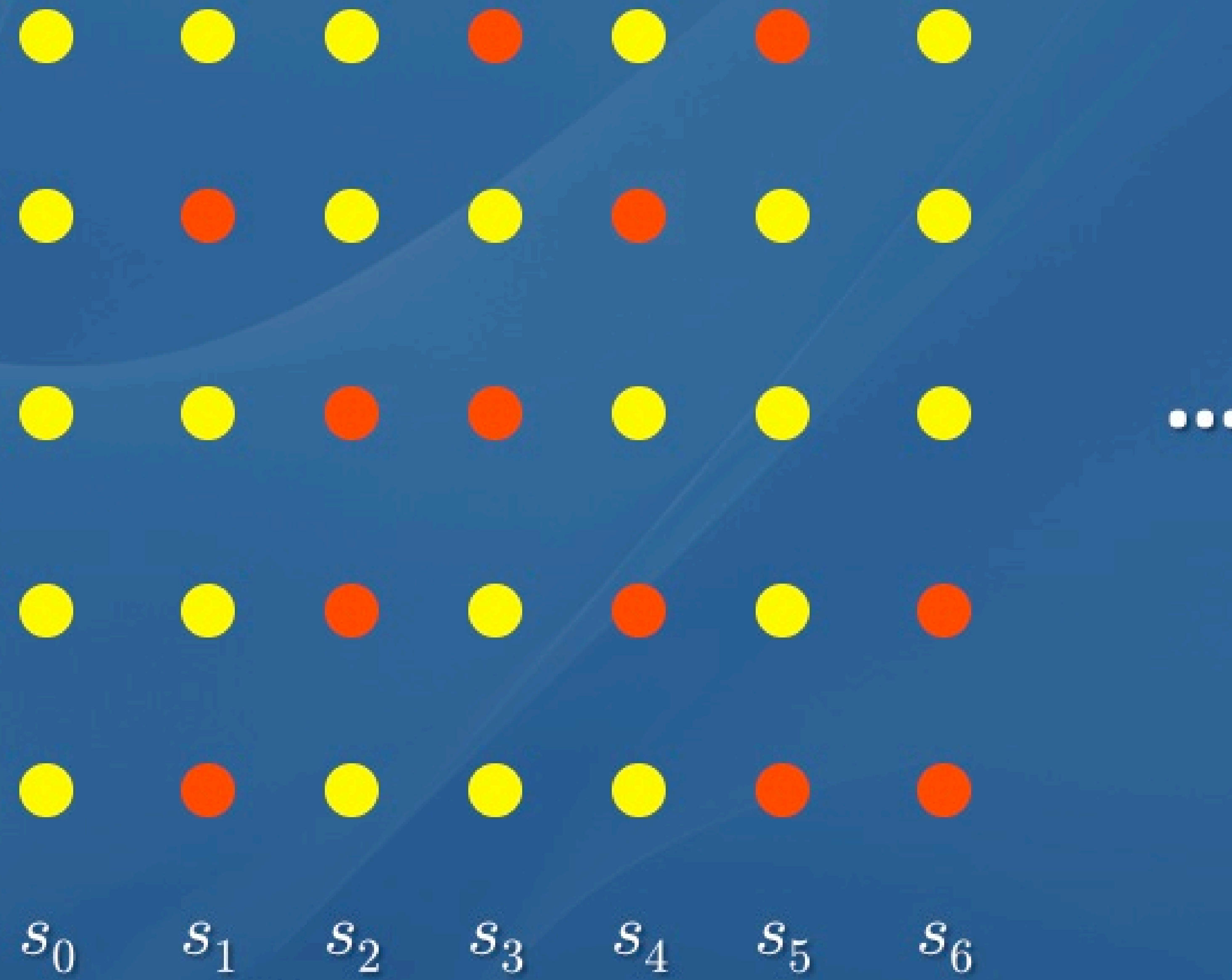
Obstacles:

Can move freely only **up** and **down**.

The dynamics between different columns are statistically independent. **Markovian model for each column.**

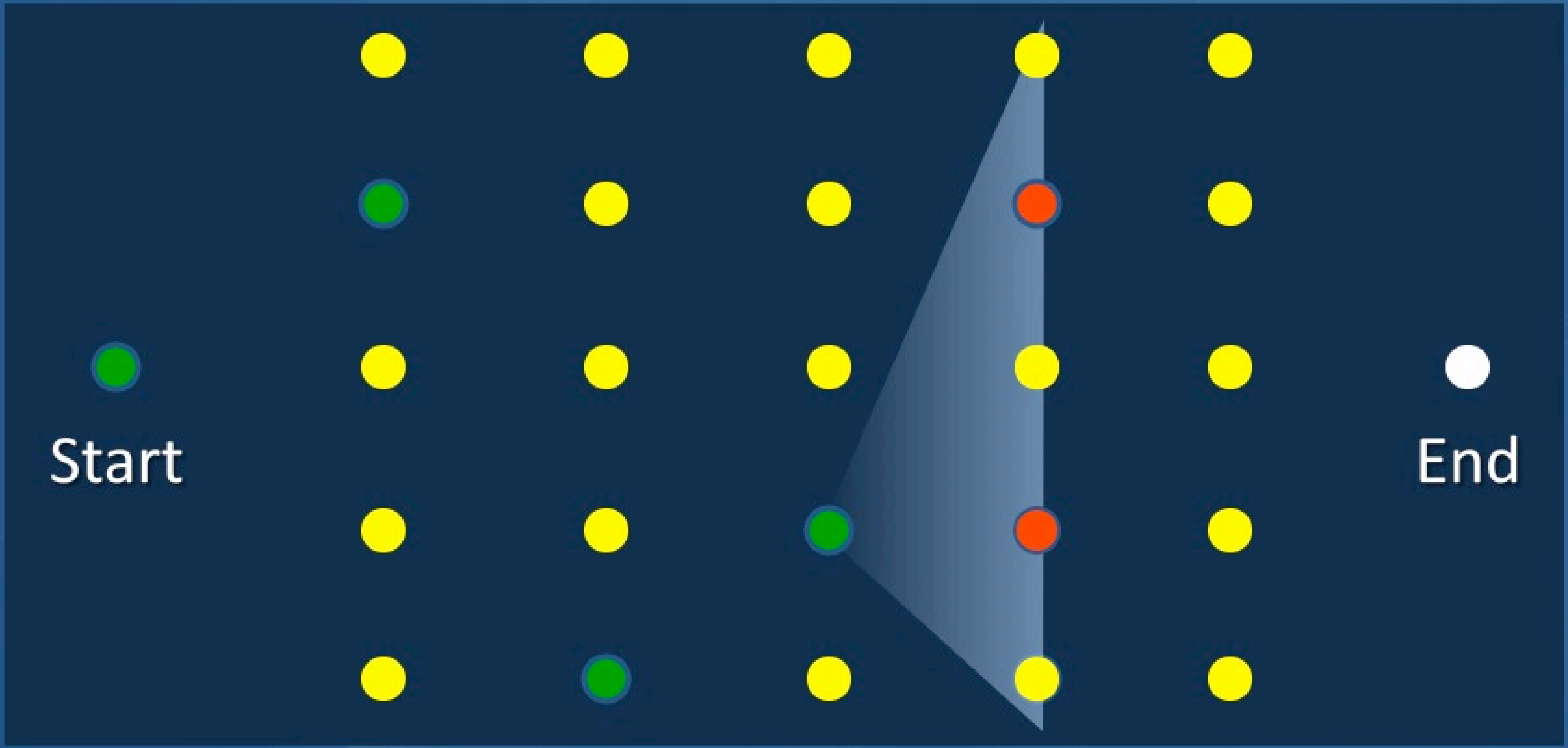
Consider the l-th column

The obstacle geometries constitute the possible states of the Markov process.



We assign transition probabilities: $\mathsf{P}_l(s_j | s_i)$ and compute the stationary probabilities $\pi_l(s_j)$:

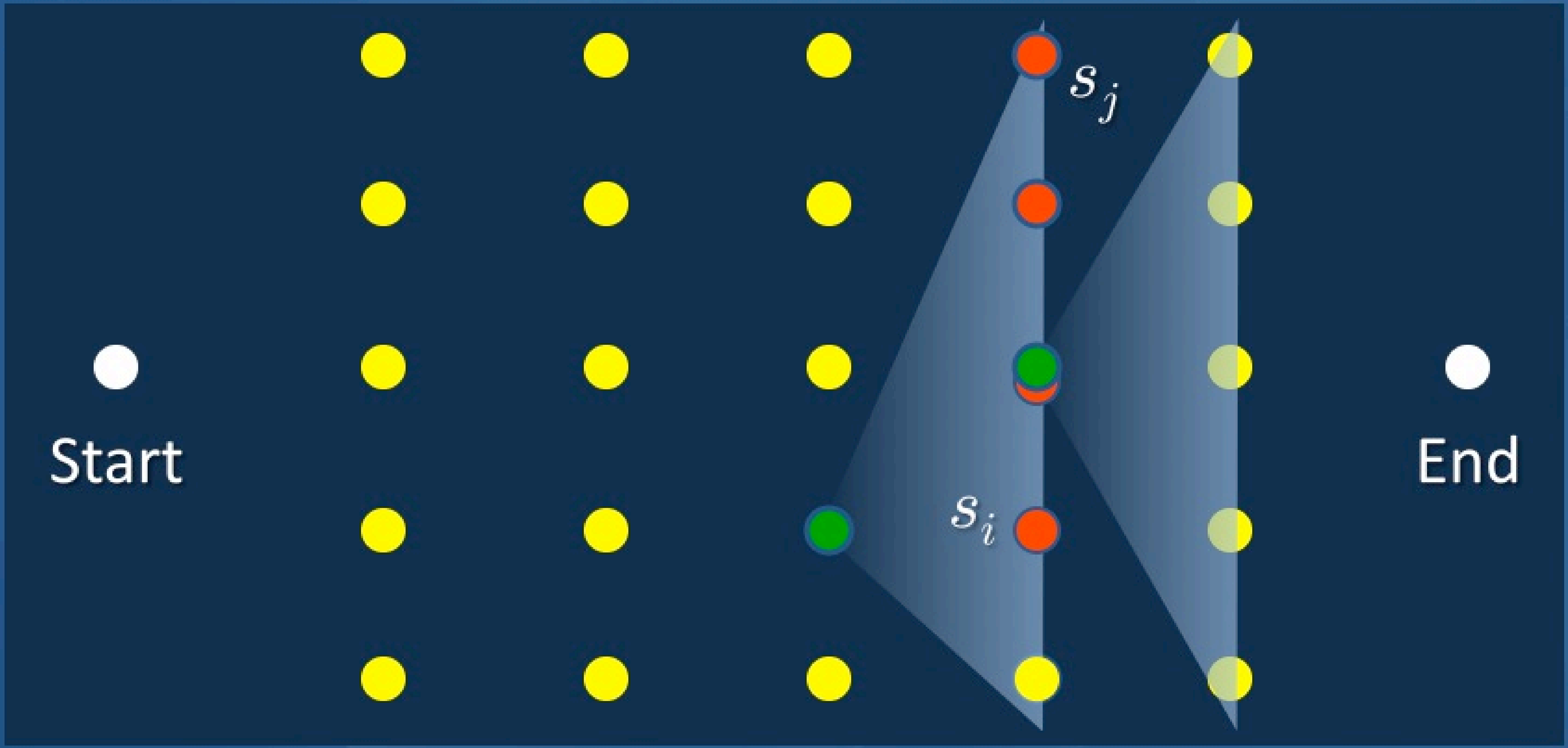
$$\pi_l(s_j) = \sum_{s_i} \mathsf{P}_l(s_j|s_i)\pi_l(s_i)$$



Vehicle:

Is allowed either to **wait** at the current node or **move forward to any node** of the **next** column.

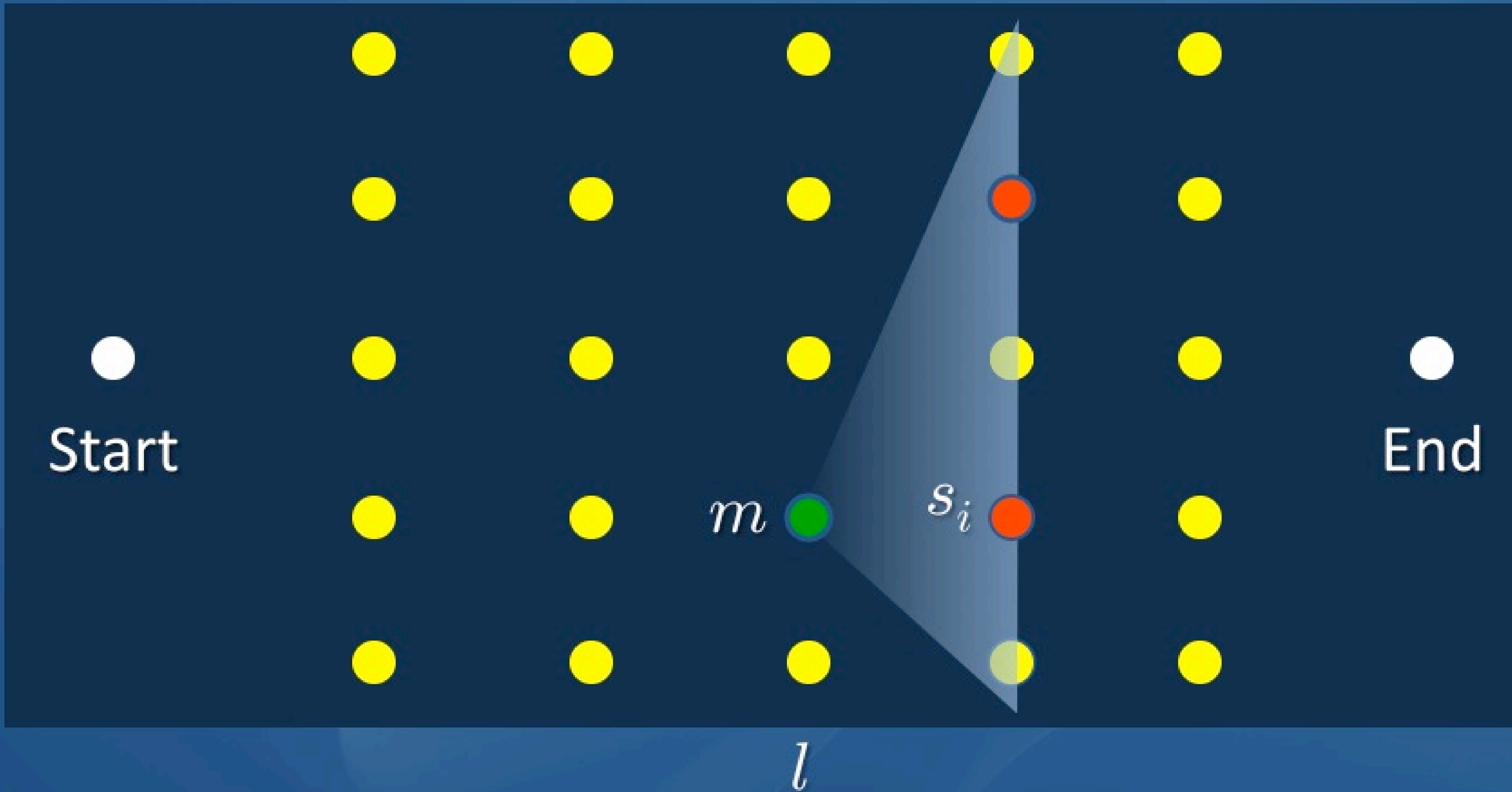
Its visual horizon extends to the next column and **recognizes the specific obstacle geometry.**



The vehicle observes state s_i of the next column.

While the vehicle moves to the next column, **the state of the next column changes to s_j**

This can result in collision



Vehicle is at node m of column l and observes state s_i of column l+1

What should the next action be?

For each column we must provide an Action Table:

Action Table for Column l

Node \ State	s_0	s_1	s_2	...
1	1	w	3	w
2	1	w	5	6
3	3	w	4	w
4	4	4	w	w
...				

These tables must be selected **optimally!!**

A cost based approach

To every action and event we assign a cost:

- **Displacement cost:**
  Moving from node m of column l
  to node n of column $l+1$, has a cost $c_l(m,n)$.
  Consider $c_l(m,n)=c_D$.
- **Waiting cost:**
  Waiting at node m of column l has a cost $c_l(m)$.
  Consider $c_l(m)=c_W$.
- **Collision cost:**
  Colliding at node n of column l has a cost $\underline{c}_l(n)$.
  Consider $\underline{c}_l(n)=c_C$.

To each collection of Action Tables there corresponds an **average cost**.

The goal is to find the Action Tables that **Minimize the Average Cost.**

These problems are conventionally solved with stochastic optimization techniques and in particular with the help of

Stochastic Dynamic Programming

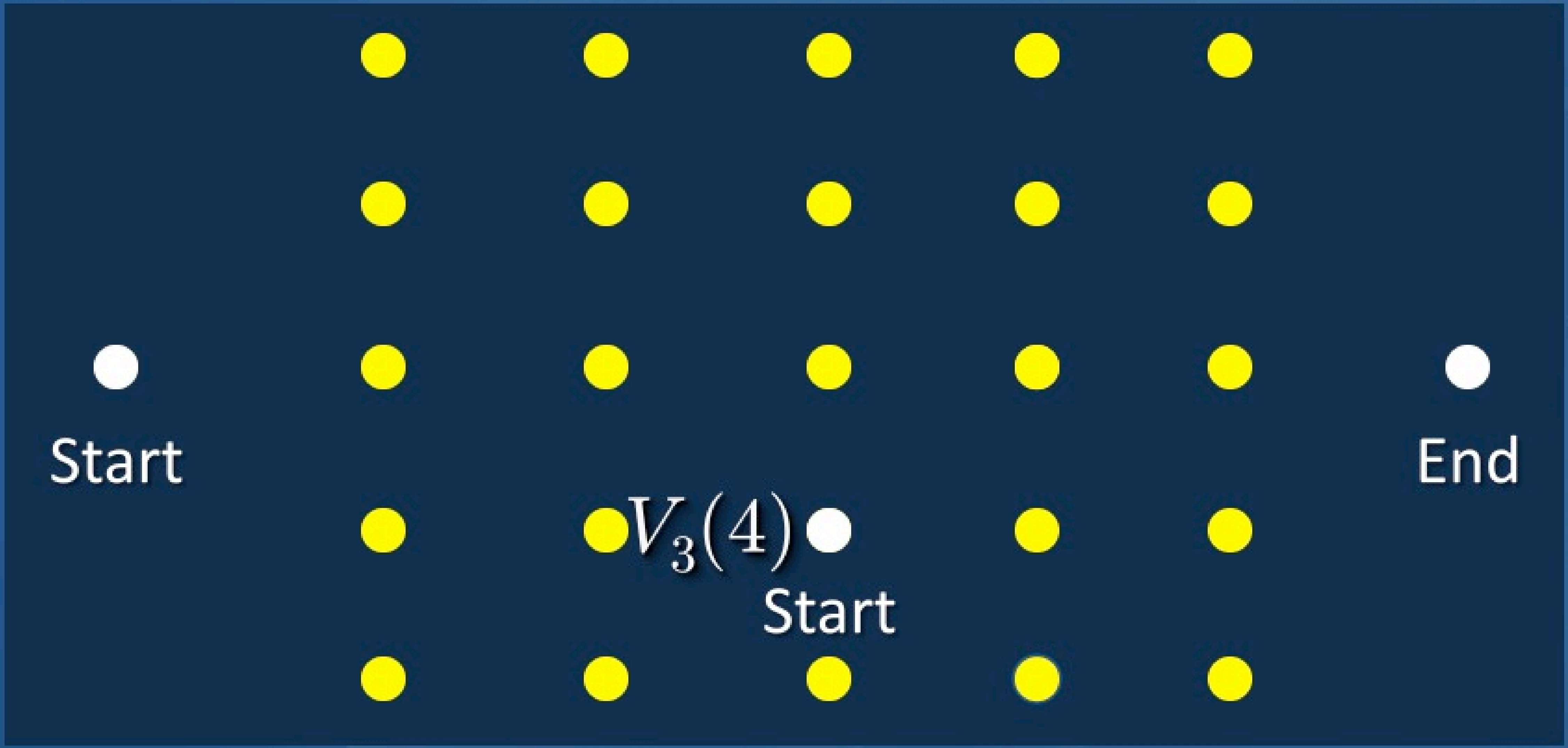
Optimal solution for "impatient" vehicles

No waiting is allowed!!

At every time instant the vehicle moves forward to the next column.

The optimum solution will be obtained by construction.

$V_l(m)$: **optimum average residual cost** from node m, column l, till the end.



$V_0(1)$ is the optimum average cost of the original problem.

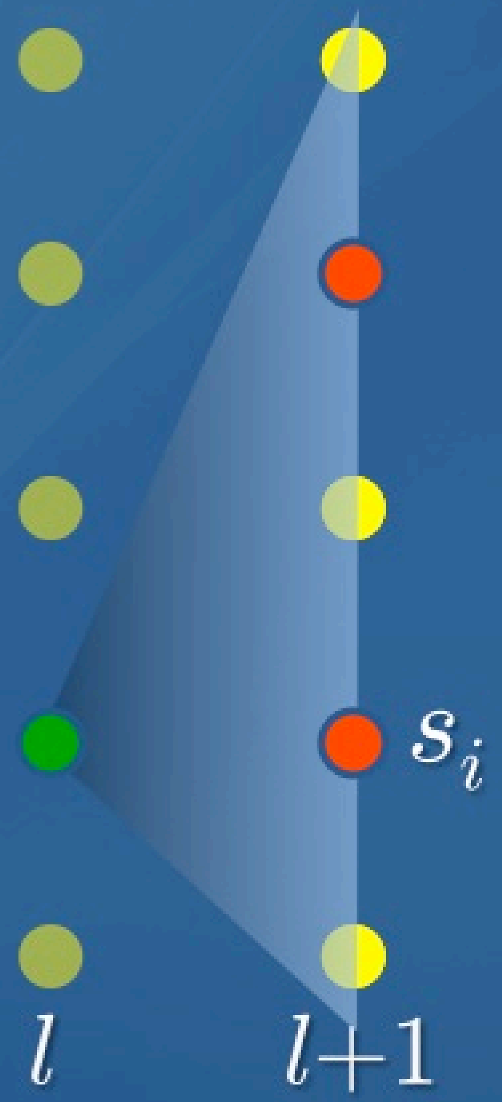
Key step in defining the optimum Action Tables is the determination of the (backward) evolution of $V_l(m)$.

$$V_N(m) = c_D$$

$$V_N(m) = c_D$$

Assume available $V_{l+1}(n)$ for any node n in column $l+1$.

We will then compute $V_l(m)$ for any node m in column l.



- Vehicle from node m can move to ANY node n of the next column.
- Vehicle when at node m, can observe state s_i of the next column.
- State s_i can change to ANY state s_j.

$V_l(m)$: optimum average residual cost **(goal)**.

$V_l(m\,|\,s_i)$: optimum average residual cost when state s_i is observed.

$V_l(m\,|\,s_i,n)$: optimum average residual cost when state s_i is observed and the vehicle decides to move to node n of the next column.

$$V_l(m|s_i,n) = c_D + c_C \sum_{s_j \ni n} \mathsf{P}_{l+1}(s_j|s_i) + V_{l+1}(n)$$

Best displacement when at m and observe s_i:

$$V_l(m|s_i) = \min_n V_l(m|s_i,n)$$

$$n_{\text{op}} = \arg\min_n V_l(m|s_i,n)$$

Action Table for Column l

Node \ State	...	...	s_i	...
⋮	...	...	⋮	...
m	...	...	n_{op}	...
⋮	...	...	⋮	...

By computing $V_l(m\,|\,s_i)$ from $V_l(m\,|\,s_i,n)$ we construct the **Optimum Action Tables**.

$$V_l(m) = \sum_{s_i} \pi_{l+1}(s_i) V_l(m|s_i)$$

With visual horizon:

$$V_l(m|s_i,n) = c_D + c_C \sum_{s_j \ni n} \mathsf{P}_{l+1}(s_j|s_i) + V_{l+1}(n)$$

$$V_l(m|s_i) = \min_n V_l(m|s_i,n)$$

$$V_l(m) = \sum_{s_i} \pi_{l+1}(s_i) V_l(m|s_i)$$

Column: l
Node: m
State: s_i
→
Column: $l+1$
Node: n_{op}

Without visual horizon:

$$V_l(m|n) = c_D + c_C \sum_{s_j \ni n} \pi_{l+1}(s_j) + V_{l+1}(n)$$

$$V_l(m) = \min_n V_l(m|n)$$

Column: l
Node: m
→
Column: $l+1$
Node: n_{op}

Optimal solution for "patient" vehicles

Waiting is allowed!!

We also assume that **no collision is possible** when the vehicle waits at a node.

$$V_l(m|s_i,n) = c_D + c_c \sum_{s_j \ni n} \mathsf{P}_{l+1}(s_j|s_i) + V_{l+1}(m)$$

$$V_l(m|w) = c_W + V_l(m)$$

Nonlinear equation

$$V_l(m|s_i) = \min\left\{\min_n V_l(m|s_i,n), V_l(m|w)\right\}$$

$$V_l(m) = \sum_{s_i} \pi_{l+1}(s_i) V_l(m|s_i)$$

Define: $\tilde{V}_l(m|s_i) = \min_n V_l(m|s_i, n)$

Without loss of generality assume the ordering:

$$\tilde{V}_l(m|s_0) \leq \tilde{V}_l(m|s_1) \leq \cdots \leq \tilde{V}_l(m|s_L)$$

For $0 \leqslant n \leqslant L$, define the increasing sequence:

$$F_n = \sum_{i=0}^{n-1} [\tilde{V}_l(m|s_n) - \tilde{V}_l(m|s_i)]\pi_{l+1}(s_i)$$

Let $0 \leqslant K \leqslant L$ be the largest integer satisfying $c_W \geqslant F_K$

$$V_l(m) = \frac{c_W + \sum_{i=0}^{K} [\tilde{V}_l(m|s_i) - c_W]\pi_{l+1}(s_i)}{\sum_{i=0}^{K} \pi_{l+1}(s_i)}$$

Extensions

- Collisions may occur during waiting

$$V_l(m|w) = c_W + V_l(m)$$

$$V_l(m|w) = c_W + c_C \mathsf{P}_l(\text{Collision}) + V_l(m)$$

$$V_l(m|w) = c_W + c_C \sum_{s_j^l \ni m} \pi_l(s_j^l) + V_l(m)$$

$$V_l(m|s_k^l, w) = c_W + c_C \sum_{s_j^l \ni m} \mathsf{P}_l(s_j^l|s_k^l) + V_l(m)$$

- Vehicle can have a larger visual horizon
- State can contain information other than location like: speed, acceleration, etc.
- Obstacles and Vehicle can move in any direction (even backwards)
- On line estimation (identification) of state elements (for example position, speed and acceleration)