Plan for Today

Nonparametric RL

- "Nonparametric" function approximation
- Strong guarantees across:
  Sample complexity, *space complexity*, *storage complexity*

Hindsight Learning

- Exogenous MDPs as model for OR problems
- Use of *Hindsight Planning* oracle for algorithm design
- Empirical results in VM allocation with Microsoft Azure

Tree-Partitions

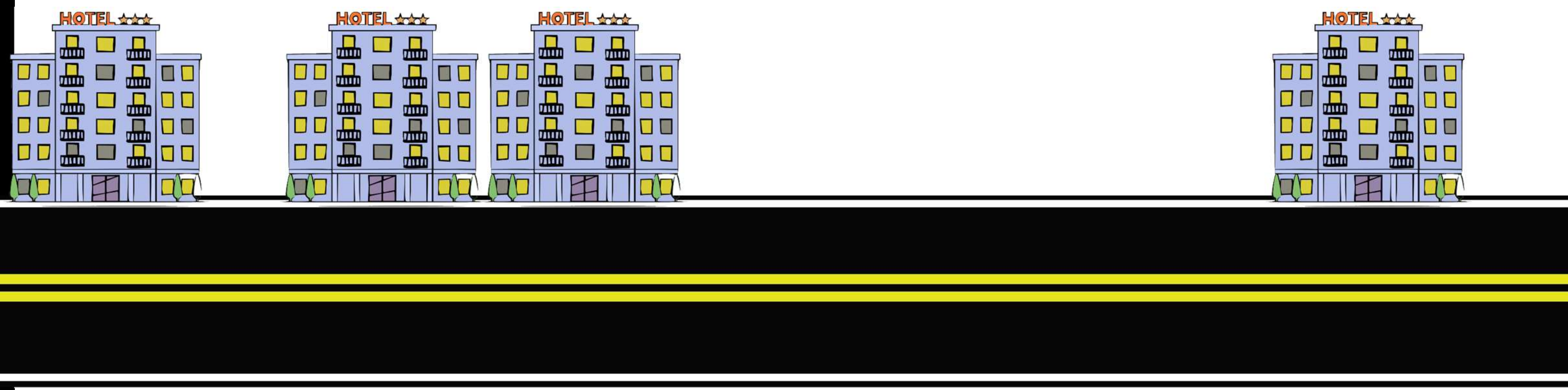
- Implement tree-based adaptive discretization from nonparametric RL algorithms
- Use ORSuite to test on "continuous Ambulance routing"

Hindsight Planning for Exo-MDPs

- Use ORSuite model for revenue management and pricing (an example of an Exo-MDP)
- Implement Bayes Selector
- Use ORSuite to run simulations to compare performance against tabular algorithms

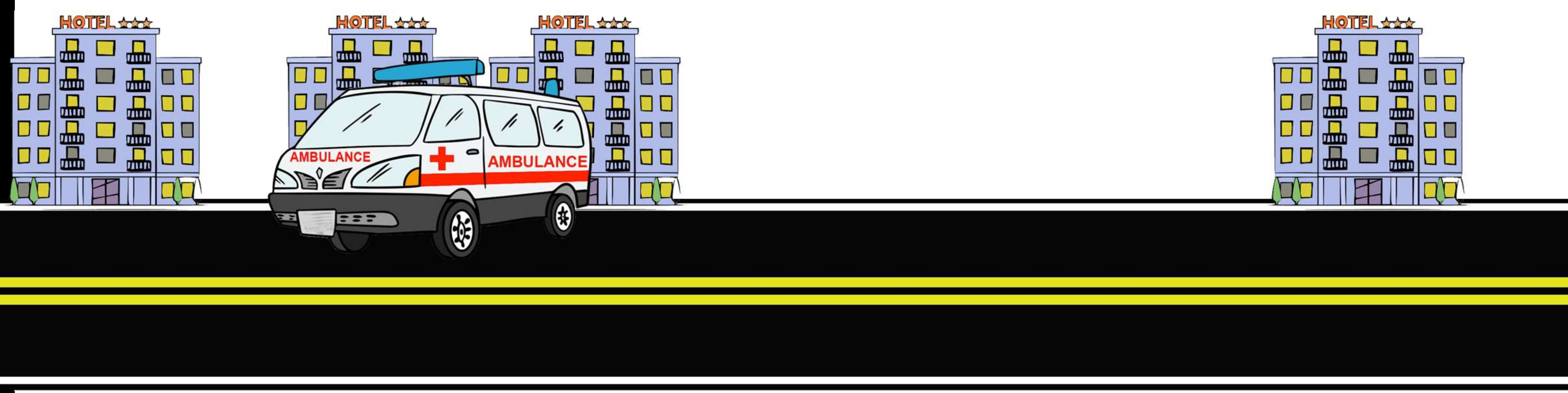
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution



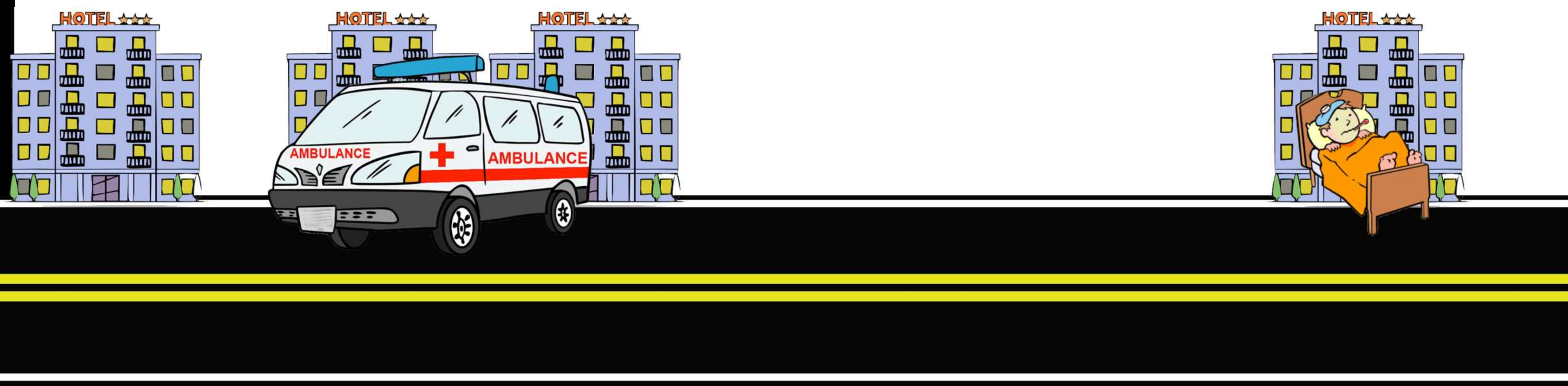
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution



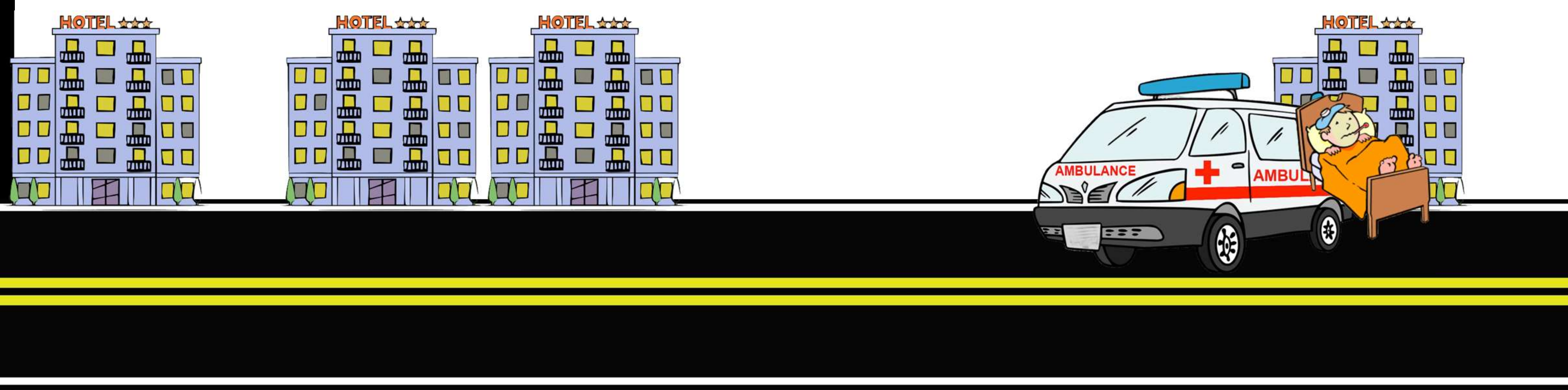
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution



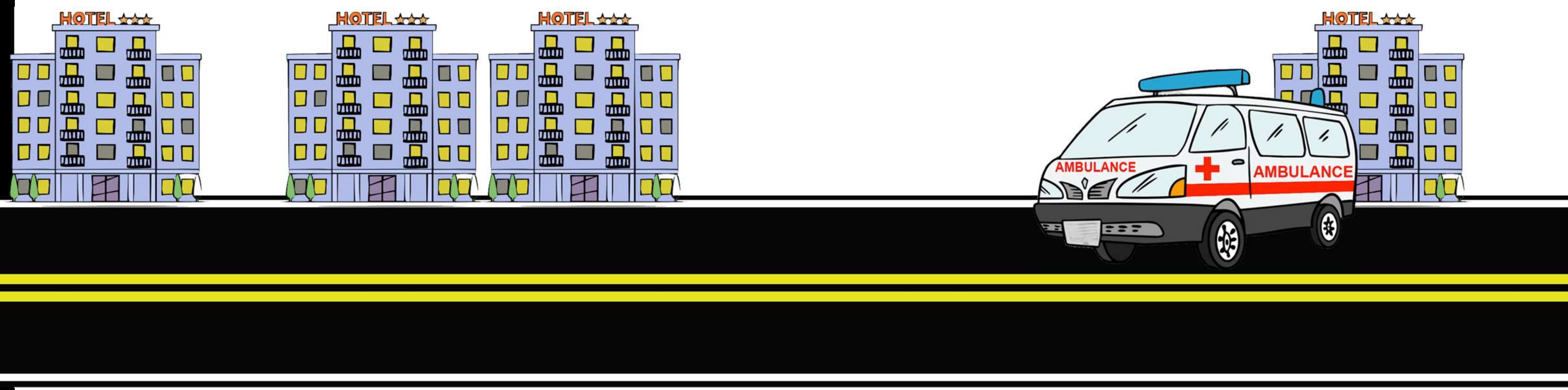
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution



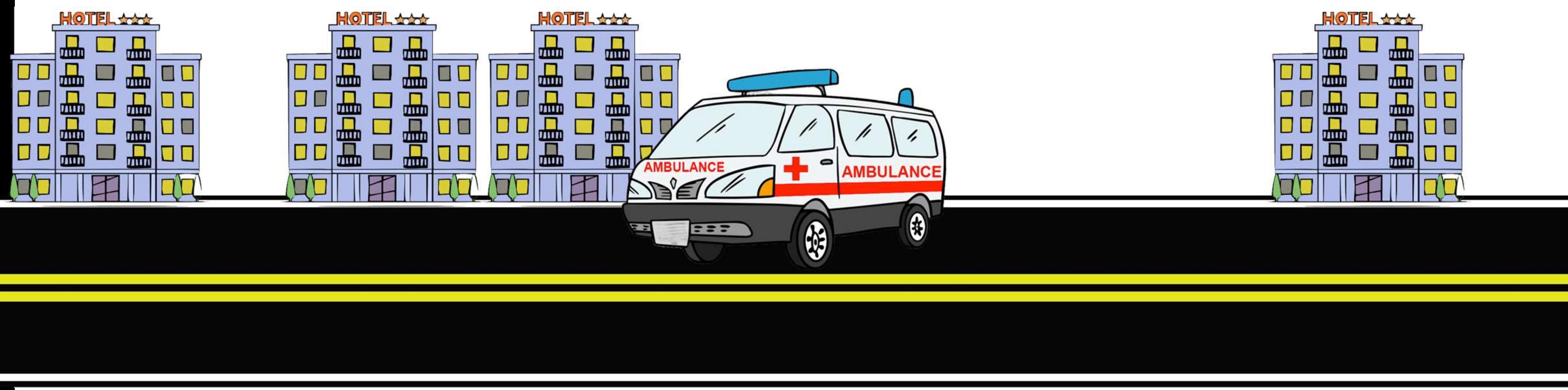
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution



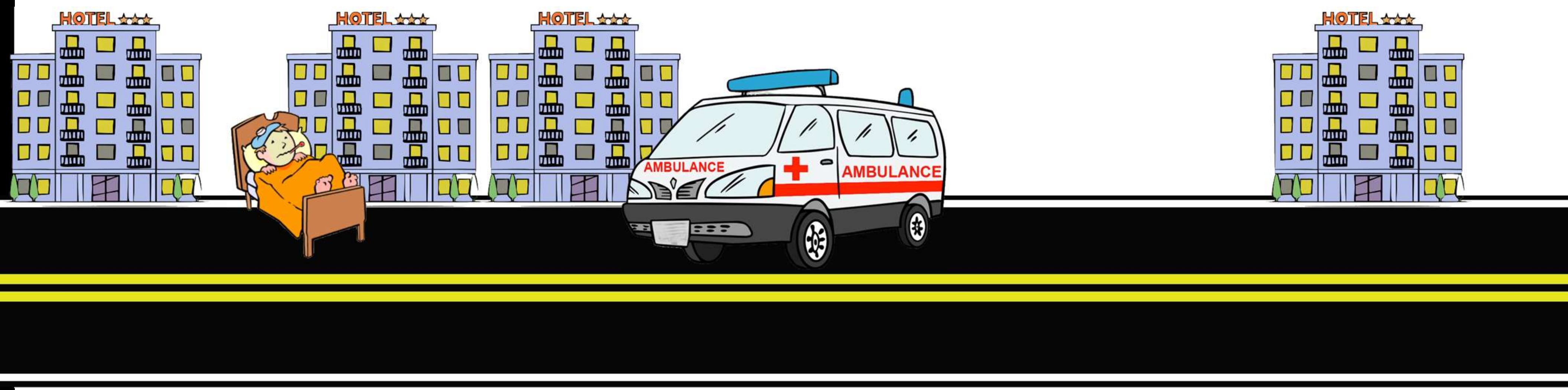
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution



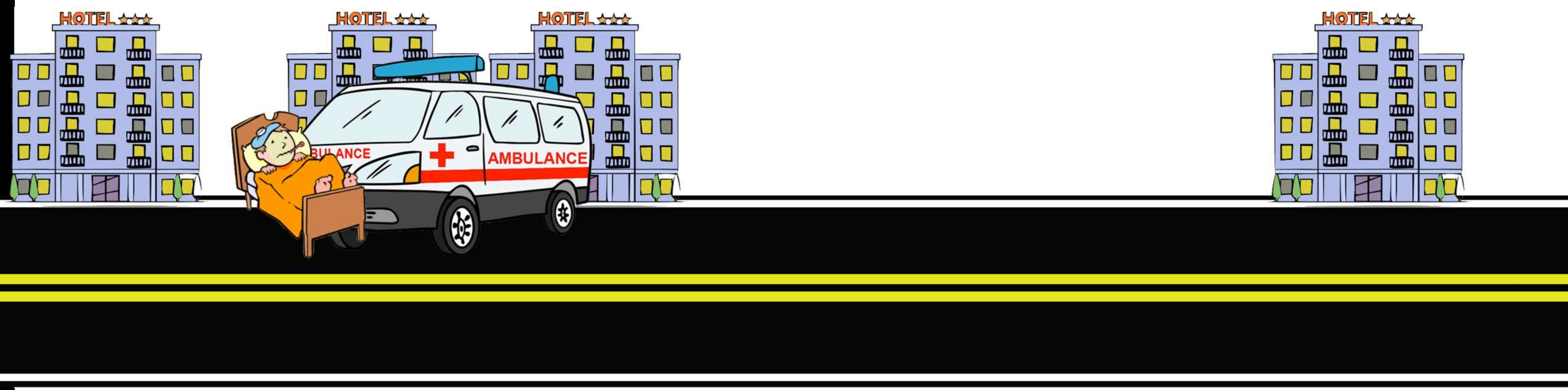
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution



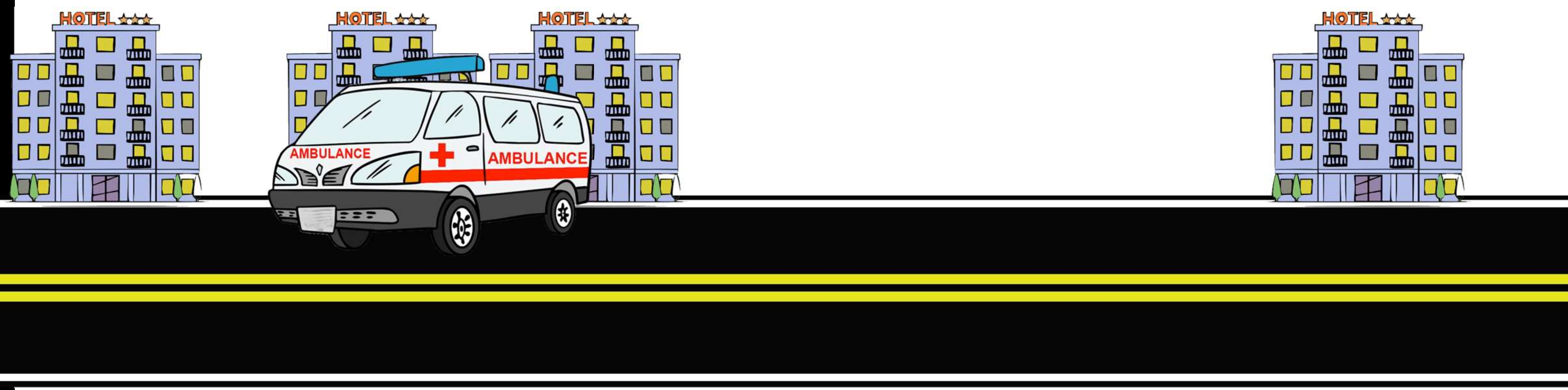
Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution

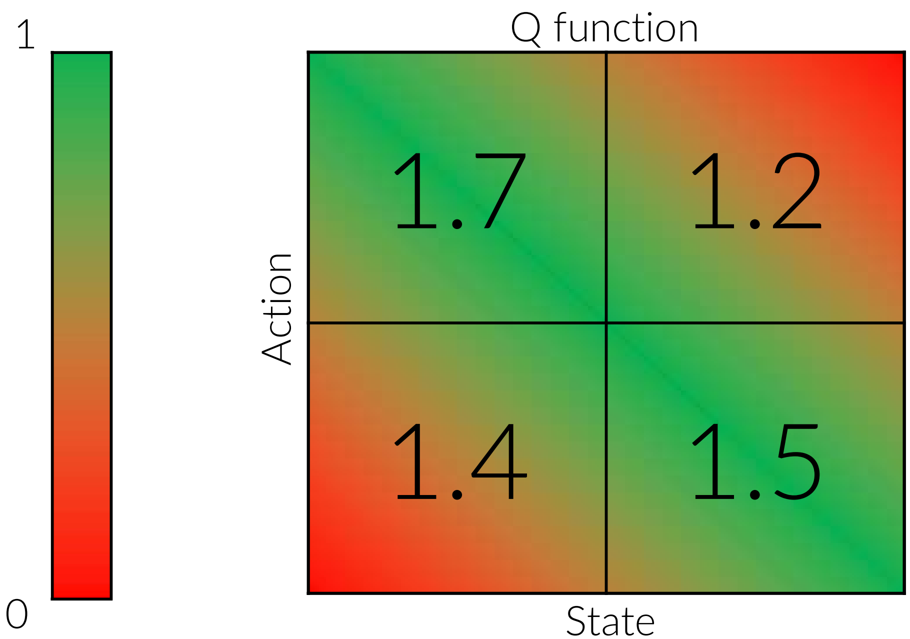


Ambulance Routing

- Operator decides location to station ambulance, paying a transportation cost
- Random request realized, ambulance pays cost for travel delay to serve patient
- Goal: learn policy which minimizes costs w/o knowledge of arrival distribution

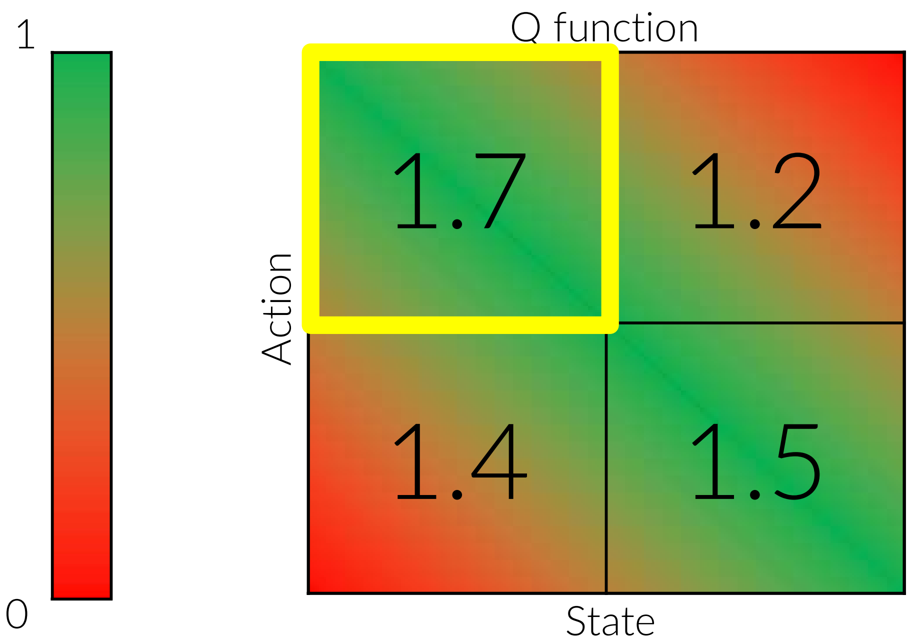


Adaptive Discretization



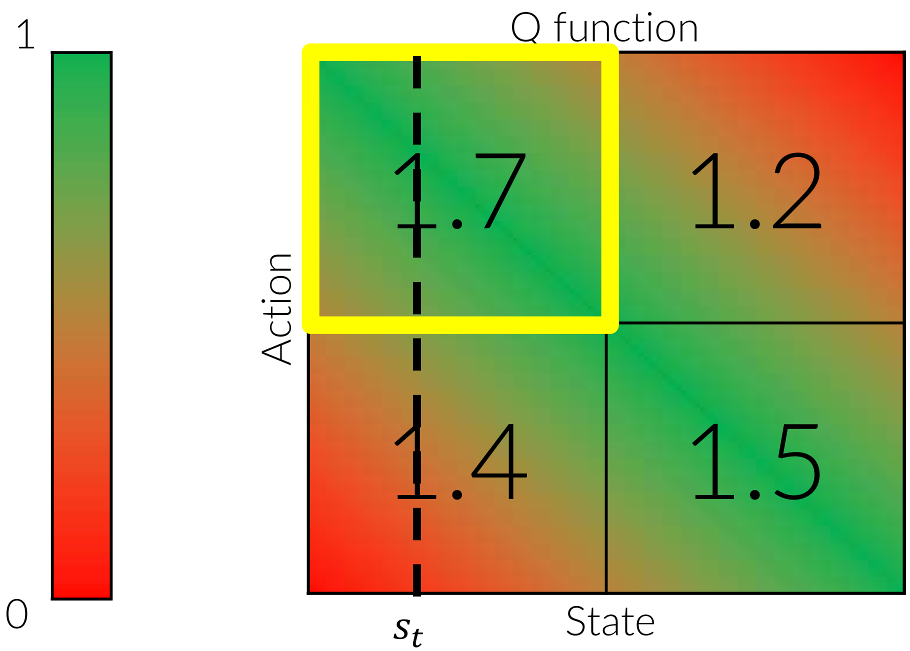
- *(Adaptive Partition):* Begin with coarse discretization of space
- *(Generate Estimates):* Maintain UCB estimates of Q function across regions
- *(Selection Rule):* Select "relevant" region that maximizes UCB for state

Adaptive Discretization



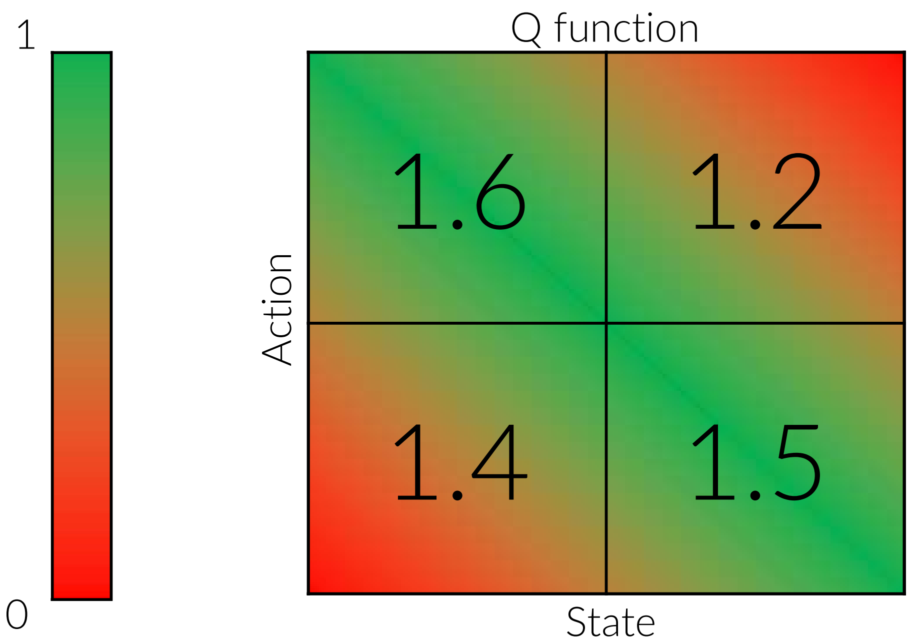
- *(Adaptive Partition)*: Begin with coarse discretization of space
- *(Generate Estimates)*: Maintain UCB estimates of Q function across regions
- *(Selection Rule)*: Select "relevant" region that maximizes UCB for state

Adaptive Discretization



- *(Adaptive Partition):* Begin with coarse discretization of space
- *(Generate Estimates):* Maintain UCB estimates of Q function across regions
- *(Selection Rule):* Select "relevant" region that maximizes UCB for state

Adaptive Discretization



- *(Update Estimates)*: Update estimates with newly collected data:

$$(s_h, a_h, s_{h+1}, r_h)$$

Model-Free:

$$\overline{Q}_h(B_{sel}) = (1-\alpha_t)\overline{Q}_h(B_{sel}) + \alpha_t\left(r_h + \textsc{Conf}(B_{sel}) + \max_{x_{h+1}\in B} \overline{Q}_{h+1}(B)\right)$$

Empirical Bellman Optimality Equation

Model-Based: More complicated…as we require estimates of:

Average Rewards

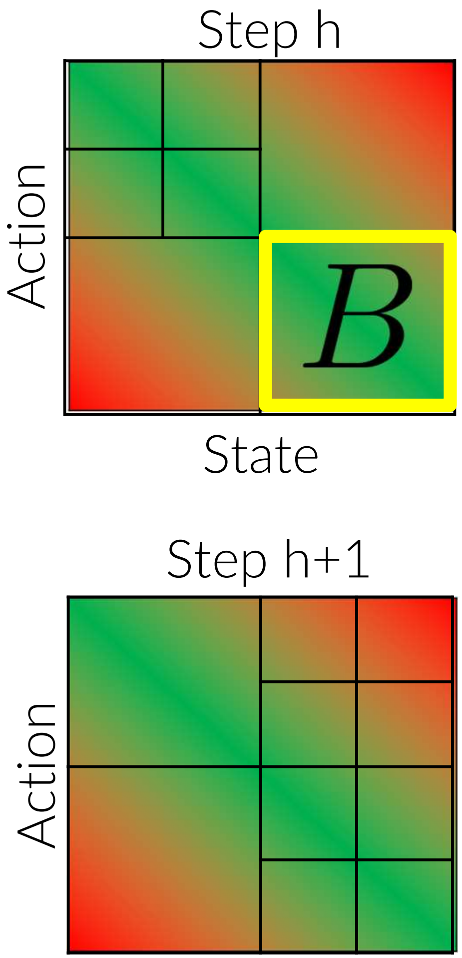
Easy – estimate as average reward of samples from region

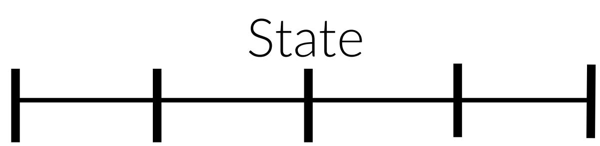
Transition Distribution

Tough – domain is region, range is state space

$$\overline{T}_h(\cdot \mid B)$$

Adaptive Discretization

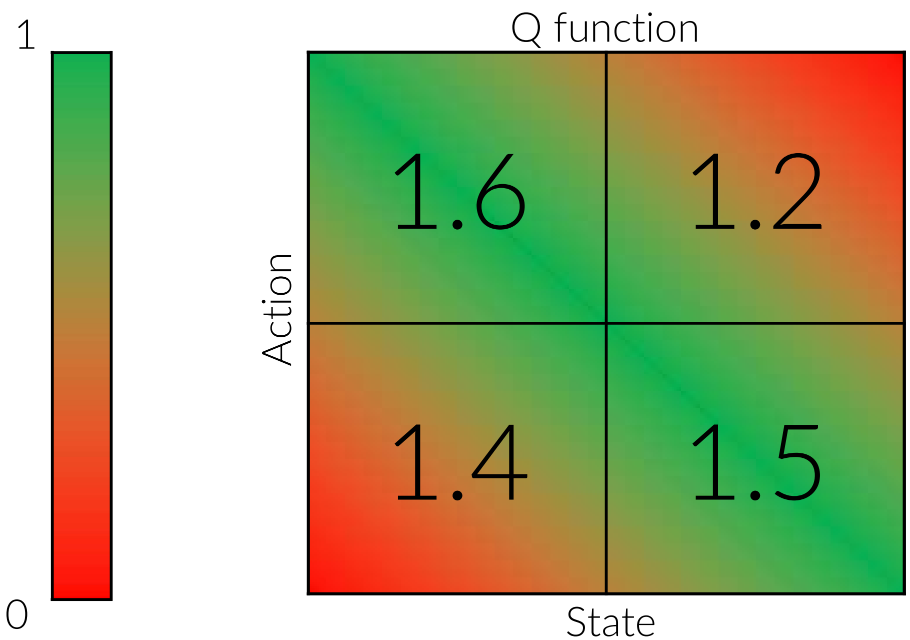




Uniform State Discretization

- Estimate transition function over a uniform-discretization of the state space up to the precision of a ball
- Ensures accuracy of the estimate is proportional to the diameter
- Limits storage complexity for the estimates

Adaptive Discretization



- *(Update Estimates)*: Update estimates with newly collected data:

$$(s_h, a_h, s_{h+1}, r_h)$$

Model-Based:

$$\overline{Q}_h(B) = \overline{r}_h(B) + \overline{\mathbb{E}}(\max_{X' \in B} \overline{Q}_{h+1}(B) \mid B) + \textsc{Conf}(B)$$

Plan for Today

Tree-Partitions

- Implement tree-based adaptive discretization from nonparametric RL algorithms.
  - Essentially a d-ary tree
  - Implement argmax over sub-tree
- Update Q estimates

Run Experiments

- Use experiment instrumentation as part of ORSuite to run experiment of AdaMB, AdaQL on ambulance routing models under various arrival distributions:
  - beta distributed arrivals
  - uniform arrivals

References

https://github.com/seanrsinclair/RLinOperations