Introduction:

In this report, I present my exploration of formulating real-world scenarios as **Finite-horizon** Markov Decision Processes (**MDPs**) and solving them using Dynamic Programming (**DP**) techniques. The primary focus is on 2 following urban mobility case studies:

1. **Safe Navigation of an Autonomous Vehicle**, which involves decision-making for safe and efficient road travel through urban environments with potential obstacles and varying road conditions. The description for this scenario mentioned that "*it has a clear view of everything*", which means agents always know the true state (no hidden information, unlike POMDPs). Moreover, since it also mentions that "*choices made by the vehicle's system can lead to safe drives or potential issues*", I will incorporate stochastic transitions here for probabilistic outcomes (e.g., accelerating might hit an obstacle with 0.4 probability), teaching expectation maximization.
2. **Urban Traffic Management for a Smart City**, which centers on optimizing traffic flow amid congestion, pedestrians, events, and safety in dynamic urban environments. The description for this scenario mentioned that "*it knows that each choice will have a clear result, without any uncertainties*". Therefore, I employed deterministic transitions here to align with that emphasis for predictable outcomes from actions.

I will focus on modelling these scenarios as **Finite-horizon MDPs** to reflect episodic tasks, where decisions are bounded by a time **horizon** *H=100* with clear termination, such as completing a commute or resolving rush-hour congestion. Both scenarios will be solved using **Value Iteration (VI)** and **Policy Iteration (PI)** algorithms, which are core **DP** methods for computing optimal policies:

- For **Finite horizon MDPs**, **Value Iteration** iteratively updates the value function $v_h(s)$ backward from the terminal step using the Bellman optimality equation backward over *h=1* to *H*: $v_h(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma v_{h-1}(s')]$, where *γ*=1.0 for undiscounted **Finite horizons**, converging to the optimal V* exactly after H steps.
- **Policy Iteration**, initialized stochastically, alternates policy evaluation (using expectation equation to compute $v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} P(s'|s,a)[r + \gamma v_\pi(s')]$ for a policy π) and improvement (greedily updating $\pi(s) = \arg\max_a \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma v_\pi(s')]$) until policy stability or reaching the end of defined **horizons** for broad exploration. This algorithm often converges faster in fewer iterations despite more computation per step.

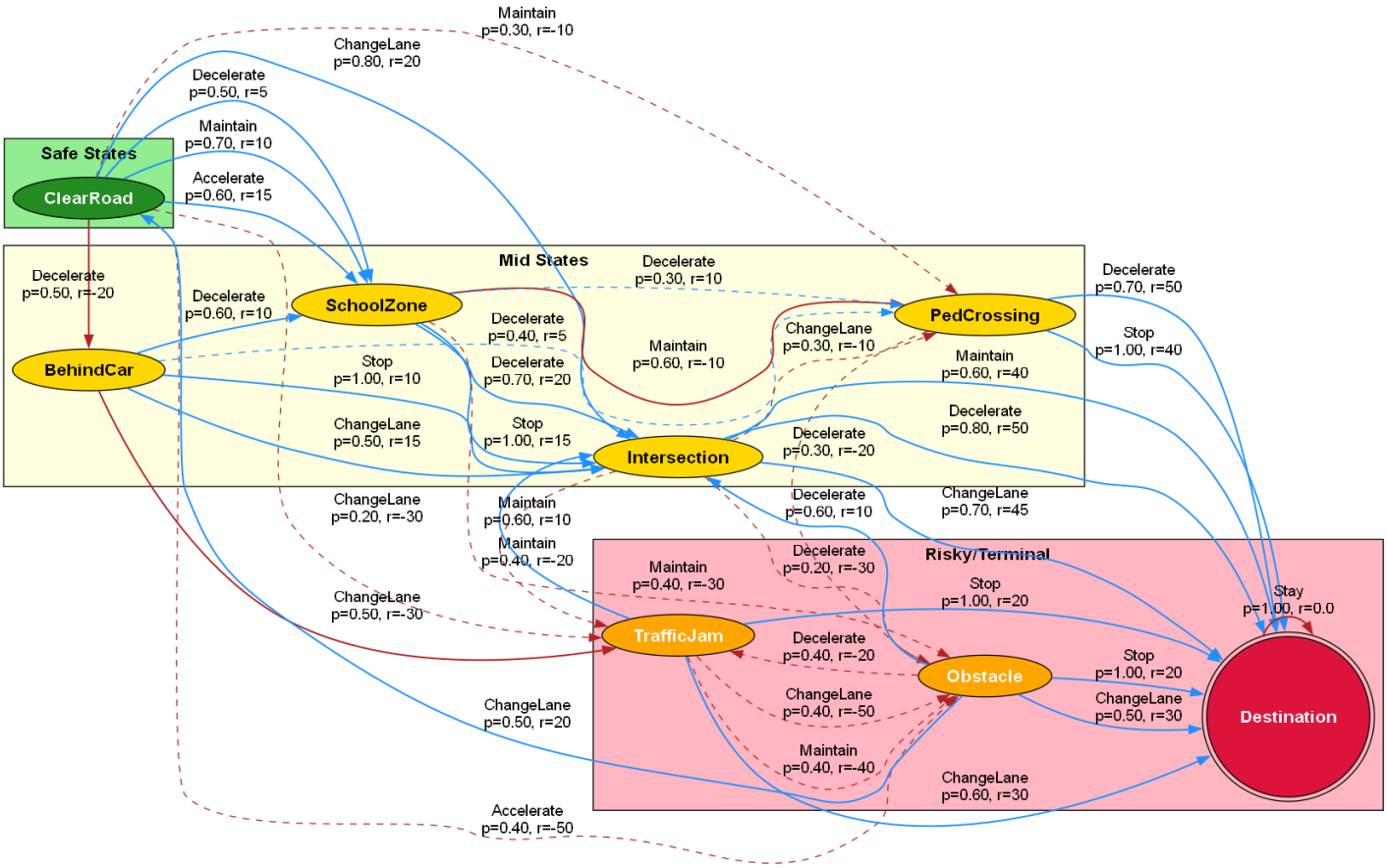
The report will begin by detailing **MDP** formulations for each scenario, including the definition of at least **8** states, actions, transition probabilities, and rewards, along with rationales for my choices. Subsequently, I will compare both **VI** and **PI** algorithms applied to solve these **MDPs** on metrics like convergence time, optimal policies, value functions, and average rewards over 1000 simulated episodes.

For deeper insights, alongside **Finite-horizon** settings, I also explore **Infinite-horizon** variants (with discounted *γ*=0.9), using convergence threshold θ=0.001. This dual approach allows for insights into convergence behaviours like how many steps would be needed without a fixed **horizon**, highlighting trade-offs between exact backward computation in **Finite horizons** and iterative approximation in **Infinite** ones. These results are also then discussed to evaluate the algorithms' strengths and limitations in these contexts.

Markov Decision Process Formulation:

In my implementation, each **MDP** is designed to capture **episodic** nature of the problem, with a horizon *H=100* to allow sufficient steps for convergence while bounding the task. **States** are chosen here to represent key situational conditions (e.g., traffic levels or road obstacles), **actions** are state-dependent to capture context-specific feasibility (e.g., adjust speeds/signals), **transitions** model environmental responses (stochastic/deterministic), and **rewards** encourage optimal outcomes (positive for progress/safety, negative for risks/delays).

Scenario 1: Safe Navigation of an Autonomous Vehicle



This scenario models an autonomous vehicle's decision-making during a commute, deciding actions at discrete time steps to reach a destination safely amid potential obstacles, traffic, and pedestrians. Its **MDP** is stochastic in transitions to capture unpredictability (e.g., probabilistic obstacles despite full observability). The **Finite horizon** bounds the episode to prevent indefinite driving, simulating a time-limited trip. Here, I used **8 states** to represent progressive road situations, chosen to cover a range from low to high-risk conditions, allowing the vehicle to encounter varied challenges without cycles (transitions push toward terminals) and learn adaptive safety. Each state is mutually exclusive and collectively exhaustive for this scenario's scope, with terminals absorbing to end episodes. This design enables multi-step planning, illustrating how state granularity affects value computation.

1. **ClearRoad**: The vehicle is on an open stretch with no immediate issues. This serves as a low-risk baseline, often an entry point, to highlight rewards for efficient progress.
2. **BehindCar**: The vehicle is tailing a slower vehicle, risking delays or mild congestion. This models common traffic delays. Stochastic transitions to riskier states emphasize the need

for actions like changing lanes, demonstrating how negative rewards discourage passivity.

3. **SchoolZone**: Approach or in an area with potential children/pedestrians, emphasizing caution: This is added for safety-critical zones and restricted actions (no acceleration).
4. **PedCrossing**: At a pedestrian crossing with possible walkers. This also heightens safety focus with limited actions to promote stopping or force slowdowns to avoid accidents.
5. **TrafficJam**: Heavy congestion causing slowdowns. This represents peak-hour bottlenecks or escalating issues from prior states, testing recovery strategies. Here, transitions to intersections/obstacles can model escalation, rationalizing rewards for diversion actions to optimize flow and teach multi-step foresight in **VI**.
6. **Intersection**: A busy junction with multiple flows. This captures decision points like traffic lights, central to urban navigation. Stochastic outcomes here simulate variable yields, with rewards favoring caution.
7. **Obstacle**: Encounter unexpected debris/construction. I will make this non-terminal for recovery, allowing "evade" actions. Stochastic transitions to safety or jams would rationalize negative rewards.
8. **Destination** (**terminal**, absorbing): Safe arrival/good end-goal. Zero ongoing reward post-arrival can teach termination, with high positive entry rewards (+50) incentivizing efficient paths, contrasting bad terminals like implied crashes in obstacles if not recovered.

I designed **state-dependent actions,** with 3-4 per state (**5 unique**: *Maintain, Accelerate, Decelerate, ChangeLane, Stop*) to reflect feasibility constraints, prioritizing safety (e.g., discouraging risky moves in vulnerable states) and avoiding irrelevant actions (e.g., no *Accelerate* in **SchoolZone** to prevent hazards).

1. In **ClearRoad**: *Maintain, Accelerate, Decelerate, ChangeLane*. There are full options for open roads. *Accelerate* can risk obstacles, teaching trade-offs via expected values.
2. In **BehindCar**: *Decelerate, ChangeLane, Stop*. No acceleration here to avoid collisions or prevent rear-ends. *ChangeLane* is for overtaking, with probabilities to jam rationalizing safety-first rewards.
3. In **SchoolZone**: *Maintain, Decelerate, Stop*. These are for safety-focused, no speeding. I exclude *Accelerate* for child safety and encourage *Decelerate* (+20 to intersection), showing constraint-driven optimality.
4. In **PedCrossing**: *Decelerate, Stop*. There are minimal actions for caution. *Stop* for guaranteed safety (+40 to destination) and *Decelerate* probability for simulating risk, illustrating probabilistic risk assessment.
5. In **TrafficJam**: *Maintain, ChangeLane, Stop*. These are for focus on clearance. *ChangeLane* to destination with probability can rationalize negative jam rewards to favor diversion.
6. In **Intersection**: *Maintain, Decelerate, ChangeLane*. These are for balanced flow. *Decelerate* will have highest reward (+50 to destination), teaching collision avoidance.
7. In **Obstacle**: *Decelerate, ChangeLane, Stop*. The state is evasive without *Accelerate*. I designed recovery options here with *ChangeLane* (+30 to destination) as the best, with probability to clear road, emphasizing adaptive policies.
8. In **Destination**: *Stay*. This is for absorption, so zero reward can ensure terminal value 0.

Transition probabilities are stochastic ($P(s'|s,a) < 1$ for multiple s'), with 2-3 possible next states per action and probabilities summing to 1, favoring positive outcomes for safe actions

(e.g., higher to **Destination** with *Decelerate*), while risky actions (e.g., *Accelerate*) have higher hazard probability (0.4 to **Obstacle**), encouraging cautious policy. They are designed forward to avoid cycles (no back to early states), ensuring acyclicity and **episodicity**: From early states to mid/late, eventually terminals, bounding potential loops via H.

1. From **ClearRoad**, *Maintain*: 0.7 to **SchoolZone**, 0.3 to **PedCrossing**. Demonstrate smooth flow with pedestrian risk. I assigned high probability to safe progress & low to pedestrians.

- *Accelerate*: 0.6 to **SchoolZone**, 0.4 to **Obstacle**. Riskier speed at this state can introduce hazards here, teaching penalty aversion.
- *Decelerate*: 0.5 to **BehindCar**, 0.5 to **SchoolZone**. This action is safer. Caution can balance delay and safety.
- *ChangeLane*: 0.8 to **Intersection**, 0.2 to **TrafficJam**. This action at this state is mostly smooth, minor jam risk but can introduce congestion chance.

2. From **BehindCar**, Decelerate: 0.6 to **SchoolZone**, 0.4 to **PedCrossing**. This is for cautious advance, where slowing can aid navigation.

- *ChangeLane*: 0.5 to **TrafficJam**, 0.5 to Intersection. The vehicle has equal risk/reward for overtaking at this state.
- *Stop*: 1.0 to **Intersection**. Safe pause would lead forward but delayed resolution.

3. From **SchoolZone**, *Maintain*: 0.6 to **PedCrossing**, 0.4 to **Obstacle**. There is risk if not slowing due to inertia.

- *Decelerate*: 0.7 to **Intersection**, 0.3 to **PedCrossing**. This is safer path. Caution can favor smooth flow.
- *Stop*: 1.0 to **Intersection**. This is for guaranteed safety.

4. From **PedCrossing**, Decelerate: 0.7 to **Destination**, 0.3 to **Obstacle**. High success with caution but partial risk for speed.

- *Stop*: 1.0 to **Destination**. Safest ends episode well with certain success.

5. From **TrafficJam**, *Maintain*: 0.4 to **Obstacle**, 0.6 to **Intersection**. Persistence/Waiting can mostly resolve here but risk escalation

- *ChangeLane*: 0.6 to **Destination**, 0.4 to **Obstacle**. Attempted escape with risk.
- *Stop*: 1.0 to **Destination**. Halt resolves. It's safe but slow.

6. From **Intersection**, *Maintain*: 0.6 to **Destination**, 0.4 to **TrafficJam**. Flow with back-up risk.

- *Decelerate*: 0.8 to **Destination**, 0.2 to **Obstacle**. Caution and safer have higher success.
- *ChangeLane*: 0.7 to **Destination**, 0.3 to **PedCrossing**. Maneuver with minor delay.

7. From **Obstacle**, *Decelerate*: 0.6 to **Intersection**, 0.4 to **TrafficJam**. Indicate slow recovery.

- *ChangeLane*: 0.5 to **Destination**, 0.5 to **ClearRoad**. Demonstrate evasive, potential reset (bounded by H).
- *Stop*: 1.0 to **Destination**. Safe end/halt.

8. From **Destination**: Self-loops with 1.0 for absorbing.

Rewards R(s,a,s') will value range -50 to +50, positive for safe/efficient transitions to low-risk states (+10 to +50), negative for risks/delays (-10 to -50), with terminals implicit (0 ongoing, but paths to **Destination** rewarded), guiding **DP** to high-value paths. This complements the

scenario by penalizing unsafe choices in "school zones" or "pedestrians", while encouraging for "clear road" efficiency, ensuring optimal policies prioritize safety without over-penalizing time (implicit -1/step via **Finite** H). For, example:

- In **ClearRoad**: *Maintain* to **SchoolZone** (+10, progress), to **PedCrossing** (-10, minor issue or delay); *Accelerate* to **SchoolZone** (+15), to Obstacle (-50, penalty); etc. I reward speed in safe contexts but punish risks.
- In **BehindCar**: *Decelerate* to **SchoolZone** (+10), to **PedCrossing** (+5); *ChangeLane* to **TrafficJam** (-30), to **Intersection** (+15). Here, I encourage overtaking only if beneficial.
- Similar for others: High +50 near terminals via safe actions (e.g., *Decelerate* in **PedCrossing**), -30 for **Obstacle** via risky (e.g., *Maintain* in **SchoolZone**).

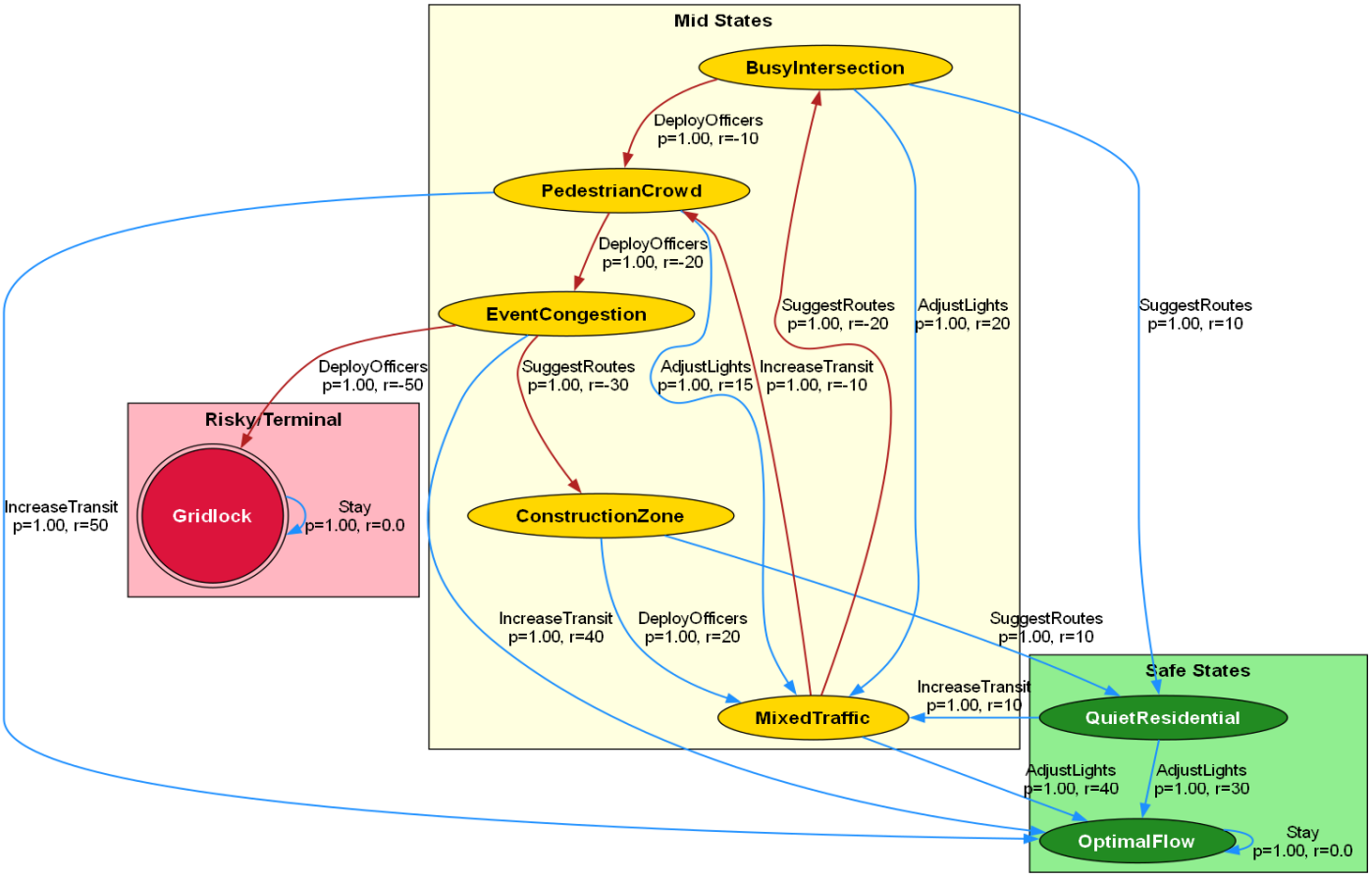
ClearRoad: *Maintain*: {SchoolZone: 10, PedCrossing: -10}, *Accelerate*: {SchoolZone: 15, Obstacle: -50}, *Decelerate*: {BehindCar: -20, SchoolZone: 5}, *ChangeLane*: {Intersection: 20, TrafficJam: -30}	**TrafficJam** *Maintain*: {Obstacle: -40, Intersection: 10}, *ChangeLane*: {Destination: 30, Obstacle: -50}, *Stop*: {Destination: 20}
BehindCar: *Decelerate*: {SchoolZone: 10, PedCrossing: 5}, *ChangeLane*: {TrafficJam: -30, Intersection: 15}, *Stop*: {Intersection: 10}	**Intersection**: *Maintain*: {Destination: 40, TrafficJam: -20}, *Decelerate*: {Destination: 50, Obstacle: -30}, *ChangeLane*: {Destination: 45, PedCrossing: -10}
SchoolZone: *Maintain*: {PedCrossing: -10, Obstacle: -30}, *Decelerate*: {Intersection: 20, PedCrossing: 10}, *Stop*: {Intersection: 15}	**Obstacle**: *Decelerate*: {Intersection: 10, TrafficJam: -20}, *ChangeLane*: {Destination: 30, ClearRoad: 20}, *Stop*: {Destination: 20}
PedCrossing: *Decelerate*: {Destination: 50, Obstacle: -20}, *Stop*: {Destination: 40}	**Destination**: *Stay*: {Destination: 0.0}

Scenario 2: Urban Traffic Management for a Smart City

This scenario models an AI-driven system managing city traffic across zones, optimizing signal timings, routes, and resources to minimize congestion and enhance safety during a **Finite** episode (e.g., a rush-hour cycle). The **MDP** is deterministic in transitions, meaning each action leads to a predictable next state, simplifying **DP** to exact path optimization rather than probabilistic averaging. The **Finite horizon** bounds the episode to prevent indefinite management, simulating a time-limited peak period. I used **8 states** to capture evolving traffic conditions across city zones, progressing from high congestion starts to resolve terminals. These states are acyclic in flow (busy to mixed to optimal), bounded by H to avoid loops.

1. **BusyIntersection**: High vehicle density at major crossroads, causing delays. This represents core urban bottlenecks or peak-hour jams; as a starting point with transitions to mid-states. High negative rewards here would motivate rapid shifts to better states.
2. **PedestrianCrowd**: Dense walker areas (e.g., near schools during peak times). This captures human-focused congestion and highlights human safety, rationalizing restricted actions to avoid accidents. Values here should penalize vehicle-favoring choices.

3. **ConstructionZone**: Roadwork-induced restrictions are narrowing lanes. This simulates planned/temporary disruptions. Deterministic transitions to quieter areas can emphasize recovery, with rewards for rerouting.
4. **EventCongestion**: Temporary surges from parades or emergencies. This captures unpredictable yet controllable events; as a branch state, it tests adaptive responses, rational for negative rewards leading to gridlock if mishandled, illustrating risk escalation and the need for greedy improvement to select escape actions.
5. **QuietResidential**: Low-traffic suburban zones with minimal issues. This acts as a neutral state for minimal intervention. Transitions from busy states will reward de-escalation, teaching value contrasts.
6. **MixedTraffic**: Blend of vehicles and pedestrians requiring balanced control. This is transitional hub state. Deterministic outcomes to optimal or backslides can model partial resolutions, with moderate rewards rationalizing trade-offs.
7. **OptimalFlow** (good terminal): Smooth, efficient traffic across the city. This is an ideal end, absorbing with zero ongoing rewards. It encourages termination via high entry rewards (+40 to +50), teaching goal-oriented optimization. **DP** can start from here (V=0), building values that favor paths maximizing the sum rewards without discounting dilution.
8. **Gridlock** (bad terminal): Total lockdown. This is a penalty end, designed as a sink for poor choices, with negative entry rewards (-50), rationalizing avoidance in policies.



Like in **Scenario 1, Actions are state-specific** with 2-3 per state (**5 unique**: *AdjustLights, SuggestRoutes, DeployOfficers, IncreaseTransit, Stay*) to reflect feasible interventions, also designed to reduce irrelevant choices and computation during policy improvement.

1. In **BusyIntersection**: *AdjustLights, SuggestRoutes, DeployOfficers*. These are vehicle-focused for heavy traffic. *AdjustLights* for immediate flow (+20), *SuggestRoutes* for diversion (+10), *DeployOfficers* for manual aid (-10 if leading to crowds).
2. In **PedestrianCrowd**: *AdjustLights, IncreaseTransit, DeployOfficers*. These are pedestrian-focused, excluding reroutes to avoid chaos. *IncreaseTransit* is best here for clearance (+50 to optimal), rationalizing safety rewards, and showing constraint-driven policies.
3. In **ConstructionZone**: *SuggestRoutes, DeployOfficers*. These are bypass options, limited to navigation aids. Lights/transit are ineffective here. *DeployOfficers* (+20 to mixed) vs. *SuggestRoutes* (+10 to quiet) can illustrate trade-offs in resource allocation.
4. In **EventCongestion**: *SuggestRoutes, IncreaseTransit, DeployOfficers*. These are for crowd management, event-specific, favoring mass transit. *IncreaseTransit* (+40 to optimal) but *DeployOfficers* risks gridlock (-50), rationalizing penalty to discourage overuse, teaching risk-aware improvement.
5. In **QuietResidential**: *AdjustLights, IncreaseTransit*. These are for minimal intervention. *AdjustLights* (+30 to optimal), *IncreaseTransit* (+10 to mixed), showing maintenance actions in low-stress states. There are no *DeployOfficers* in quiet areas to avoid overkill.
6. In **MixedTraffic**: *AdjustLights, SuggestRoutes, IncreaseTransit*. These are balanced options. AdjustLights (+40 to optimal), others risk backslides (-10/-20), rationalizing to favor stability.
7. In **OptimalFlow** and **Gridlock**: *Stay*. Absorption with zero reward, ensuring terminal focus.

Transition probabilities are deterministic (P=1 to one s per s,a), forward to mid/terminals. Actions have predictable effects leading to specific outcomes (e.g., *IncreaseTransit* resolves crowds to **OptimalFlow**). The forward design prevents cycles, bounded by H. This highlights determinism's role in exact optimality, contrasting Scenario 1's averaging.

1. From **BusyIntersection**, *AdjustLights* to **MixedTraffic**. Timings can reliably ease vehicles.
- *SuggestRoutes* to **QuietResidential**. Diversion clears mains predictably.
- *DeployOfficers* to **PedestrianCrowd**. Officers handle but shift to walkers.
2. From **PedestrianCrowd**, *AdjustLights* to **MixedTraffic**. Lights prioritize crossers safely.
- *IncreaseTransit* to **OptimalFlow**. Buses clear crowds efficiently.
- *DeployOfficers* to **EventCongestion**. Overuse escalates to events.
3. From **ConstructionZone**, SuggestRoutes to **QuietResidential**. Rerouting avoids work zone.
- *DeployOfficers* to **MixedTraffic**. Guidance mixes traffic smoothly.
4. In **EventCongestion**, *SuggestRoutes* to **ConstructionZone**. Diversion but to related issues.
- *IncreaseTransit* to **OptimalFlow**. Transit resolves temporarily.
- *DeployOfficers* to **Gridlock**. Misapplication worsens.
5. From **QuietResidential**, *AdjustLights* to **OptimalFlow**. Fine-tuning optimizes low traffic.
- *IncreaseTransit* to **MixedTraffic**. Added service mixes flows.
6. From **MixedTraffic**, *AdjustLights* to **OptimalFlow**. Balances to ideal.
- *SuggestRoutes* to **BusyIntersection**. Potential backslide if misused.
- *IncreaseTransit* to **PedestrianCrowd**. Increase walkers.

7. **Terminals**: Stay to self.

Rewards R(s,a,s) are deterministic, scaled to drive efficiency (+10 to +50 for progress, -10 to -50 for setbacks). They are designed to prioritize flow and safety, with positives for resolutions reducing congestion and negatives penalizing escalation, encouraging policies that maximize cumulative sum and toward efficient actions like *IncreaseTransit* for crowds.

1. In **BusyIntersection**, *AdjustLights* to **MixedTraffic**: +20. Moderate gain for partial relief.
- *SuggestRoutes* to **QuietResidential**: +10. Lesser for diversion delay.
- *DeployOfficers* to **PedestrianCrowd**: -10. Costly shift to walkers.
2. In **PedestrianCrowd**, *AdjustLights* to MixedTraffic: +15. Balanced safety/efficiency.
- *IncreaseTransit* to **OptimalFlow**: +50. High for resolution.
- *DeployOfficers* to **EventCongestion**: -20. Worsens to events.
3. In **ConstructionZone**, *SuggestRoutes* to **QuietResidential**: +10. Basic reroute gain.
- *DeployOfficers* to **MixedTraffic**: +20. Effective guidance.
4. In **EventCongestion**, *SuggestRoutes* to **ConstructionZone**: -30. Cycles to issues.
- *IncreaseTransit* to **OptimalFlow**: +40. Strong resolution.
- *DeployOfficers* to **Gridlock**: -50. Severe penalty.
5. In **QuietResidential**, *AdjustLights* to **OptimalFlow**: +30. Easy optimization.
- *IncreaseTransit* to **MixedTraffic**: +10. Minor improvement.
6. In **MixedTraffic**, *AdjustLights* to **OptimalFlow**: +40. Key balance.
- *SuggestRoutes* to **BusyIntersection**: -20. Backslide cost.
- *IncreaseTransit* to **PedestrianCrowd**: -10. Increases density.
7. In **Terminals**: 0.0 self.

Experimental results and discussion:

a. Experimental settings:

Value Iteration (VI) and **Policy Iteration (PI)** were applied to solve the formulated **MDPs** for both scenarios, with adaptations for both **Finite horizon** (undiscounted *γ*=1.0) and **Infinite-horizon** (discounted *γ*=0.9) variants to compare behaviours under bounded episodes versus long-term optimization. The **Finite**-**horizon** setup used a **horizon** of *H=100*, chosen to exceed the observed convergence steps while preventing indefinite computation. This *H* value was selected after my initial runs showed that smaller **horizons** (e.g., 10) truncated before full value propagation, leading to suboptimal policies, whereas 100 allowed sufficient depth for the state chains without excessive runtime.

For **VI**, the **Finite** version computed time-dependent values $V_h(s)$ backward from *h=0* (terminal values 0) to *h=H*, updating via $V_h(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_{h-1}(s')]$. In other words, for **Finite-horizon**, the objective is to maximize the expected *undiscounted* sum of rewards over *H* steps: $E[\sum_{t=0}^{H-1} R(s_t, a_t, s_{t+1})]$, computed via backward DP. In contrast, **Infinite-horizon MDPs** use *discount* to ensure convergence: $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$. For deeper insights, I also tracked delta changes here ($\Delta = \max_s |V_h(s) - V_{h-1}(s)|$) without

breaking the loop to observe natural stabilization in **Finite** settings, revealing how many steps "would converge" if treated infinitely (e.g., first *h* where $\Delta < \theta$). The convergence threshold was θ=0.001, a standard small value ensuring near-optimality without over-iteration. Policies were extracted greedily at each *h*, resulting in deterministic mappings (probability 1.0 for best action), as optimal policies suffice in these **MDPs** per theory.

For **PI,** I initialized it with uniform stochastic policies ($\pi(a|s) = 1/|A(s)|$) to promote initial broad evaluation, then alternated evaluation and improvement until policy stability (all probability differences <1e-6), counting iterations without a max limit. The evaluation will solve Bellman expectation $V_\pi[s] = \sum_a \pi(a|s) \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma V_\pi[s']\right]$ iteratively until $\Delta < \theta$ in **Infinite** cases, or backward over *h* in **Finite**.

Simulations computed average rewards over 1000 episodes from starting states (**ClearRoad** for **Scenario 1**, **BusyIntersection** for **Scenario 2**), using undiscounted sums for **Finite** and discounted for **Infinite**, to quantify solution quality.

b. Experimental results:

i. **Performance:**

For **Scenario 1** (stochastic transitions), overall, both algorithms yielded identical optimal policies and value functions in **Finite** settings (e.g., favouring *Decelerate* for safety), with 0 differences in value (0.0) and policy probabilities (0.0), implying equivalence at optimality as per policy improvement theorem (once optimal, further improvements cease). However, **PI** was slower (0.167s vs 0.022s) but needed only 3 iterations compared to effective 20 for **VI**'s delta stabilization, highlighting **PI**'s efficiency in iteration count for stochastic cases (fewer policy changes suffice as evaluations capture uncertainties holistically). Average rewards were nearly identical (~50.3-50.4), close to max possible path sums, but **Infinite** variants showed ~2% lower rewards (~49.3-49.5) due to *γ*=0.9 discounting longer risky paths, with minor time savings (0.004-0.010s) but similar iterations compared to **Finite** versions, implying discounting accelerates convergence by bounding **Infinite** sums.

To be more specific, **Finite VI** with *H=100* took 0.022s, stabilizing deltas at *h=20* (would converge early if threshold-based, illustrating how stochasticity requires more propagation steps for averaging uncertainties via $\sum_{s'} P(s'|s,a)$). The optimal value function at *H* showed high returns for safe states (e.g., $V(\text{BehindCar}) = 63.66$, reflecting recovery potential) and lower for risky ones (e.g., $V(\text{TrafficJam}) = 39.34$), with policy favoring conservative actions like *Decelerate* in **SchoolZone** and **PedCrossing** to maximize expected safety rewards. Average reward was 50.32, close to theoretical max paths (e.g., sequences summing +50 progress rewards minus minor penalties). **Finite PI** converged in 3 iterations (0.167s), yielding identical values/policies/rewards (50.425 with minor variance), highlighting fewer but costlier steps as evaluation solves linear systems per policy, which is expensive in stochastic settings but fast overall due to quick stability. On the other hand, **Infinite** variants (*γ*=0.9) converged faster with **VI** in 17 iterations (0.004s) and **PI** in 3 (0.010s), with slightly lower values (e.g., $V(\text{SchoolZone}) \approx 55.1$) and rewards (~49.3-49.5) due to discounting penalizing long paths, but same policies. Discounting contracts the effective horizon, reducing iterations as updates dampen far-future effects per contraction mapping theorem ($\left||V - TV|\right| \leq \gamma \left||V - V'|\right|$). This explains quicker times.

For **Scenario 2** (deterministic transitions), overall, **Finite** algorithms again matched exactly (value diff=policy diff=0), but **VI** was faster (0.016s vs. 0.191s) despite full *H=100* steps, as

deterministic updates are lightweight (no prob sums), while **PI**'s 6 iterations reflect deeper refinements in chain-like structures. Rewards were exact (40.0), matching deterministic path maxima, with **Infinite** versions converging in just 3 iterations each (ultra-fast times <0.003s), demonstrating determinism's role in rapid stability, where discounting further trims long chains, yielding scaled values (e.g., 56.0 vs. 305.0 **Finite**) but consistent rewards.

To be more specific, **Finite VI** (0.016s) required full *H=100* for delta stabilization (long deterministic chains propagate exactly, no early averaging halt), with high values for resolvable states (e.g., $V(\text{BusyIntersection}) = 305.0$, summing multi-step +20/+40 rewards) and policy selecting efficiency actions like *AdjustLights*. Average reward 40.0 matches path sums. **Finite PI** took 6 iterations (0.191s) with identical outcomes and more iterations than **Scenario 1** due to deeper policy refinements. On the other hand, **Infinite** versions were rapid with **VI** in 3 iterations (0.001 seconds), **PI** also in 3 (0.003 seconds), scaled-down values (e.g., $V(\text{ConstructionZone}) = 56.0$), and rewards (40.0). Compared to **Finite horizon**, these **Infinite** approaches achieve same core policies but exhibit minor differences (e.g., *IncreaseTransit* vs *SuggestRoutes* in **EventCongestion**) due to discounting, as γ<1 prefers shorter paths to terminals. Determinism explains low iterations with no probability branching and updates are direct max over successors. This also illustrates how discounting aids stability in non-episodic views, as indicated by the fact **Finite VI** with *γ*=1.0 could go beyond its **horizon**, while **Infinite** one with *γ*=0.9 only needs 17 steps to converge.

ii. **Comparative study:**

Case Study	Algorithm	Optimal Policy	Optimal Value Function	Convergence Time	Number of Iterations	Average Reward
1	**Policy Iteration**	**Finite (at H)**: {'ClearRoad': {'Maintain': 1.0}, 'BehindCar': {'Decelerate': 1.0}, 'SchoolZone': {'Decelerate': 1.0}, 'PedCrossing': {'Decelerate': 1.0}, 'TrafficJam': {'Maintain': 1.0}, 'Intersection': {'Decelerate': 1.0}, 'Obstacle': {'ChangeLane': 1.0}} **Infinite**: same as **Finite**	**Finite (at H)**: {'ClearRoad': 61.32075471698112, 'BehindCar': 63.66037735849056, 'SchoolZone': 62.30188679245283, 'PedCrossing': 45.698113207547166, 'TrafficJam': 39.343396226415095, 'Intersection': 45.132075471698116, 'Obstacle': 55.660377358490564, 'Destination': 0.0} **Infnite**: {'ClearRoad': 50.00531689696372, 'BehindCar': 52.810674721234925, 'SchoolZone': 55.09949661126098, 'PedCrossing': 41.825478684551314, 'TrafficJam': 30.07776691281328, 'Intersection': 42.55031912303421, 'Obstacle': 47.5021230575247, 'Destination': 0.0}	**Finite**: 0.167s **Infinite**: 0.010s	**Finite**: 3 (stable) **Infinite**: 3 (stable)	**Finite**: 50.425 **Infinite**: 49.54
1	**Value Iteration**	**Finite (at H)**: same as **PI** **Infinite**: same as **PI**	**Finite (at H)**: same as **PI** **Infnite** (almost similar to **PI**): {'ClearRoad': 50.005212355610865, 'BehindCar': 52.8105772031458, 'SchoolZone': 55.09945415440576, 'PedCrossing': 41.82551181465404, 'TrafficJam': 30.077788934880857, 'Intersection': 42.55034120976936, 'Obstacle': 47.5021493538891, 'Destination': 0.0}	**Finite**: 0.022s **Infinite**: 0.004s	**Finite**: 100 (full H, conv at 20) **Infinite**: 17 (*Δ*<θ)	**Finite**: 50.32 **Infinite**: 49.34

2	**Policy Iteration**	**Finite (at H)**: {'BusyIntersection': {'AdjustLights': 1.0}, 'PedestrianCrowd': {'AdjustLights': 1.0}, 'ConstructionZone': {'DeployOfficers': 1.0}, 'EventCongestion': {*'SuggestRoutes'*: 1.0}, 'QuietResidential': {'IncreaseTransit': 1.0}, 'MixedTraffic': {*'IncreaseTransit'*: 1.0}} **Infinite**: {'BusyIntersection': {'AdjustLights': 1.0}, 'PedestrianCrowd': {'AdjustLights': 1.0}, 'ConstructionZone': {'DeployOfficers': 1.0}, 'EventCongestion': {*'IncreaseTransit'*: 1.0}, 'QuietResidential': {'IncreaseTransit': 1.0}, 'MixedTraffic': {*'AdjustLights'*: 1.0}}	**Finite (at H)**: {'BusyIntersection': 305.0, 'PedestrianCrowd': 300.0, 'ConstructionZone': 305.0, 'EventCongestion': 270.0, 'QuietResidential': 295.0, 'MixedTraffic': 285.0, 'OptimalFlow': 0.0, 'Gridlock': 0.0} **Infinite**: {'BusyIntersection': 56.0, 'PedestrianCrowd': 51.0, 'ConstructionZone': 56.0, 'EventCongestion': 40.0, 'QuietResidential': 46.0, 'MixedTraffic': 40.0, 'OptimalFlow': 0.0, 'Gridlock': 0.0}	**Finite**: 0.191s **Infinite**: 0.003s	**Finite**: 6 (stable) **Infinite**: 3 (stable)	**Finite**: 40.0 **Infinite**: 40.0
2	**Value Iteration**	**Finite (at H)**: same as **PI** **Infinite**: same as **PI**	**Finite (at H)**: same as **PI** **Infinite**: same as **PI**	**Finite**: 0.016s **Infinite**: 0.001s	**Finite**: 100 (full H, conv at 100) **Infinite**: 3 (Δ<θ)	**Finite**: 40.0 **Infinite**: 40.0

iii. Discussion:

VI, being a model-based method that iteratively refines the value function across all states until convergence, excels in scenarios with moderate state spaces by providing granular updates but can require more iterations in stochastic environments like **Scenario 1**, where probabilistic transitions necessitate repeated averaging to stabilize expectations. In contrast, **PI**, which decouples evaluation (solving for V_π under a fixed policy) from improvement (greedy update to π), often converges in fewer outer iterations but incurs higher per-iteration costs due to full policy evaluations, making it advantageous in deterministic settings like **Scenario 2** where exact computations are straightforward.

Specifically, from my experiments**, VI** showed its primary strength in simplicity and fine-grained updates, making it computationally lightweight per iteration, as evidenced by shorter times in both scenarios (e.g., 0.022s **Finite** in **Scenario 1**, 0.016s in **Scenario 2**). This stems from sweeping all states/actions in each pass, allowing parallelizable implementations and adaptability to large state spaces without solving full linear systems. In **Scenario 1**'s stochastic, this proved advantageous for handling probabilistic branches, where **VI** required an effective 20 steps for delta stabilization in **Finite** (tracking Δ revealed exponential decay due to averaging over probabilities) and 17 iterations in **Infinite**, yielding high-quality solutions (avg. reward 50.32 **Finite**, 49.34 **Infinite**) by gradually refining expectations over uncertain outcomes like obstacle probabilities. This demonstrates **VI**'s resilience to noise, as successive max operations propagate conservative estimates

backward, aligning with real-vehicle needs for risk-averse planning. In **Scenario 2**'s deterministic chains, its strength amplified by only 3 **Infinite** iterations (0.001s), with exact path maxima (rewards 40.0), illustrating how absence of probabilities reduces to efficient tree search, ideal for time-critical traffic adjustments where predictable transitions allow quick value flooding from terminals.

However, **VI**'s limitations emerge in iteration counts and potential slow convergence. In **Finite Scenario 1**, full *H=100* was run despite early stabilization at h=20, wasting cycles. This is an issue in undiscounted settings where long-**horizon** effects persist without contraction. In **Scenario 2**, **Finite** needed all 100 steps for delta drop (long deterministic paths require full backward passes), highlighting inefficiency in chain-like structures, while **Infinite** mitigated this via γ=0.9, contracting influences and converging in 3 steps, but at the cost of scaled-down values (56.0 vs. 305.0 **Finite** for **BusyIntersection**), underestimating long-term rewards in undiscounted contexts like sustained traffic flow. This limitation implies **VI** suits exploratory phases or when partial solutions suffice, but may falter in precise, long-**horizon** planning without tuning θ or early stopping.

In contrast, **PI** exhibits strength in fewer outer iterations and guaranteed monotonic improvement, converging rapidly to the global optimum, as evidenced by only 3 **Finite** iterations in **Scenario 1** (0.167s) and 6 in **Scenario 2** (0.191s), with identical quality to **VI** (zero diffs), as each policy upgrade strictly betters or equals the prior via the improvement theorem $V_{\pi'} \geq V_{\pi}$. This efficiency in steps arises from exact evaluations capturing full policy implications. In stochastic **Scenario 1**, holistic summing over π(a|s) and **PI** stabilized quickly, yielding robust policies (e.g., *Decelerate* dominance for safety) and rewards (50.425 **Finite**), ideal for uncertain environments where intermediate policies guide exploration. In deterministic **Scenario 2**, the strength held with 3 **Infinite** iterations (0.003s), but more **Finite** (6) due to refining deeper chains, still faster overall in steps than **VI**'s 100 & 3.

Based on my experiments, I found that the limitations of **PI** can include higher per-iteration costs. This is because evaluation solves systems iteratively until Δ<θ. This is expensive in large/stochastic spaces, which explains longer times like 0.167s in **Scenario 1** vs **VI**'s 0.022s, potentially scaling poorly without approximations. Additionally, in **Scenario 2**, **Infinite** variants showed minor policy divergences (e.g., *IncreaseTransit* over *SuggestRoutes*), as discounting altered greedy choices toward shorter resolutions. This is a subtlety absent in **Finite**'s full summation, implying sensitivity to γ in undiscounted settings.

Conclusion

In conclusion, my experiments have demonstrated a robust application of **MDPs** and **dynamic programming** to real-world decision-making challenges, yielding optimal policies and value functions that balance safety, efficiency, and uncertainty in autonomous vehicle navigation and urban traffic management. In **Scenario 1**, the stochastic transitions and risk-oriented rewards (e.g., -50 for obstacles) led to conservative policies like *Decelerate* dominance, achieving average rewards around 50 in finite settings through expected value maximization, with convergence in 20 effective steps for **Value Iteration** and 3 iterations for **Policy Iteration**, which is justified by the need to average probabilistic paths, as per the Bellman equation's summation over P(s'|s,a). **Scenario 2**'s deterministic dynamics and progress-focused rewards (e.g., +50 for resolution) produced efficiency-driven policies like *AdjustLights*, with exact rewards of 40 and rapid **infinite** convergence (3 iterations), attributable to single-outcome

transitions simplifying updates to direct max operations, though **finite** required full *H=100* for long-chain propagation. I also noticed that **infinite-horizon** variants can converge quickly under discounting as it does not depend on a fixed *H* to stop. However, this setting (using θ to stop) can be a disadvantage if the problems can't be converged, making the use of **finite** setting important to break the loop. Moreover, I think that **VI**'s scalability is an advantage for large stochastic spaces (finer-grained progress) while **PI** will excel in scenarios needing few refinements (e.g., deterministic ones), as although it typically requires fewer iterations, it needs more time per iteration compared to **VI**. However, I think this can also be interpreted in another way: Stochasticity would favor **PI**'s stability for holistic evaluations, while determinism leans towards **VI**'s speed. Overall, these outcomes have validated my **MDPs**' formulations.