Decision-Making Applications

Retail

- Pricing
- Managing fast inventory
- Making real-time recommendations
- Fulfilling orders

Finance

- Selecting financial advisors
- Automated trading

Vehicle routing

- Deciding route to work
- Deciding route in a new city
- Deciding route in a city under construction

Video game

- Self play on AlphaGo Zero
- Supervised learning on AlphaGo Zero

Model Predictive Control for Retail Pricing: Prediction

Neural networks are best suited for predicting retail pricing because:

- Retail has high volumes of data
- Retail data is unstructured

Model Predictive Control for Retail Pricing: Optimization

Define the objective

Examples:

- Maximize average revenue
- Increase profit
- Balance risk

Describe the operational constraints

Examples:

- Sell #units
- Manufacture not more than #units
- Manufacture at least #units

Solve the optimization problem

Model:

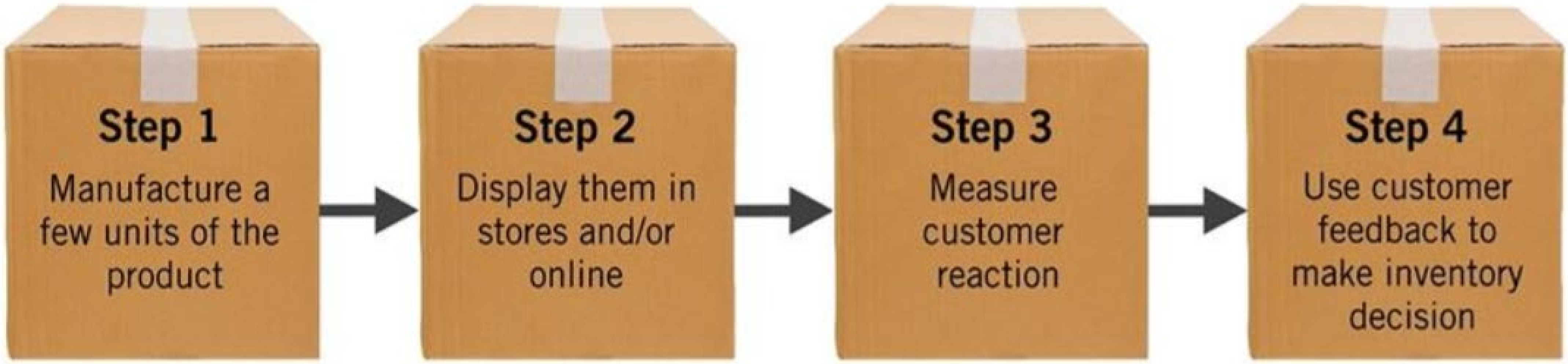
- Use the incremental likelihood of purchase

Calculate output

- If product is sold → Probability of revenue
- If product is not sold → Probability of loss

Fast Inventory

Fast inventory is a creative and pragmatic solution for the challenges in inventory decision-making.



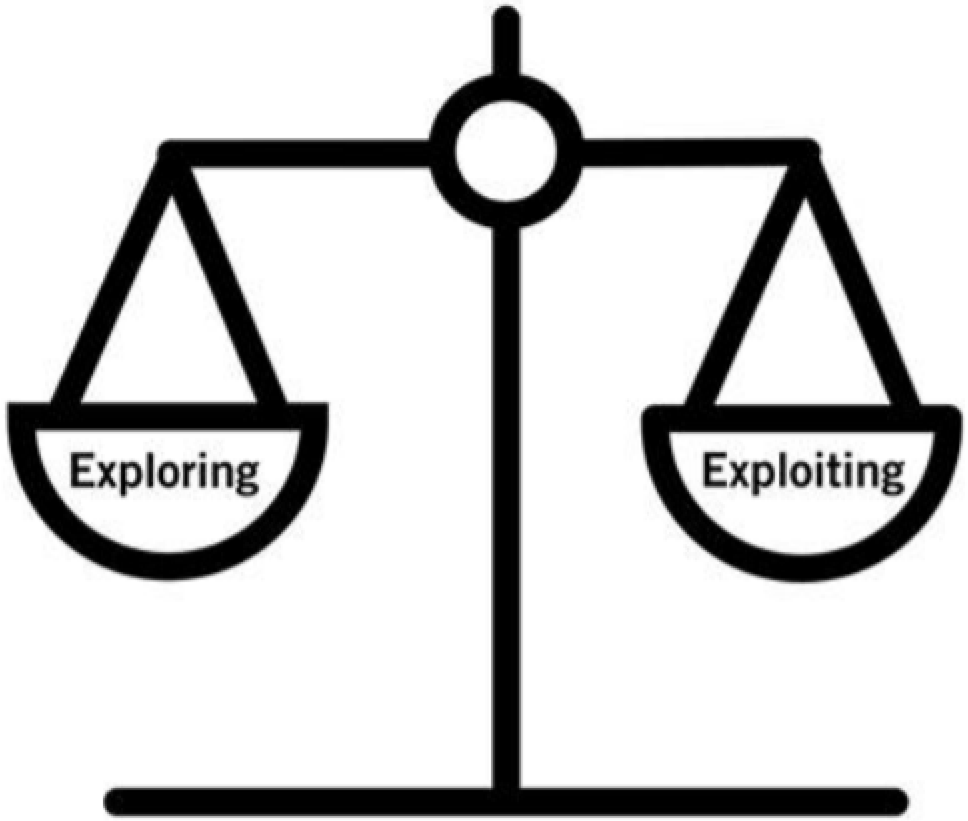
Fast Inventory Is Decision-Making Case 2

State dynamics	Fixed	
Information rate	High	Low

Solution: Multi-armed bandit

- State/environment does not change
  - Since products are introduced in a short period, customers do not change
- Low rate of data collection
  - When products are introduced in a short period, data collection is relatively low

Multi-armed Bandit for Inventory Decision-Making



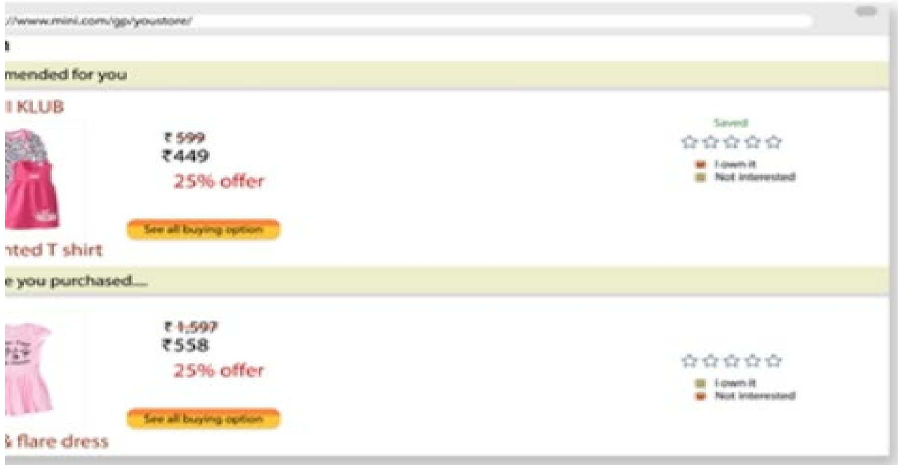
- Explore with a small number of units across products and exploit the products that sell better
- Balance exploration and exploitation
- Use machine learning and domain knowledge to determine how to explore and exploit

Machine Learning and Multi-armed Bandit

Contextual multi-armed bandit

- Choose products for testing so that all features and attributes are reasonably presented
- Extrapolate the results across untested products

Designing Real-Time Recommendation Systems



Challenge: Website has many too products to display when a customer visits a website

Objective: To display only **a few relevant products** of the many options available

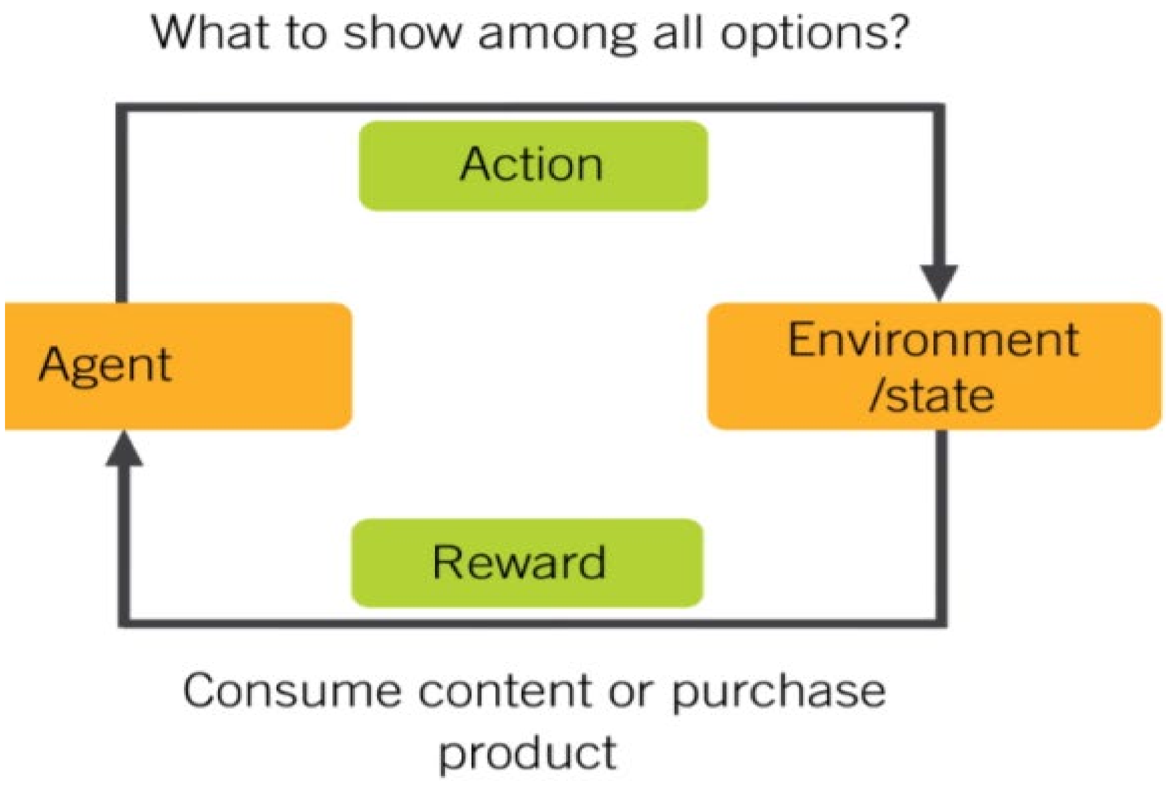
End goal: To maximize engagement, profit, revenue, and/or information access

Decision-making approach

Build the recommendation system using data such as:

- Customer history
- Brand attributes
- Product choices
- Real-time information

Designing Real-Time Recommendation Systems



- State is individual's history of interaction and attributes
- It changes with every interaction

Designing Recommendation Systems Is Case 3 of Decision-Making

Markov decision process (MDP) can be used to design real-time recommendation systems.

State dynamics	High	Low
Information rate	High	Low

Designing recommendation systems is similar to playing a game of chess with a known opponent because:

- Rich historical collective data is available
- State is constantly changing

MDP for Real-Time Recommendation System Design

Offline decision

- Use policy iteration method to determine the optimal policy

→

Real-time decision (when a customer interacts with the website)

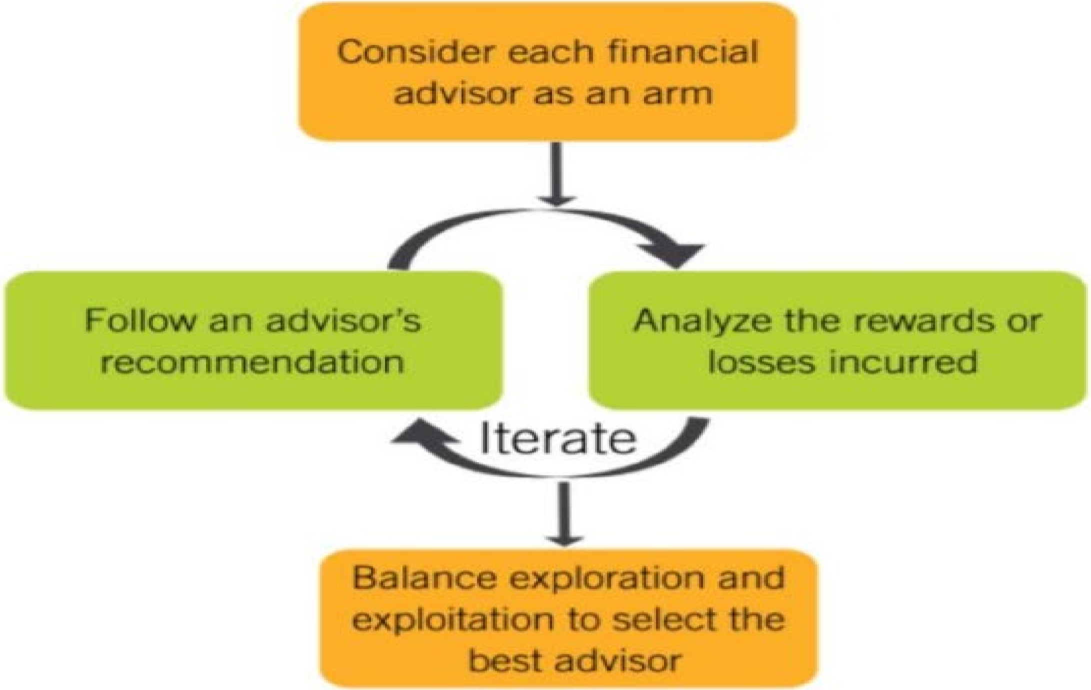
- Use the best decision from policy iteration method
- React to the existing state and provide the best recommendations

Decision-Making Framework for Order Fulfillment

Order fulfillment is similar to playing a game of chess with an unknown opponent.

Multi-armed Bandit for Choosing the Best Financial Advisors

Explore so that every financial advisor is well-tested and exploit the good advisors.



The right exploration and exploitation trade-off using multi-armed bandit can help determine the optimal solution, equivalent to having prior knowledge of the best financial advisor

Portfolio Decision-Making

Assumption 1

- You have a predictive model that predicts the performance of your investment portfolio over a period with standard deviation and distribution
- The environment does not change during the decision period

Human-Driven Portfolio Management

Assume that:

- You have a predictive model
- Using the predictive model, you want to **invest once** for your decision period
- During the decision period, there is no active management
- You revisit the decision for the subsequent decision period

Model Predictive Control for Portfolio Management

State dynamic	Fixed	
Information rate	High	Low

- Robust predictive model is available
- Environment remains fixed during the decision period
- Decisions are made just once

Model Predictive Control for Portfolio Management

Use the objective with all the constraints and the predictive model to solve the optimization problem.

- Predictive perfomance of portfolio instruments over a period

Step 1: Prediction

- Determine the constraints
- Determine the objective

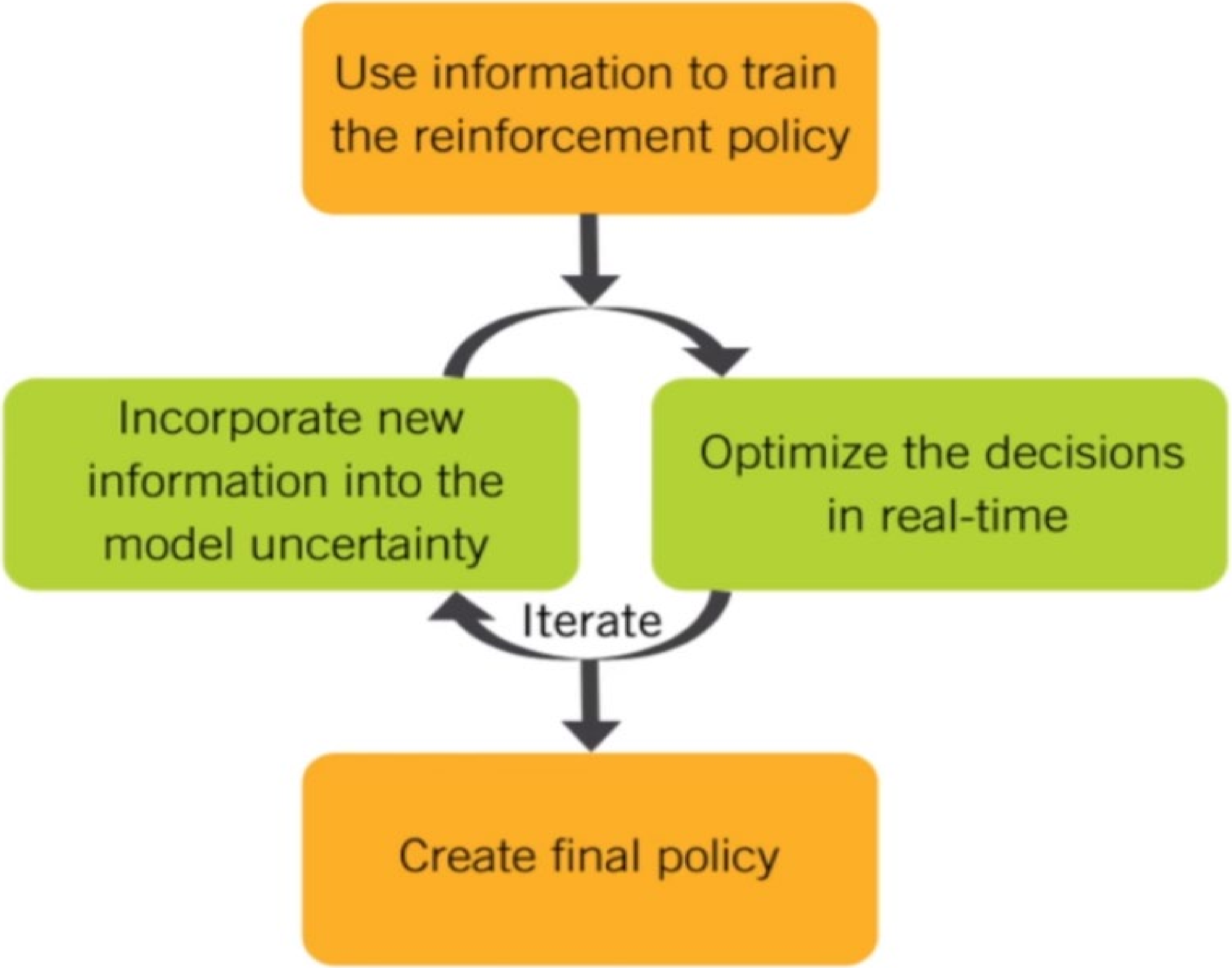
Step 2: Optimize

Automated Trading Is Case 3 of Decision-Making

State dynamics	High	Low
Information rate	High	Low

- Environment is constantly changing
- Information level is high

Reinforcement Learning for Automated Trading



Selecting the Best Route: Gathering Data

Inputs

- Start and end points
- Historical and current traffic information

Goal

- To minimize the average travel time and variation in it

State

- Current state of roads

Selecting the Best Route: Gathering Data

Inputs

- Start and end points
- Historical and current traffic information

Goal

- To minimize the average travel time and variation in it

State

- Current state of roads

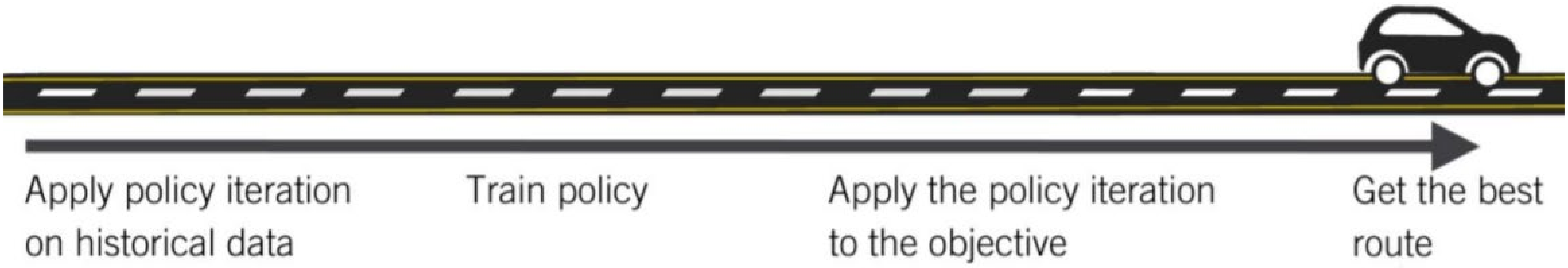
Markov Decision Process (MDP) for Selecting the Best Route

Markov decision process (MDP) should be used to determine the best route to work.

State dynamics	High	Low
Information rate	High	Low

- Information level is high
- Environment varies
  - With every routing decision, the state changes

Selecting the Best Route: Policy Iteration



Multi-armed Bandit for Deciding the Route in a New City

Inputs

- Start and end points
- Possible routes to the end point

Exploration

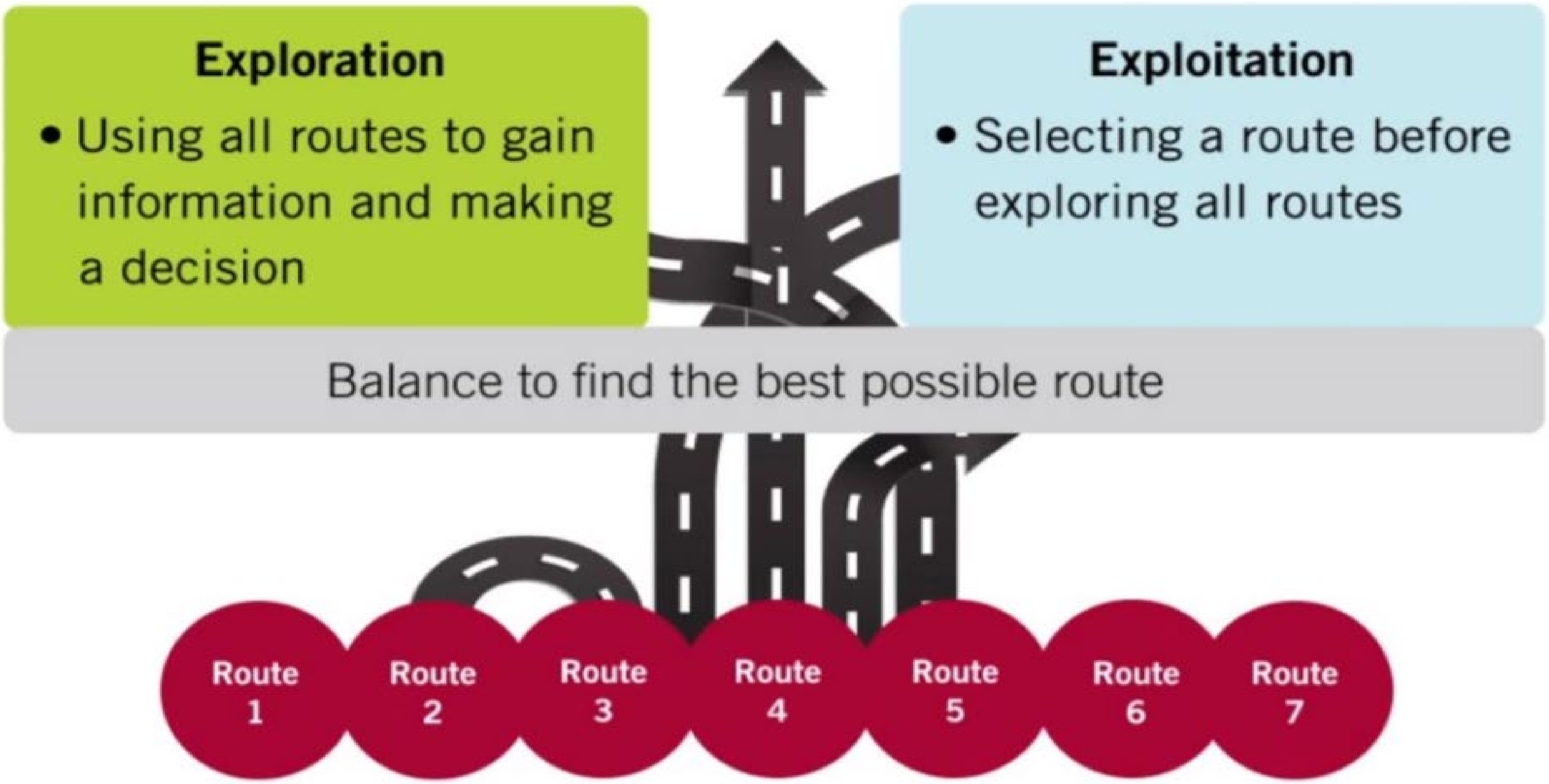
- Using all routes to gain information and making a decision

Exploitation

- Selecting a route before exploring all routes

Collect information about the routes → Select the best route for the objective

Multi-armed Bandit for Deciding the Route in a New City



Multiple Objectives While Gathering Data

- During exploration for an objective, information not related to the objective is also collected
- This information can be used if the objective changes

Transferred learning

Collecting information for one objective and transferring it to make better decisions for other objectives

Selecting the Best Route in a City Under Construction: Gathering Data

Input

Current state of roads

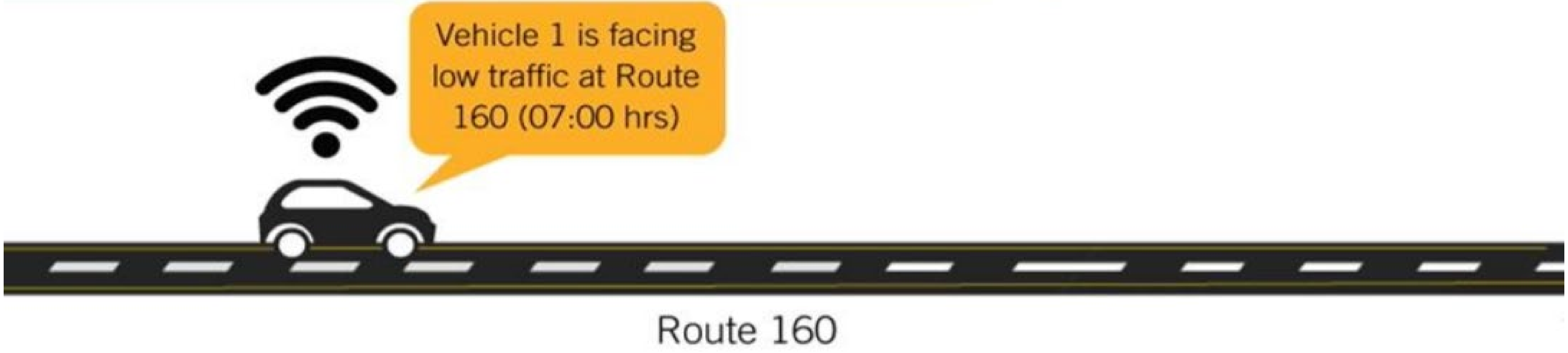
Data (prior and current) for future prediction

Goal

- To minimize travel time

Challenge

- Changing state



Reinforcement Learning for Deciding the Best Route

Reinforcement learning should be used to determine the best route in a city under construction.

State dynamics	High	Low
Information rate	High	Low

- Information rate is extremely low
- Environment constantly changes