



量产自动驾驶中的决策规划

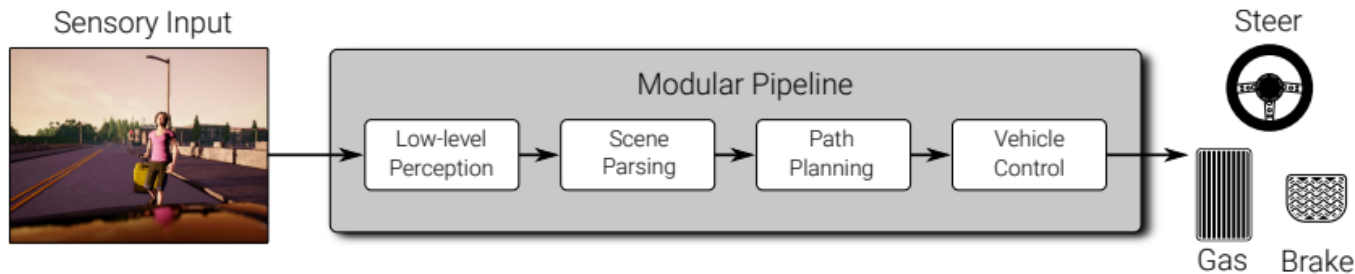
从领航高速(NOP)到领航城区(City-NOP)

肖飞宇, 2/17/2023

引言

自动驾驶中的决策规划

Planning and Decision Making



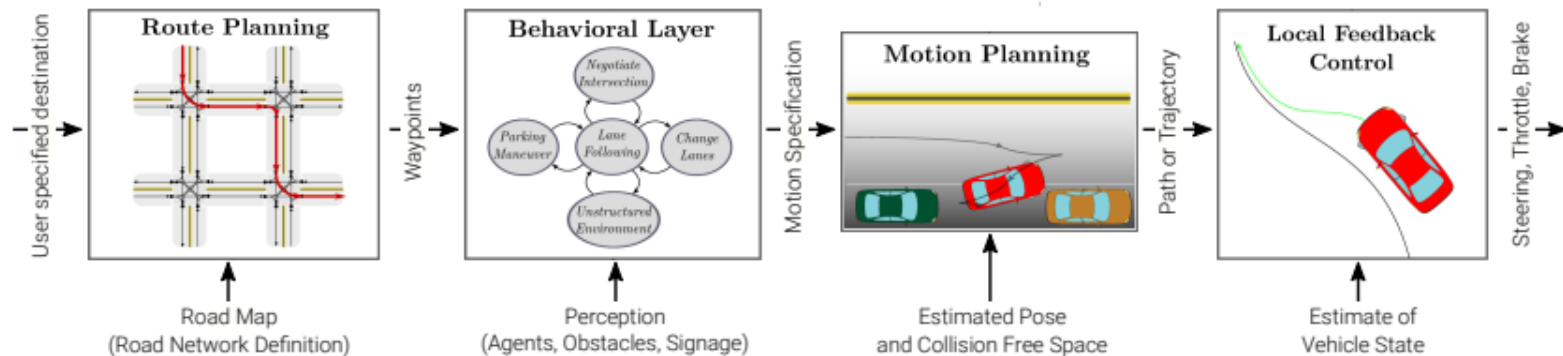
Problem Definition

- Goal: find and follow path from current location to destination
- Take static infrastructure and dynamic objects into account
- Input: vehicle and environment state (via perception stack)
- Output: Trajectory as input to vehicle controller

Challenges:

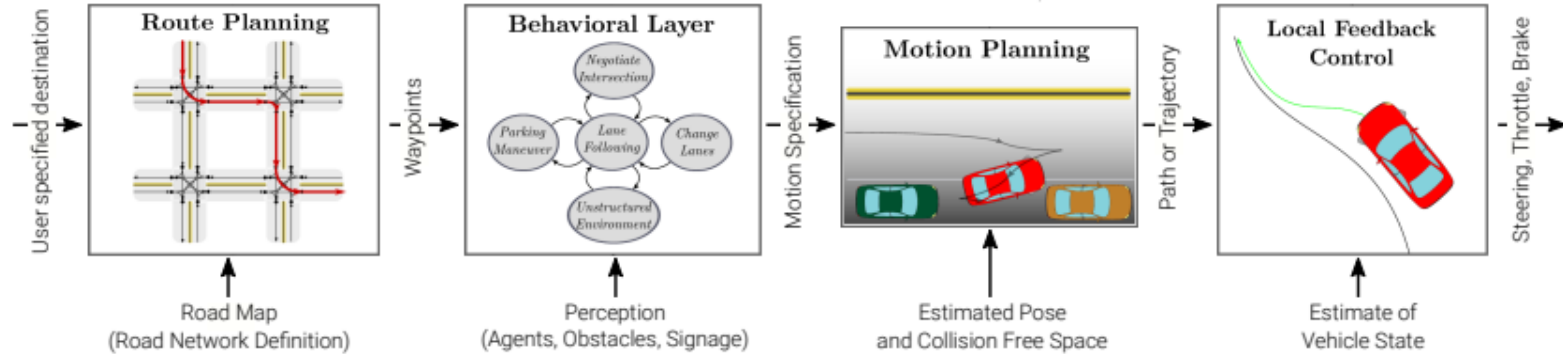
- Driving situations and behaviors are very complex
- Difficult to model as a single optimization problem

Planning and Decision Making



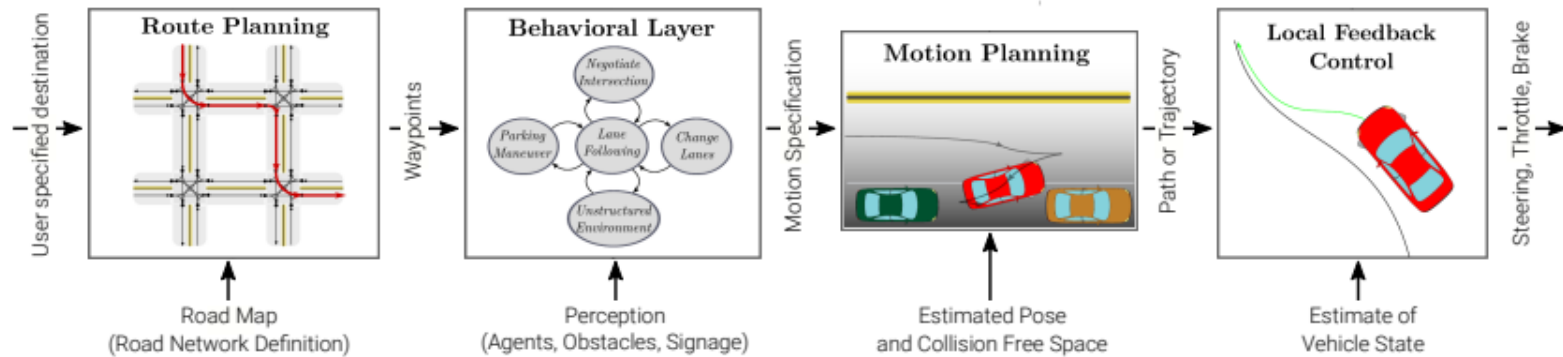
- Idea: Break planning problem into a **hierarchy of simpler problems**
- Each problem tailored to its scope and level of abstraction
- Earlier in this hierarchy means higher level of abstraction
- Each optimization problem will have constraints and objective functions

Planning and Decision Making



Hierarchy: A destination is passed to a route planner that generates a route through the road network. A behavioral layer reasons about the environment and generates a motion specification to progress along the selected route. A motion planner then solves for a feasible motion accomplishing the specification. A feedback control adjusts actuation variables to correct errors in executing the reference path.

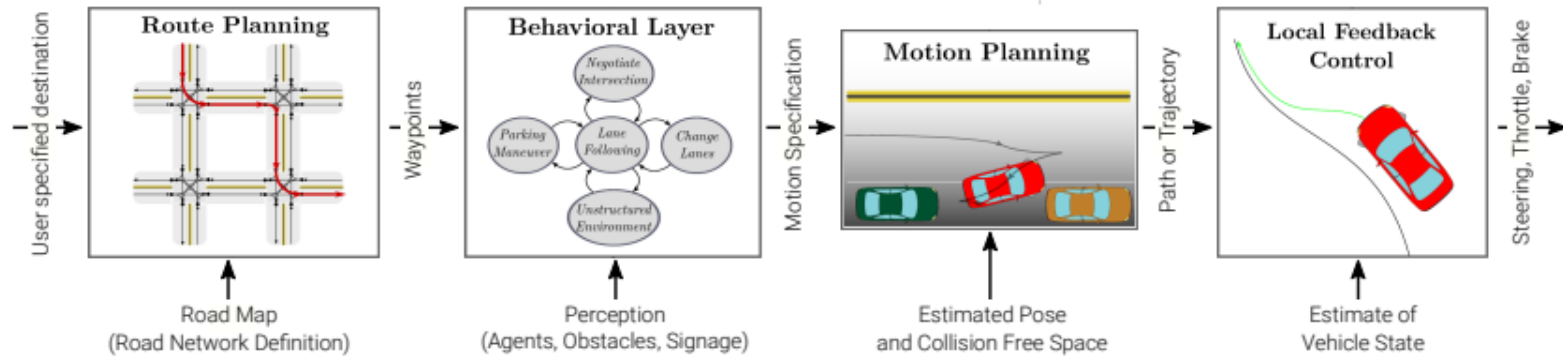
Step 1: Route Planning



- Represent **road network** as **directed graph**.
- Edge weights correspond to road segment length or travel time
- Problem translates into a minimum-cost graph network problem
- Inference algorithms: Dijkstra, A^*

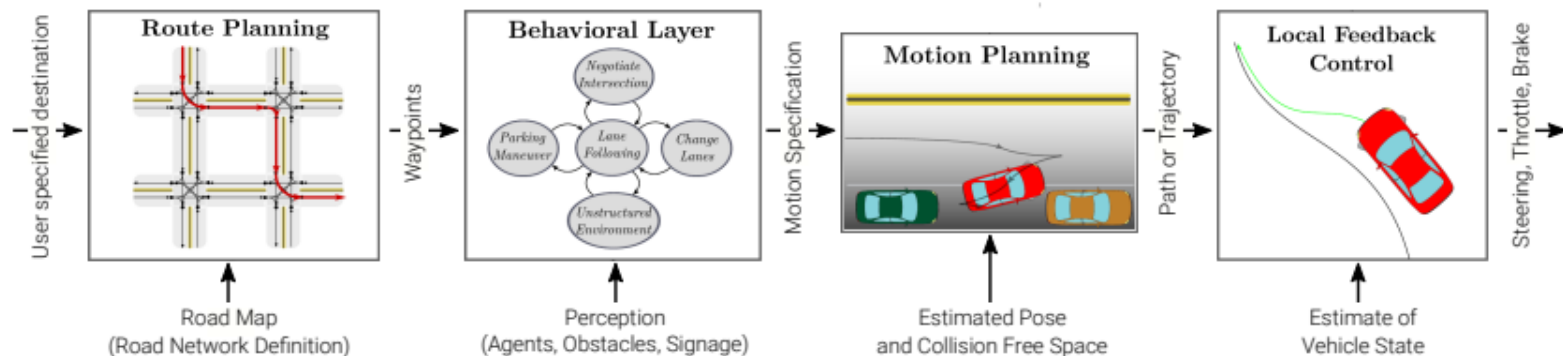
Solved for industrial applications like Map Apps.

Step 2: Behavior Planning



- Select **driving behavior** based on current vehicle/environment state
- E.g. at stop line: stop, observe other traffic participants, traverse
- Often modeled via **finite state machines** (transitions governed by perception)
- Can be modeled probabilistically, e.g., using Markov Decision Processes (MDPs)

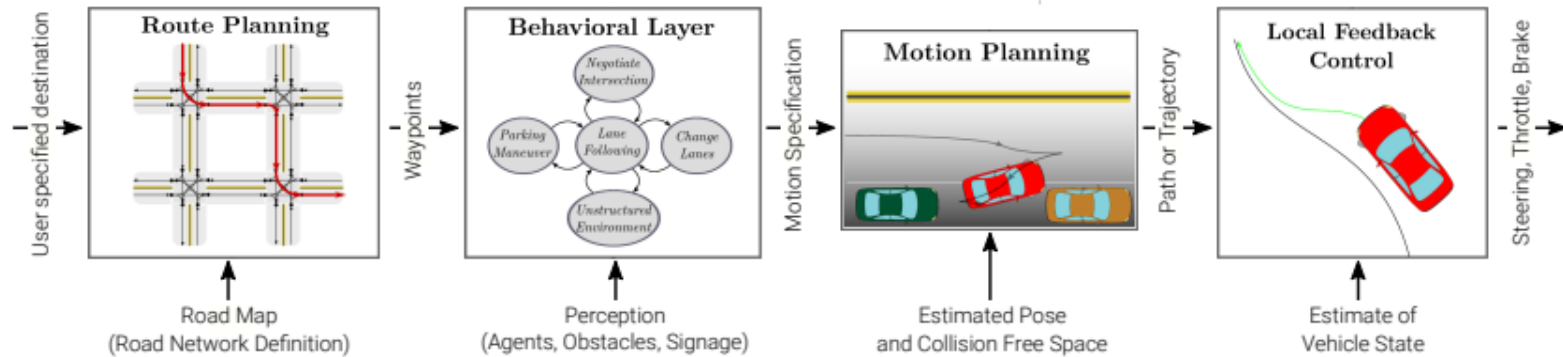
Step 3: Motion Planning



- Find feasible, comfortable, safe and fast vehicle trajectory
- Exact solutions in most cases computationally intractable
- Often numerical approximations are used
- Approaches: variational methods, graph search, incremental tree-based

¹ Note: find feasible not optimal trajectory.

Step 4: Local Feedback Control



- **Feedback controller** executes the trajectory from the motion planner
- Corrects errors due to inaccuracies of the vehicle model
- Emphasis on **robustness, stability and comfort**
- *Open-loop planning plus feedback control* gets robust execution.

¹ Note: find feasible not optimal trajectory.

Behavior Planning

An overall introduction

Behavior Planning

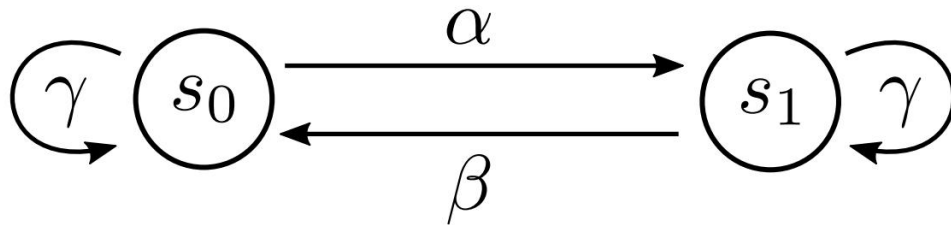
- To follow a planned route, the vehicle must conduct **various maneuvers**
- Examples include: speed tracking, car following, stopping, merging, etc.
- It is difficult to design a motion planner for all maneuvers jointly
- The **behavior planning** stage thus **discretizes the behaviors** into simpler (atomic) maneuvers, each of which can be addressed with a dedicated motion planner
- The behavior layer must take into account traffic rules, static and dynamic objects
- **Input:** High-level route plan and output of perception stack
- **Output:** Motion planner constraints: corridor, objects, speed limits, target, ...
- Frequently used models:
 - Deterministic: Finite State Machines (FSMs) and variants
 - Probabilistic: Markov Decision Processes(RL related)

Finite State Machine

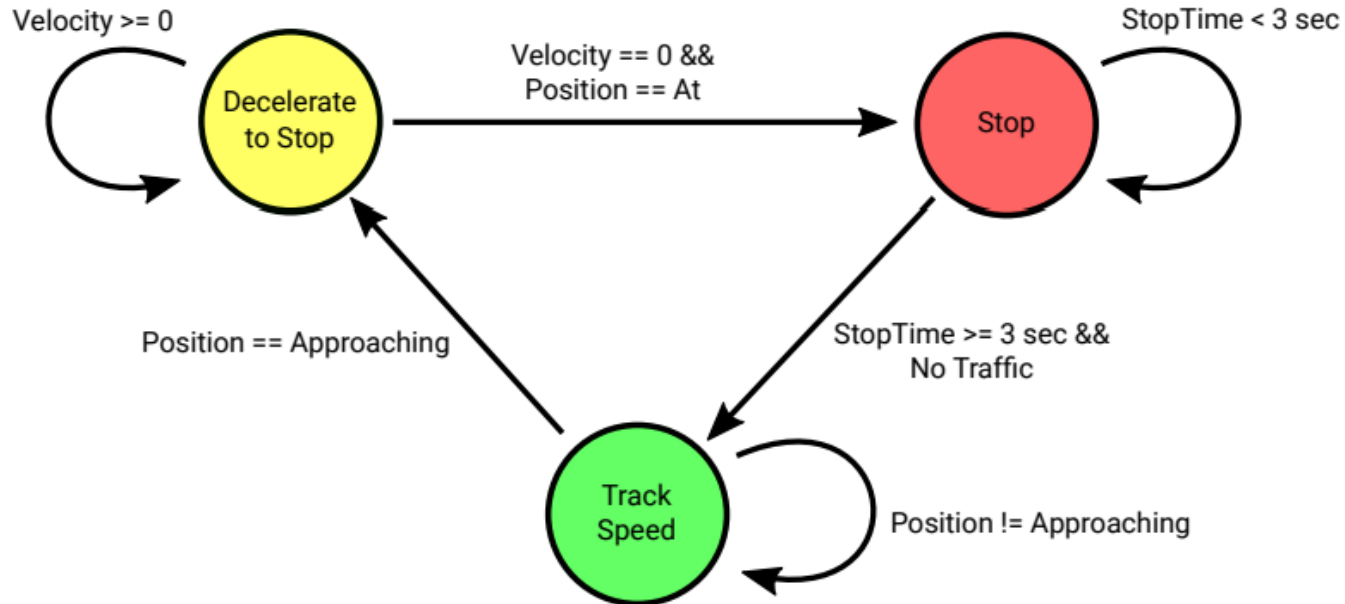
A Finite State Machine (FSM) is defined by quintuple $(\Sigma, S, F, s_0, \delta)$

- Σ is the input alphabet
- S is a non-empty set of states
- $F \subset S$ is the set of final states
- $s_0 \in S$ is the initial state
- $\sigma : S \times \Sigma \rightarrow S$ is the state transition function

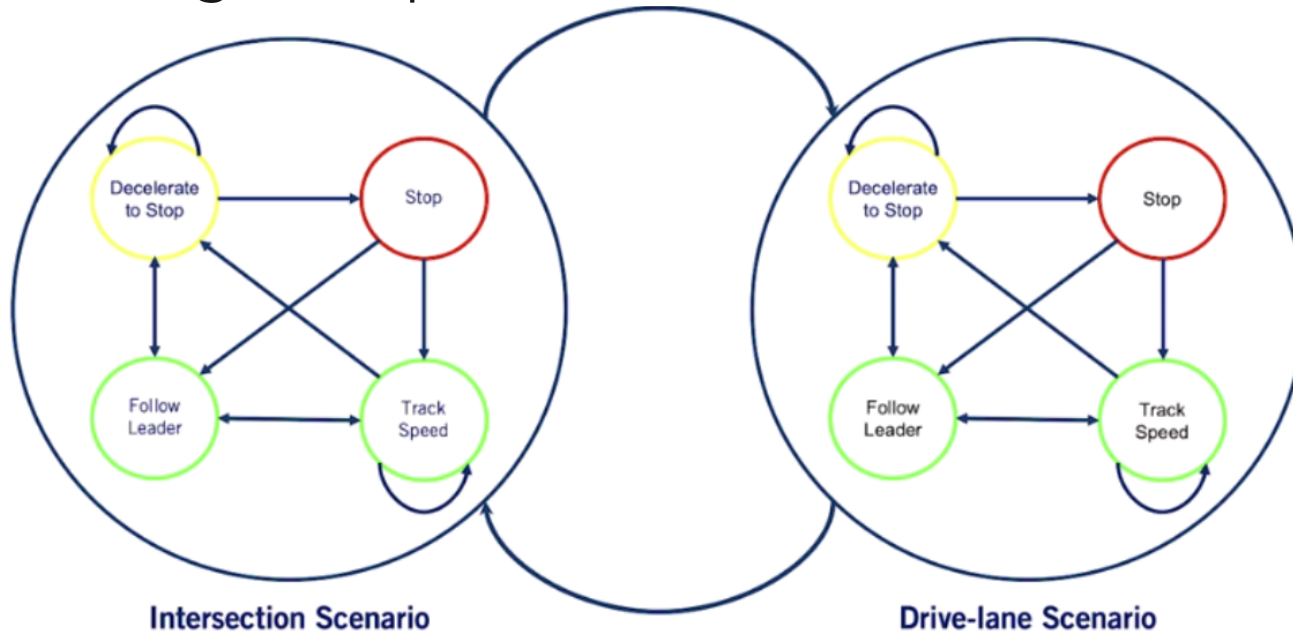
Example:



FSM for a Simple Vehicle Behavior

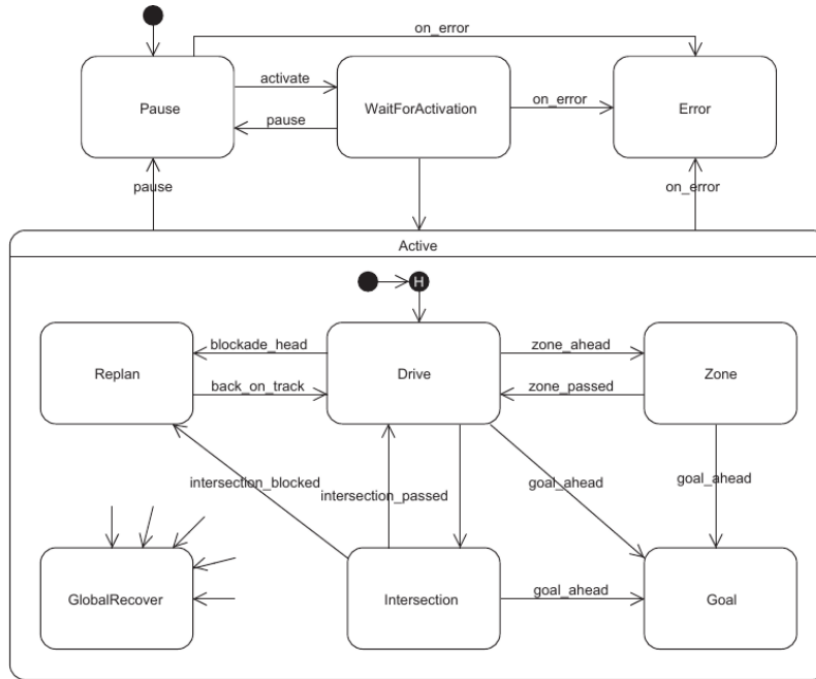


Handling Multiple Scenarios

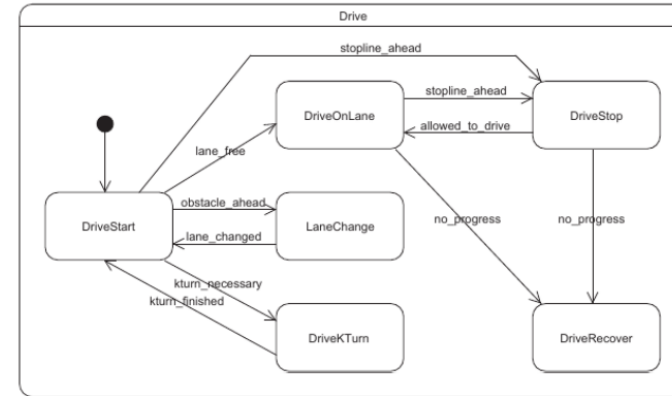


- Hierarchical State Machine (HSM)
- Advantages: Simpler, more efficient – Disadvantages: Rule duplication

Example from DARPA Challenge



(a) State chart of the main level of the FSM.



(b) Substate of the state *Drive*.

¹ Gindele et al.: Design of the planner of team AnnieWAY's autonomous vehicle used in the DARPA Urban Challenge 2007. IV, 2008.

Summary Finite State Machines

- Elegant way to break complex behaviors into simpler maneuvers
- Interpretable and easy to design
- Rule explosion when dealing with complex scenarios
- **Cannot handle noise / uncertainty** → MDPs
- **Expert-designed hyperparameters** → Reinforcement Learning

Widely used in autonomous systems for highway driving(in most cases), not suitable for urban driving.

Note: FSM based decision system is usually denoted as rule-based decision system.

Motion Planning

An overall introduction

Motion Planning

Goal:

- Compute safe, comfortable and feasible trajectory from the vehicle's current configuration to the goal based on the output of the behavioral layer
- Local goal: center of lane a few meters ahead, stop line, parking spot
- Takes as input static and dynamic obstacles around vehicle and generates collision-free trajectory

Focus on Trajectory not only Path

- Path: $\sigma(l) : [0, 1] \rightarrow \mathcal{X}$ (does not specify velocity)
- Trajectory: $\pi(t) : [0, T] \rightarrow \mathcal{X}$ (explicitly considers time)
- **Completeness of planning:** in the configuration space \mathcal{X} of the vehicle and T the planning horizon

Motion Planning

Main Formulations:

- Variational Methods
- Graph Search Methods
- Incremental Search Techniques

Variational Methods

Variational methods minimize a functional:

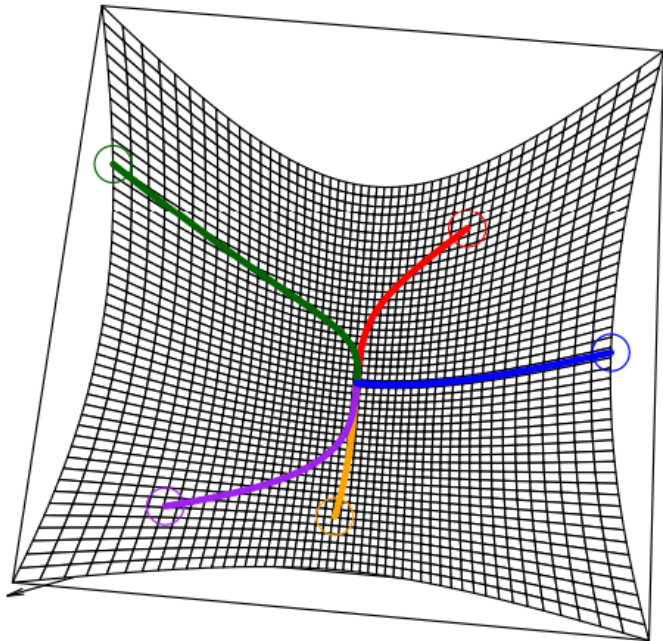
$$\begin{aligned} \operatorname{argmin}_{\pi} J(\pi) &= \int_0^T f(\pi) dt \\ \text{s.t. } \pi(0) &= \mathbf{x}_{\text{init}} \wedge \pi(T) \in \mathbf{X}_{\text{goal}} \end{aligned}$$

- The functional integrates soft constraints (spatial, velocity, jerk, etc.)
- Additional hard constraints can be formulated (minimum turn radius, etc.)
- Solved using numerical optimization
- Often nonconvex problem → **converges slowly to local optimal or even not converge**

Note: that's the reason why we need good hierarchy, namely, if we have better decision inputs to formulate the state space into sub-convex space, the problem can be numerically solved.

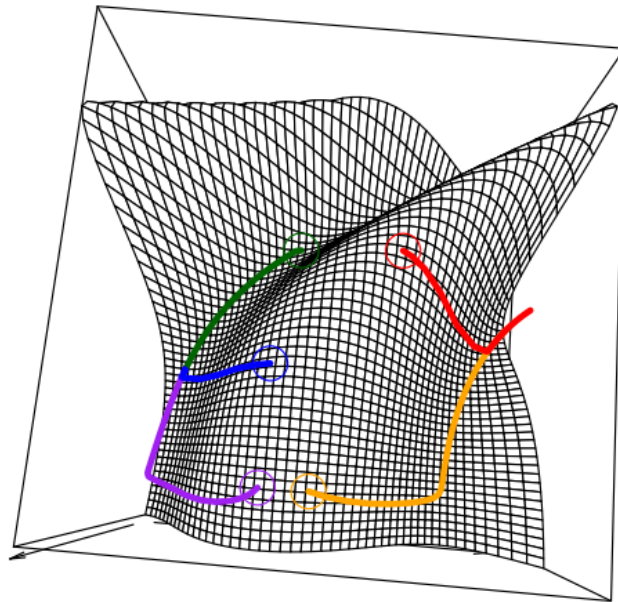
Left

Convex cases can reach global minimum.

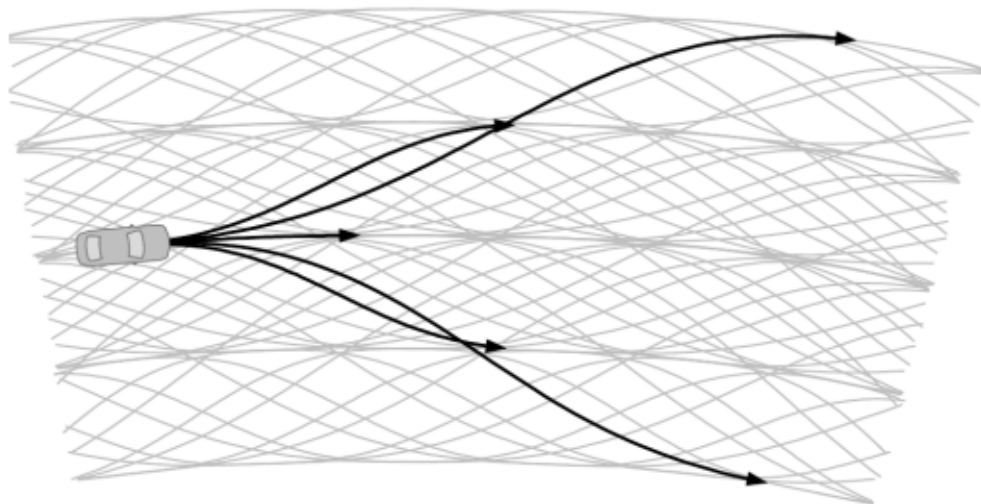
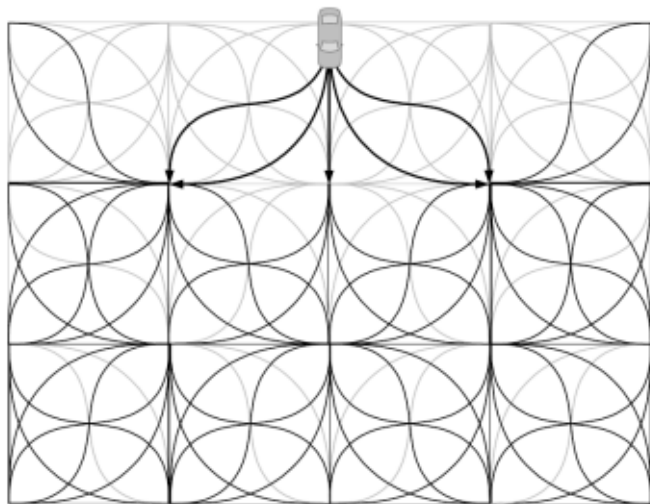


Right

Nonconvex cases only fall into local minimum.

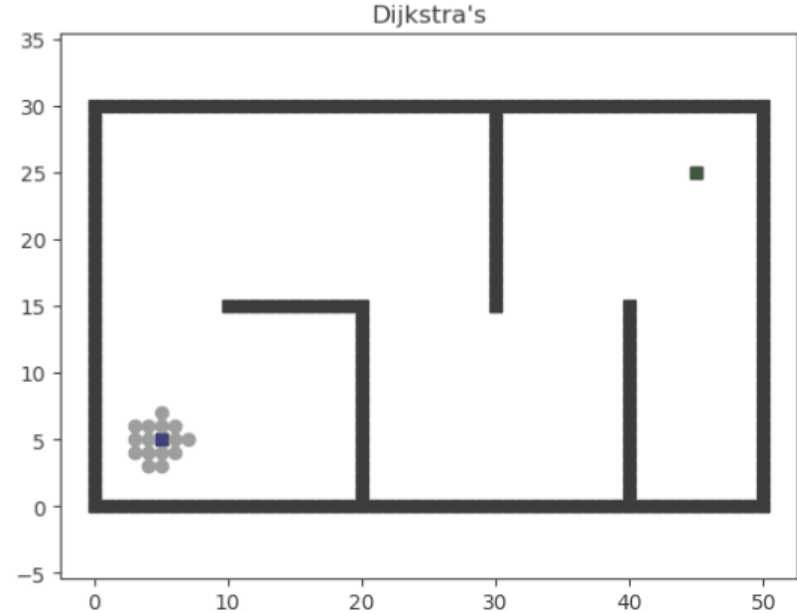
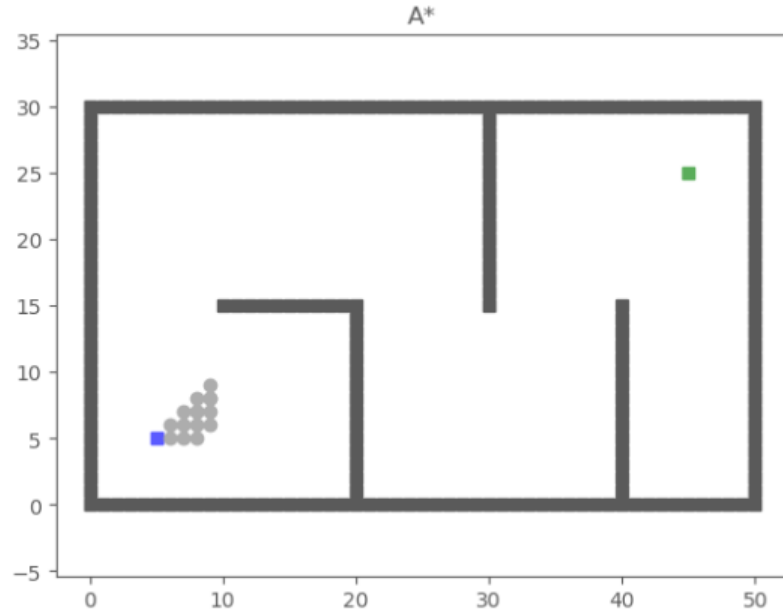


Graph Search Methods

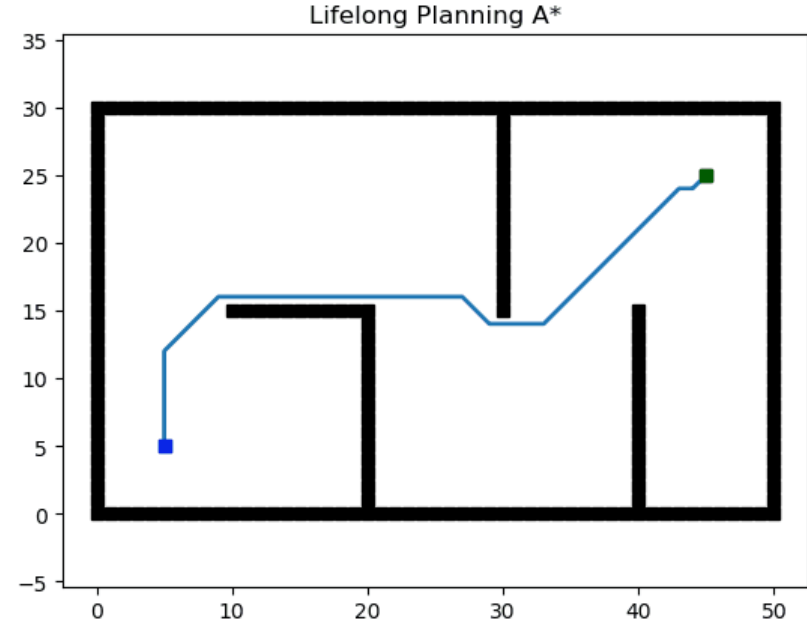
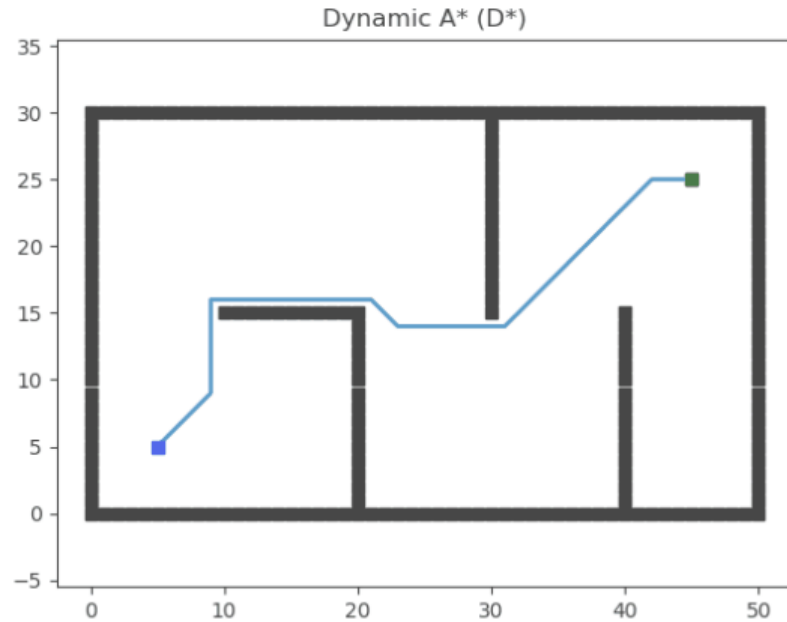


- Idea: Discretize configuration space into graph
- Various algorithms for constructing graphs
- Search strategies: Dijkstra, A^* , ... (like *route planning*)

Graph Search Methods : Examples



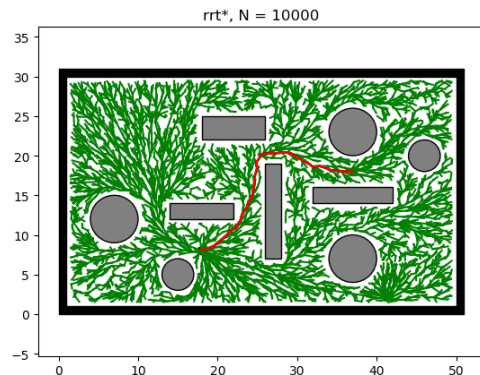
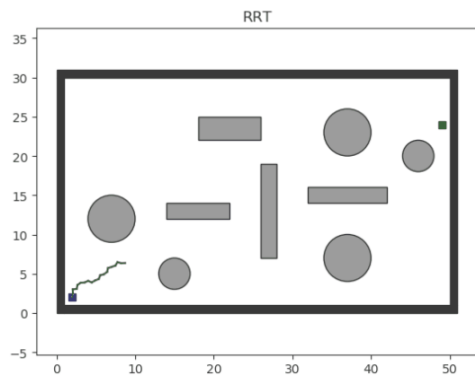
Graph Search Methods : Examples



Incremental Search Techniques

- Idea: Incrementally build increasingly finer discretization of configuration space
- Guaranteed to provide feasible path given enough computation time
- But: computation time can be unbounded
- Prominent example: Rapidly exploring random trees (RRTs)

Note: Not complete and usually suitable for vehicle motion planning.



Hybrid A^*

- Hybrid A^* is an A^* variant that guarantees kinematic feasibility of the path
- Planning is re-applied continuously as the car explores the environment
- A practical method for **kinematic motion planning**

Kinodynamic : Kinematic + Dynamic

- Coarse-to-fine proces
- Trajectory only optimizes locally
- Infeasible path means nothing to nonholonomic system



Practical Search Techniques in Path Planning for Autonomous Driving. STAIR, 2008.¹

Summary

- Driving situations and behaviors are very complex
- Thus, we break the problem into a hierarchy of simpler problems:
 - Route planning, behavior planning and motion planning
 - Each problem is tailored to its scope and level of abstraction
- A* exploits planning heuristics to improve efficiency
- Behavior planning can be implemented using finite state machines
- For motion planning, variational and graph search methods are often used

量产 L2+ 中的决策规划算法实践

以高速领航辅助驾驶(NOP)为例

量产自动驾驶中的决策规划挑战

基于城区领航辅助驾驶(City-NOP)分析