11. Optimization in Smart Cities

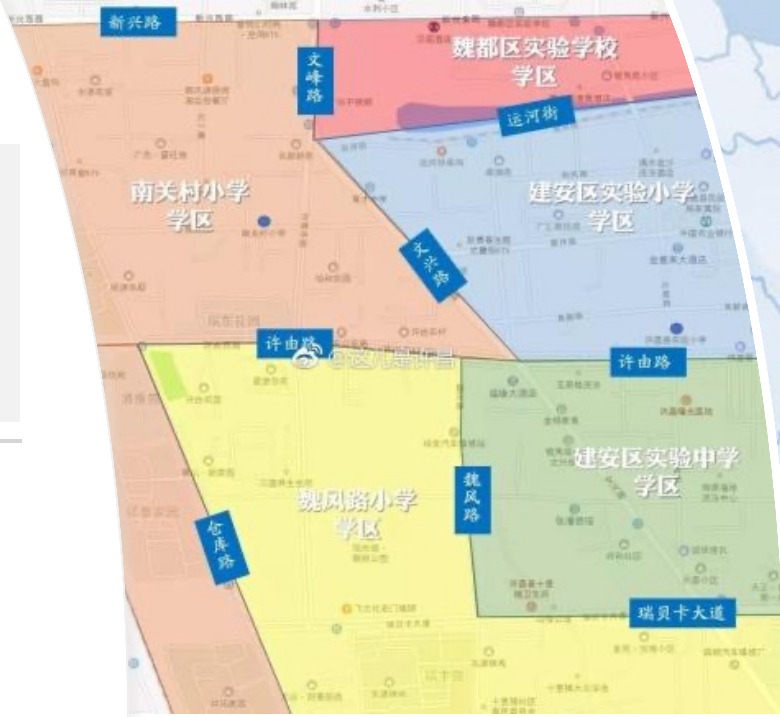
金力 Li Jin
li.jin@sjtu.edu.cn
上海交通大学密西根学院
Shanghai Jiao Tong University UM Joint Institute

Outline

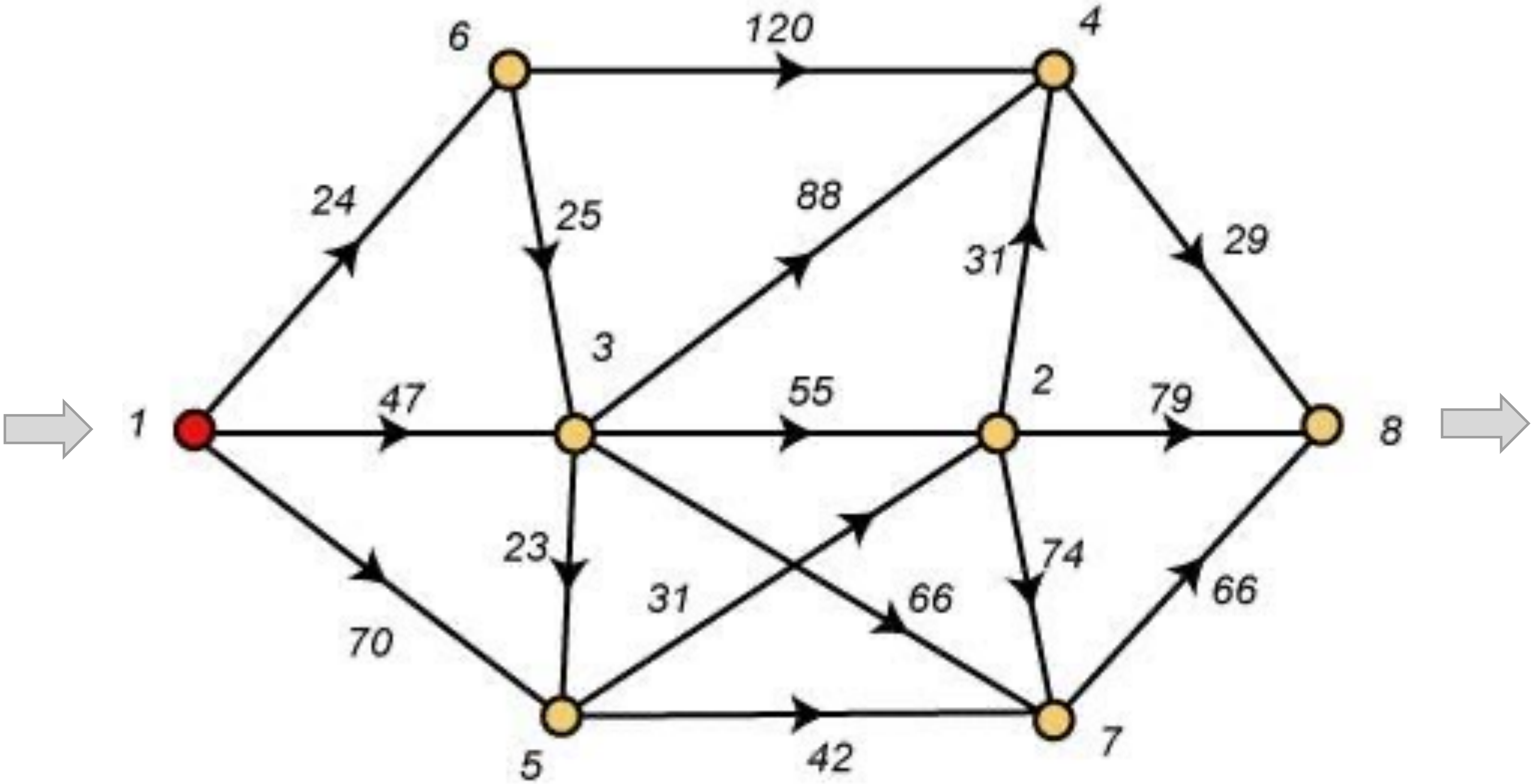
- Optimization problems in smart cities
  - Path planning for vehicles
  - Routing for transportation networks
  - Location of urban facilities
  - Rebalancing of shared bikes
  - Trajectory planning for CAVs
  - Balancing for smart grids
- Optimization problems and methods
  - Linear programming
  - Convex optimization
  - Dynamic programming
- Course project instructions

Optimization
优化

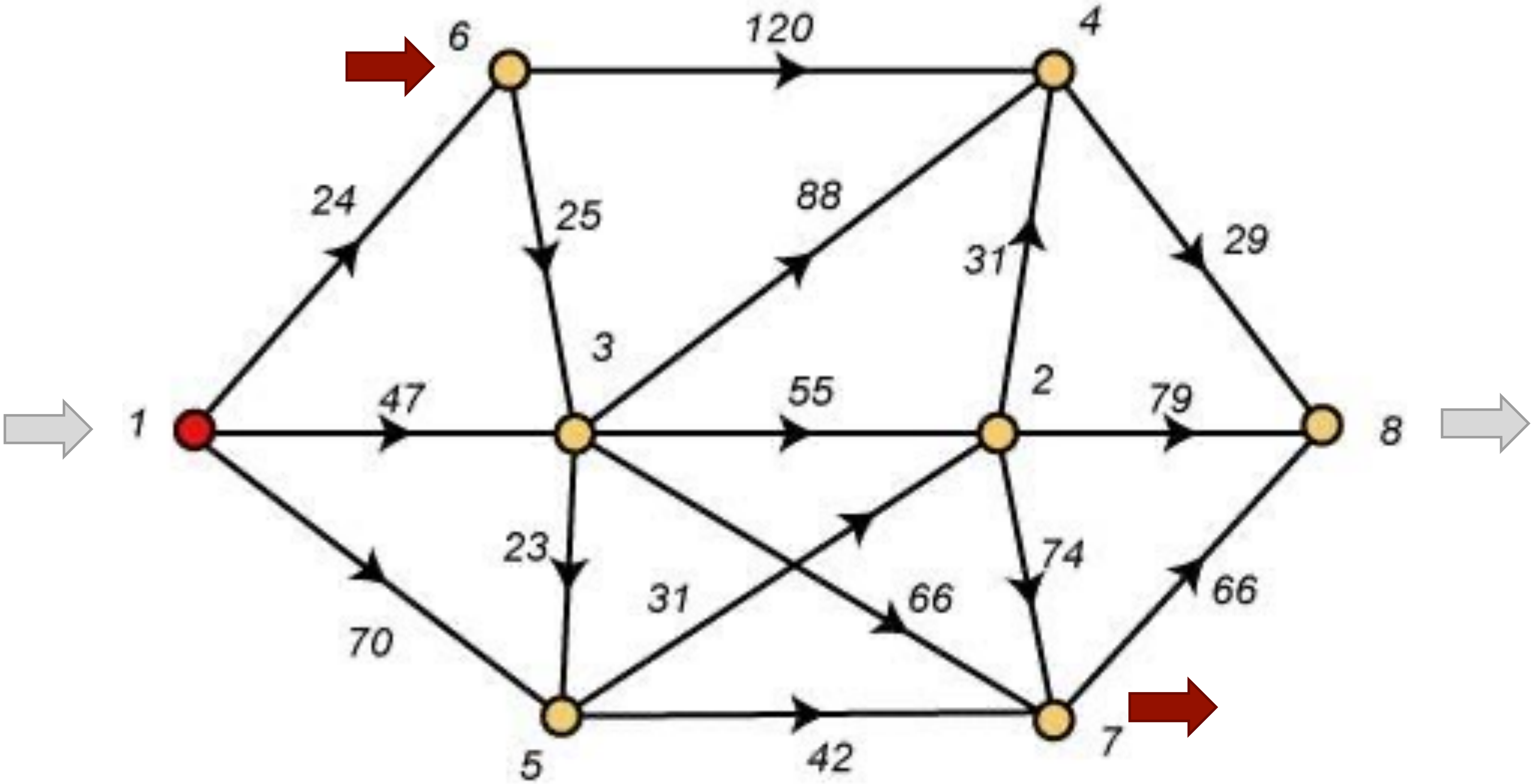
- Typically refers to decision-making at one time.
- That is, once the decision is made, it cannot be changed soon.
- Search the best action or sequence of actions to minimize some cost or maximize some reward.
- Here are examples of one-time actions.



Path planning

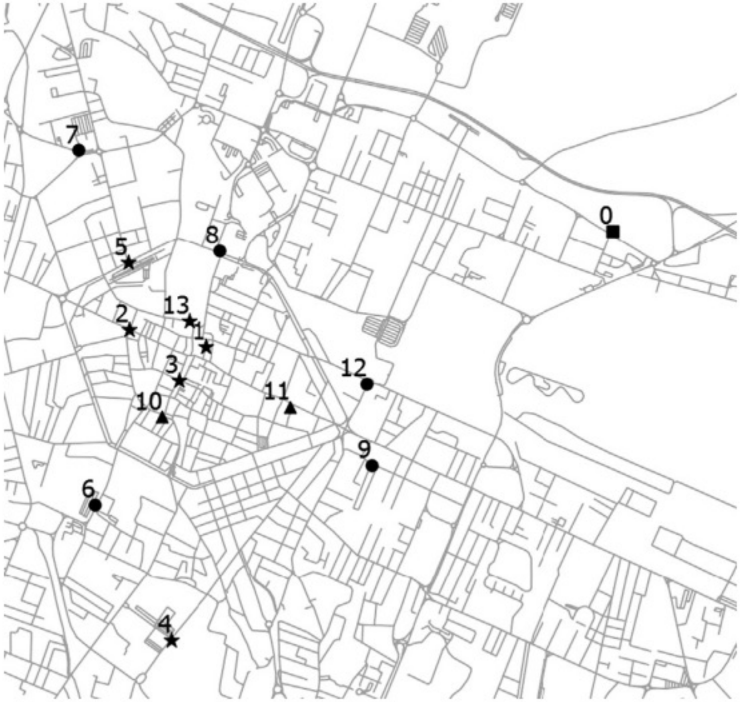


Routing for transportation networks

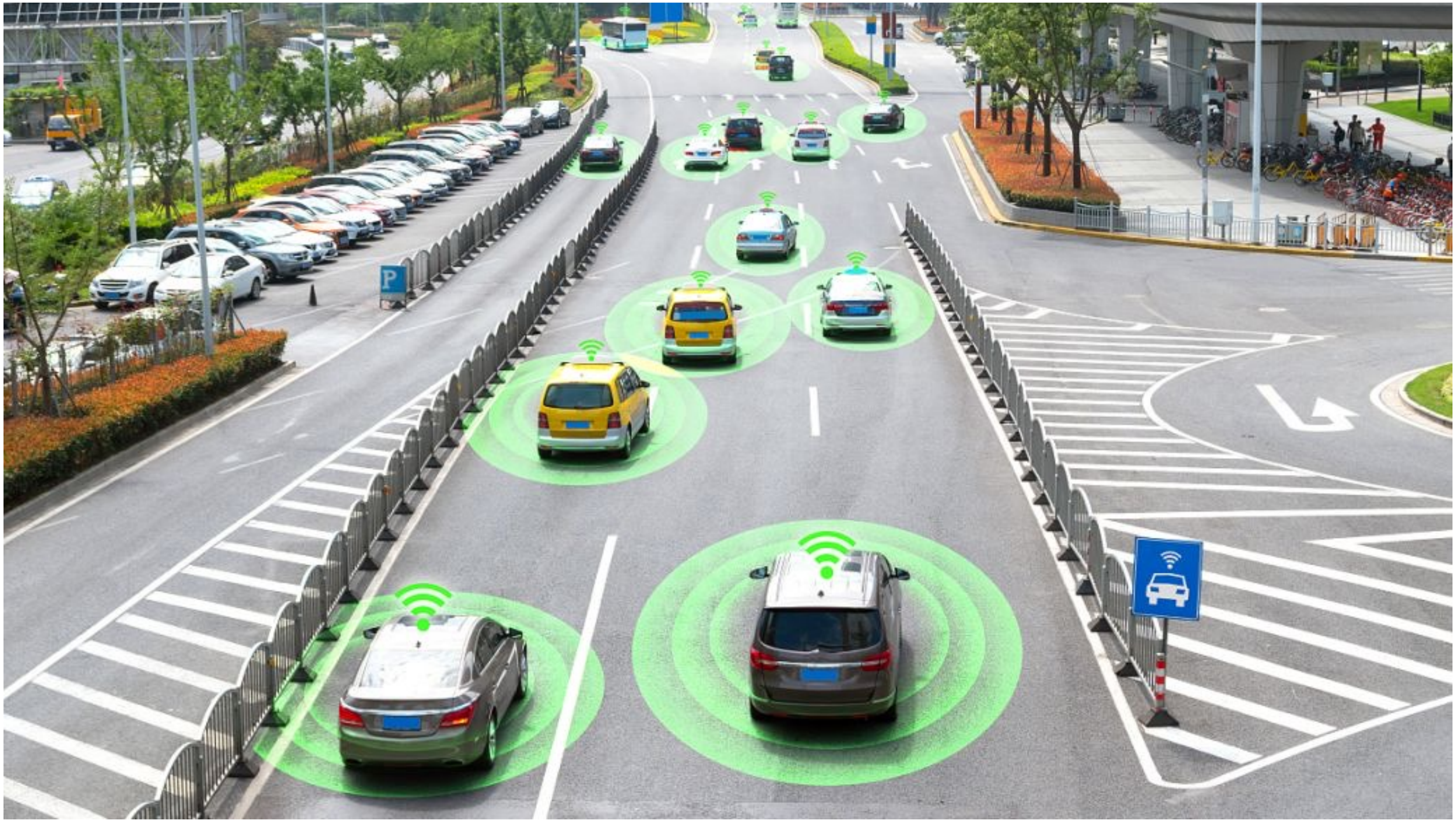


Location of urban facilities

Rebalancing of shared bikes



Trajectory planning for CAVs



Balancing for smart grids

Outline

- Optimization problems in smart cities
  - Path planning for CAVs
  - Routing for transportation networks
  - Location of urban facilities
  - Rebalancing of shared bikes
  - Balancing for smart grids
- Optimization problems and methods
  - Linear programming
  - Convex optimization
  - Dynamic programming
- Course project instructions

Optimization

Architecture of optimization problems:

- Data
  - Prior information that defines the environment in which you make decisions
- Decision variables
  - Quantities that you need to specify/select
- Constraints
  - Conditions that the decision variable must satisfy
- Objective function
  - Criterion you use to evaluate decisions

Standard form

$$\min z = f(x)$$
$$\text{s.t.} \quad g(x) \le 0.$$

- z = objective value.
- $f(x)$ = objective function.
- x = decision variable; continuous or discrete, scalar or vector.
- $g(x) \le 0$: constraints.

Note: equality constraints can always be reformulated as inequality constraints:

$$g(x) = 0 \Leftrightarrow g(x) \ge 0, g(x) \le 0.$$

Linear programming (LP)

$$\min c^T x$$
$$\text{s.t.} \quad Ax \leq b,$$
$$x \in \mathbb{R}^n.$$

- Decision variable: n-dimensional real-valued vector.
- Objective function: linear in x.
- Constraints: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$.
- Both objective function & constraints are linear!
- Simplest optimization problem.
- Comprehensive theory.
- Extensive applications.

Mixed-integer linear programming (MILP)

$$\min c^T x$$
$$\text{s.t.} \quad Ax \leq b,$$
$$x \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}.$$

- Decision variable: n-dimensional mixed-valued vector.
- Objective function: linear in x.
- Constraints: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.
- Significantly more complex than LP
- Very limited theoretical guarantees.
- Extensive applications.

Convex optimization

$$\begin{aligned} &\min z = f(x) \\ &\text{s.t.} \quad g(x) \le 0, \\ &\qquad\quad x \in \mathbb{R}^n. \end{aligned}$$

- $f(x)$ = objective function convex in x.
- **Recall**: $f(x)$ is convex in x if

$$f(\lambda x_1 + (1-\lambda)x_2) \le \lambda f(x_1) + (1-\lambda) f(x_2),$$
$$\forall x \in \mathbb{R}^n, \qquad \forall \lambda \in [0,1].$$

- $g(x)$: convex in x.
- For a problem to be convex:

1. Minimization, convex objective function;
2. Non-positivity, convex left-hand side.

Convex optimization

- Consider an unconstrained convex problem

$$\min z = f(x)$$
$$\text{s.t.} \quad x \in \mathbb{R}^n.$$

- The optimal solution x^* can be obtained by analytically solving the first-order optimality condition

$$\nabla_x f(x) \Big|_{x=x^*} = 0.$$

- Practically, people also use gradient-descent approach to obtain numerical solutions.
- Solution to constrained convex problem is significantly more complex than solving the first-order optimality condition.

Dynamic optimization

- Overlap between control and optimization
- State $s[t]$ with state space $\mathbb{S}$.
- Action $a[t]$ with action space $\mathbb{A}$ or $\mathbb{A}(s[t])$.
- Dynamics:

1. Deterministic: $s[t+1] = f(s[t]);$
2. Stochastic:
   $\Pr\{s[t+1] = s' | s[t] = s, a[t] = a\} = p(s'|s,a)$.

- Reward $r[t]$ depends on $s[t-1]$ and $a[t-1]$.
- Objective:

maximize $\sum_t r[t]$ by selecting $a[0], a[1], \ldots$

Dynamic optimization

- You can indeed consider $a[0], a[1], \ldots$ as individual decision variables and formulate a huge optimization problem:

$$\max_{a[0],a[1],\ldots} \sum_t r[t]$$

$$\text{s.t.} \quad s[t+1] = f(s[t]),\, t = 0,1,\ldots$$

$$r[t] = g(s[t-1], a[t-1]),\, t = 1,2,\ldots$$

$$a[t] \in \mathbb{A}(s[t]),\, t = 0,1,\ldots$$

- Unfortunately, such a brutal-force method may be hard to code and hard to compute, even with state-of-the-art computation capability.
- Need more structural and efficient methods.

Dynamic optimization

There are three typical ways of solving a dynamic optimization problem:

1. Optimal control
   a) Linear dynamics with quadratic reward function
   b) Analytical solution available (linear-quadratic regulation)
2. Dynamic programming
   a) Iteration-based algorithm converging to optimal solution
   b) Requires full knowledge of model parameters & strong computation/storage capability
3. Reinforcement learning
   a) Iteration-based algorithm converging to optimal solution
   b) Synthesis of experimental sampling and dynamic programming

Outline

- Optimization problems in smart cities
  - Path planning for CAVs
  - Routing for transportation networks
  - Location of urban facilities
  - Rebalancing of shared bikes
  - Balancing for smart grids
- Optimization problems and methods
  - Linear programming
  - Convex optimization
  - Dynamic programming
- Course project instructions

Grouping

- 48 students registered
- 12 groups of 4 students
- First round: self-motivated grouping.
  - Send your grouping preferences to Yumeng
  - Partial grouping (i.e., cluster of 2 or 3) is allowed
- Second round: random grouping for the rest of the students.
- Deadline for self-motivated grouping: 7.1
- Random grouping to be published on 7.3

Topic selection

- Option 1: learning-based adaptive cruise control
- Option 2: rebalancing shared bikes on campus
- Option 3: whatever topic you are interested in
- Expectation: simulation, computation, and exploration
- Beyond expectation: theoretical analysis
- Project proposal
  - one-page write-up including title, objective, methodology, and expected results.
  - Due on 7.13 in class

Evaluation

Presentation: 10 min + 5 min Q&A, 5 points of total grade

- 2 students present, and the other 2 students answer questions.
- Presenters are not allowed to answer questions.

Report: 15 points of total grade.

- Will provide word template with recommended font style/size and page format.
- No more than 8 pages, everything included.

Summer 2021 mean: 88%

Peer evaluation

- Opportunity for feedback on group member(s) with inadequate contribution.
- Every one needs to submit a peer evaluation through CANVAS.
- You only need to indicate the team member(s) that contribute(s) **significantly more or significantly less** than the others, along with a brief description.
- Otherwise, just say that "every one contributes about equally".
- This will partially affect the distribution of credits awarded to your team, so think about this carefully before responding.