大数据分析中的算法

文再文

北京大学北京国际数学研究中心

http://bicmr.pku.edu.cn/~wenzw/bigdata2026.html

提纲

课程信息

- 大数据分析中的算法
  侧重数值代数和最优化算法
- 课程代码：00136720 (本科生），00100863 （本研合）
- 教师信息：文再文，wenzw@pku.edu.cn, 微信：wendoublewen
- 助教信息：周瑞松
- 上课地点：二教407
- 上课时间：每周周一3-4节(10:10am - 12:00am)，双周周四1-2节(8:00am - 9:50am)

参考资料

class notes, and reference books or papers

- “Convex optimization”, Stephen Boyd and Lieven Vandenberghe
- “Numerical Optimization”, Jorge Nocedal and Stephen Wright, Springer
- “Optimization Theory and Methods”, Wenyu Sun, Ya-Xiang Yuan
- 文再文，刘浩洋, 户将, 李勇锋，最优化：建模、算法与理论, 高教出版社，第二版，ISBN：9787040550351
- 文再文课题组，大数据分析中的算法（微信群共享链接）

课程计划

侧重数值代数和最优化的模型与算法

- 线性规划，半定规划
- 压缩感知和稀疏优化基本理论和算法
- 低秩矩阵恢复的基本理论和算法
- Optimal transport
- 整数规划
- 随机优化算法
- 随机特征值算法
- 相位恢复
- 强化学习(reinforcement learning)

课程信息

- 教学方式：课堂讲授
- 成绩评定办法：
  - 迟交一天(24小时）打折10%，不接受晚交4天的作业和项目（任何时候理由都不接受）
  - 大作业，包括习题和程序：40%
  - 期中考试：30%
  - 课程项目：30%
  - 作业要求：i) 计算题要求写出必要的推算步骤，证明题要写出关键推理和论证。数值试验题应该同时提交书面报告和程序，其中书面报告有详细的推导和数值结果及分析。ii) 可以同学间讨论或者找助教答疑，但不允许在讨论中直接抄袭，应该过后自己独立完成。iii) 严禁从其他学生，从互联网，从往年的答案，其它课程等等任何途径直接抄袭。iv) 如果有讨论或从其它任何途径取得帮助，请列出来源。
- 请谨慎选课

提纲

机器学习：监督学习简介

- 机器学习，人工智能
  - 计算机视觉，自然语言处理(ChatGPT)
  - 围棋：AlphaGo, AlphaGo Zero
- 科学发现经常围绕建立函数关系：$f : X \to Y$
  - 蛋白质结构预测(Alphafold2): X: 蛋白质序列; Y: 三维结构
  - 深度势能: X: 分子结构; Y: 原子势能
- 挑战: f 非常高维、高度非线性，只有它的很少量已知的知识或者计算非常昂贵
- 机遇: 我们有很多的数据:

$$\{(x_i, y_i) \mid x_i \in X, y_i \in Y, 1 \leq i \leq N\}.$$

- 机器学习是构造 $\hat{f} \approx f$ 的强大工具

监督学习中典型问题形式

机器学习构造$\hat{f}$的典型方式：$\hat{f} = h(x, \theta)$

$$\min_{\theta \in \mathcal{W}} \quad \frac{1}{N} \sum_{i=1}^{N} \|x_i^\top \theta - y_i\|_2^2 + \mu\varphi(\theta) \quad \text{线性回归}$$

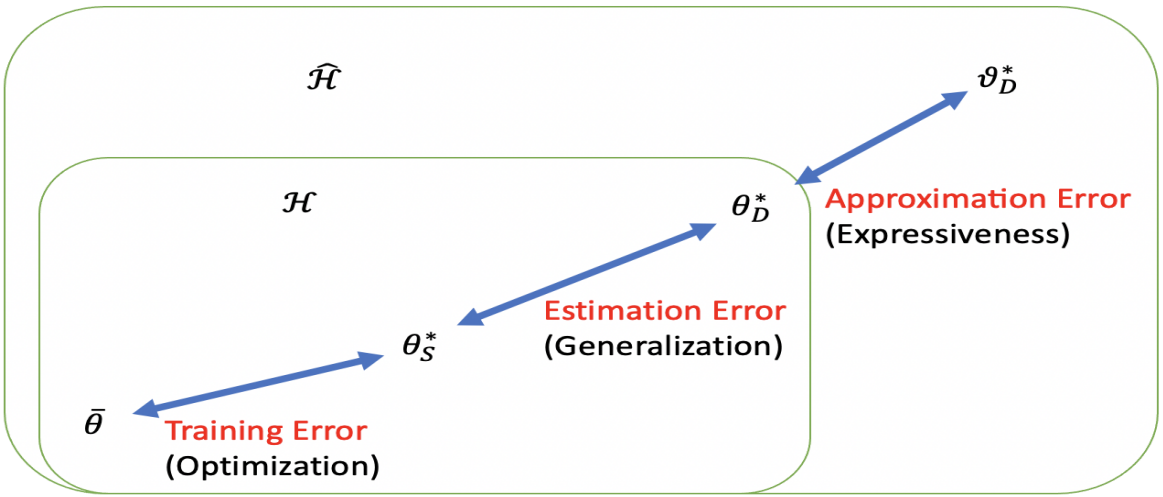
$$\min_{\theta \in \mathcal{W}} \quad \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i x_i^\top \theta)) + \mu\varphi(\theta) \quad \text{逻辑回归}$$

$$\min_{\theta \in \mathcal{W}} \quad \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i, \theta), y_i) + \mu\varphi(\theta) \quad \text{一般形式}$$

- (x_i, y_i)是给定的数据对，y_i是数据x_i对应的标签
- $\ell_i(\cdot)$: 度量模型拟合数据点i的程度(避免拟合不足)
- $\varphi(\theta)$: 避免过拟合的正则项: $\|\theta\|_2^2$或者$\|\theta\|_1$等等
- $h(x, \theta)$: 线性函数、决策树/森林或者由深度神经网络构造的模型

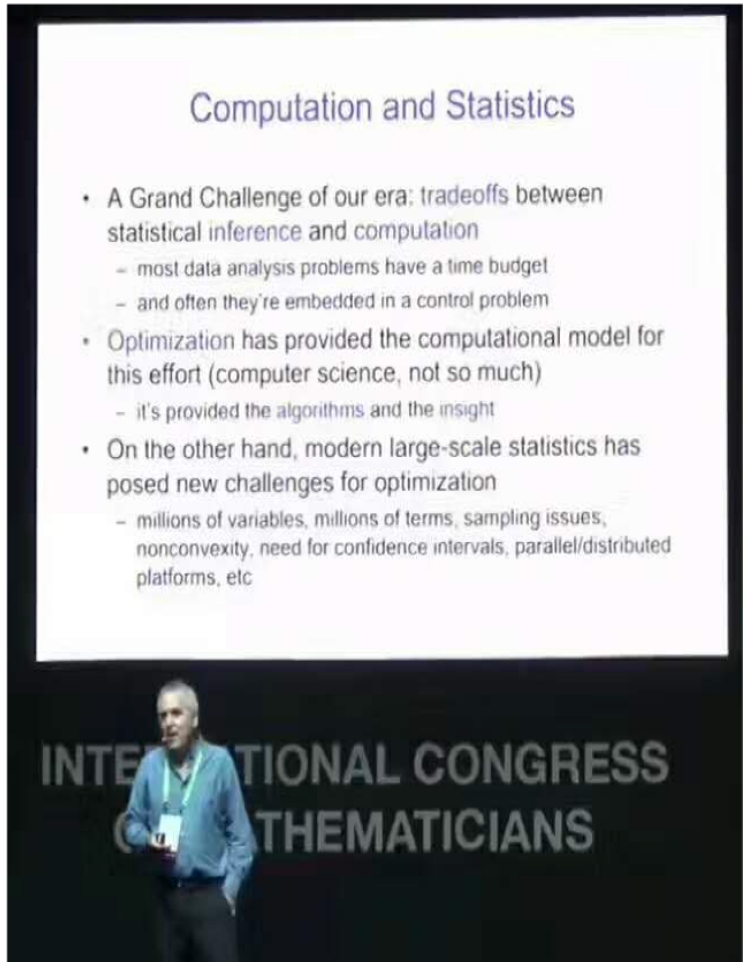
机器学习：表达能力，泛化，优化

- ground truth: $\vartheta_D^* = \underset{\vartheta \in \widehat{\mathcal{H}}}{\operatorname{argmin}} \mathbf{E}[\ell(h(x,\theta),y)]$
- optimal hypothesis: $\theta_D^* = \arg\min_{\theta \in \mathcal{H}} \mathbf{E}[\ell(h(x,\theta),y)]$
- empirical optimal hypothesis: $\theta_S^* = \underset{\theta \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i,\theta),y_i)$
- returned hypothesis: $\bar{\theta}$



美国三院院士Michael Jordan教授国际数学家大会一小时报告

- 我们时代的一个很大挑战是统计推理和计算的平衡。大部分数据分析有时间限制，他们经常被嵌入到某个控制问题里。
- 最优化为这个努力提供了计算模型，给出了算法和深刻的理解。
- 现代大规模统计给优化带来了新的挑战：百万量级变量/函数项，抽样问题，非凸，置信区间，并行/分布式平台等等



"Statistics and the Oncoming AI Revolution"

- What has made ML so successful? What are the disciplines supporting ML and providing a good basis to understand the challenges, open problems, and limitations of the current techniques?
  1) **basic statistical tools**: linear models, generalized linear models, logistic regression, cross validation, overfitting . . .
  2) **probability theory and probabilistic modeling**.
- How about engineering disciplines?
  Clearly, progress in optimization—particularly in convex optimization—has fueled ML algorithms for the last two decades.

美国科学院院士Emmanuel Candès

通用最优化模型和算法

最优化问题一般可以描述为

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned}$$

- 按照目标函数和约束函数的形式来分：
  线性规划/非线性规划、凸优化/非凸优化、非光滑优化、半定规划、锥规划、整数规划、无导数优化、几何优化、稀疏优化、低秩矩阵优化、张量优化、鲁棒优化、全局优化、组合优化、网络规划、随机优化、动态规划、带微分方程约束优化、微分流形约束优化、分布式优化等

- 具体应用涵盖：
  运筹学、供应链管理、物流管理、资产管理、统计学习、压缩感知、最优运输、信号处理、图像处理、机器学习、强化学习、模式识别、金融工程、电力系统等领域

提纲

稀疏优化

$$(\ell_0)\begin{cases}\min\limits_{x\in\mathbb{R}^n} & \|x\|_0,\\ \text{s.t.} & Ax=b.\end{cases}\qquad (\ell_2)\begin{cases}\min\limits_{x\in\mathbb{R}^n} & \|x\|_2,\\ \text{s.t.} & Ax=b.\end{cases}\qquad (\ell_1)\begin{cases}\min\limits_{x\in\mathbb{R}^n} & \|x\|_1,\\ \text{s.t.} & Ax=b.\end{cases}$$

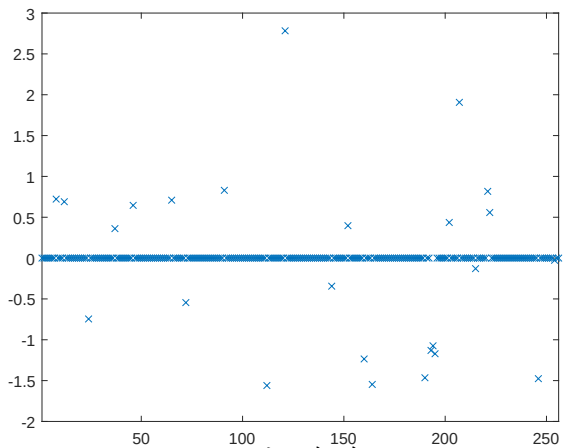
- 其中$\|x\|_0$是指x 中非零元素的个数. 由于$\|x\|_0$是不连续的函数，且取值只可能是整数，ℓ_0问题实际上是NP难的，求解起来非常困难.
- 若定义ℓ_1范数：$\|x\|_1=\sum_{i=1}^n|x_i|$，我们得到了另一个形式上非常相似的问题，又称$\ell_1$范数优化问题，基追踪问题
- ℓ_2范数：$\|x\|_2=\left(\sum_{i=1}^n x_i^2\right)^{1/2}$

稀疏优化

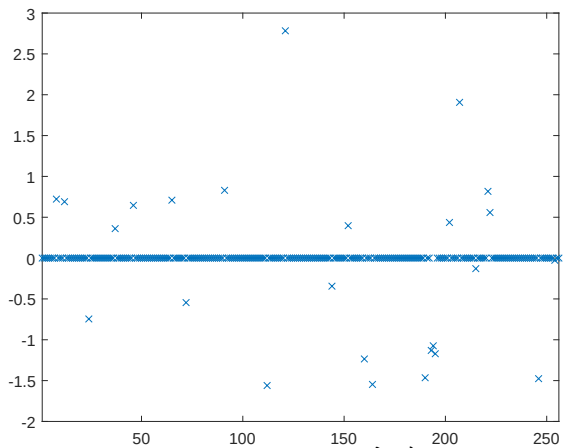
在MATLAB环境里构造A, u 和b：

```
m = 128; n = 256;
A = randn(m, n);
u = sprandn(n, 1, 0.1);
b = A * u;
```

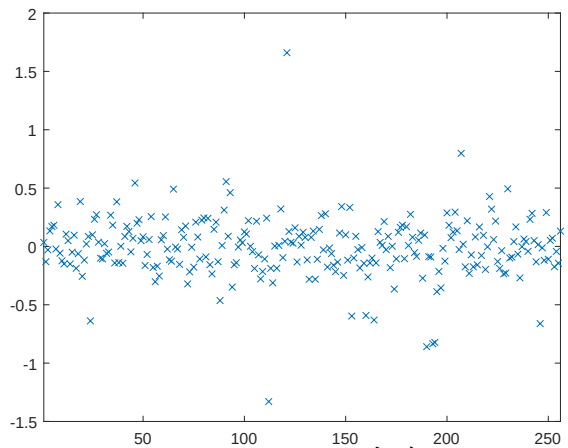
构造一个128×256矩阵A，它的每个元素都服从高斯（Gauss）随机分布．精确解u只有10%的元素非零，每一个非零元素也服从高斯分布



(a) 精确解u



(b) ℓ_1问题的解



(c) ℓ_2问题的解

Figure: 稀疏优化的例子

低秩矩阵恢复

- 某视频网站提供了约48万用户对1万7千多部电影的上亿条评级数据，希望对用户的电影评级进行预测，从而改进用户电影推荐系统，为每个用户更有针对性地推荐影片.

- 显然每一个用户不可能看过所有的电影，每一部电影也不可能收集到全部用户的评级. 电影评级由用户打分1 星到5 星表示，记为取值1~5 的整数. 我们将电影评级放在一个矩阵M中，矩阵M的每一行表示不同用户，每一列表示不同电影. 由于用户只对看过的电影给出自己的评价，矩阵M中很多元素是未知的

	电影1	电影2	电影3	电影4	…	电影n
用户1	4	?	?	3	…	?
用户2	?	2	4	?	…	?
用户3	3	?	?	?	…	?
用户4	2	?	5	?	…	?
⋮	⋮	⋮	⋮	⋮		⋮
用户m	?	3	?	4	…	?

低秩矩阵恢复

- 令Ω是矩阵M中所有已知评级元素的下标的集合，则该问题可以初步描述为构造一个矩阵X，使得在给定位置的元素等于已知评级元素，即满足$X_{ij} = M_{ij},\ (i,j) \in \Omega$.
- 低秩矩阵恢复（low rank matrix completion）

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \operatorname{rank}(X), \\ \text{s.t.} \quad & X_{ij} = M_{ij},\ (i,j) \in \Omega. \end{aligned}$$

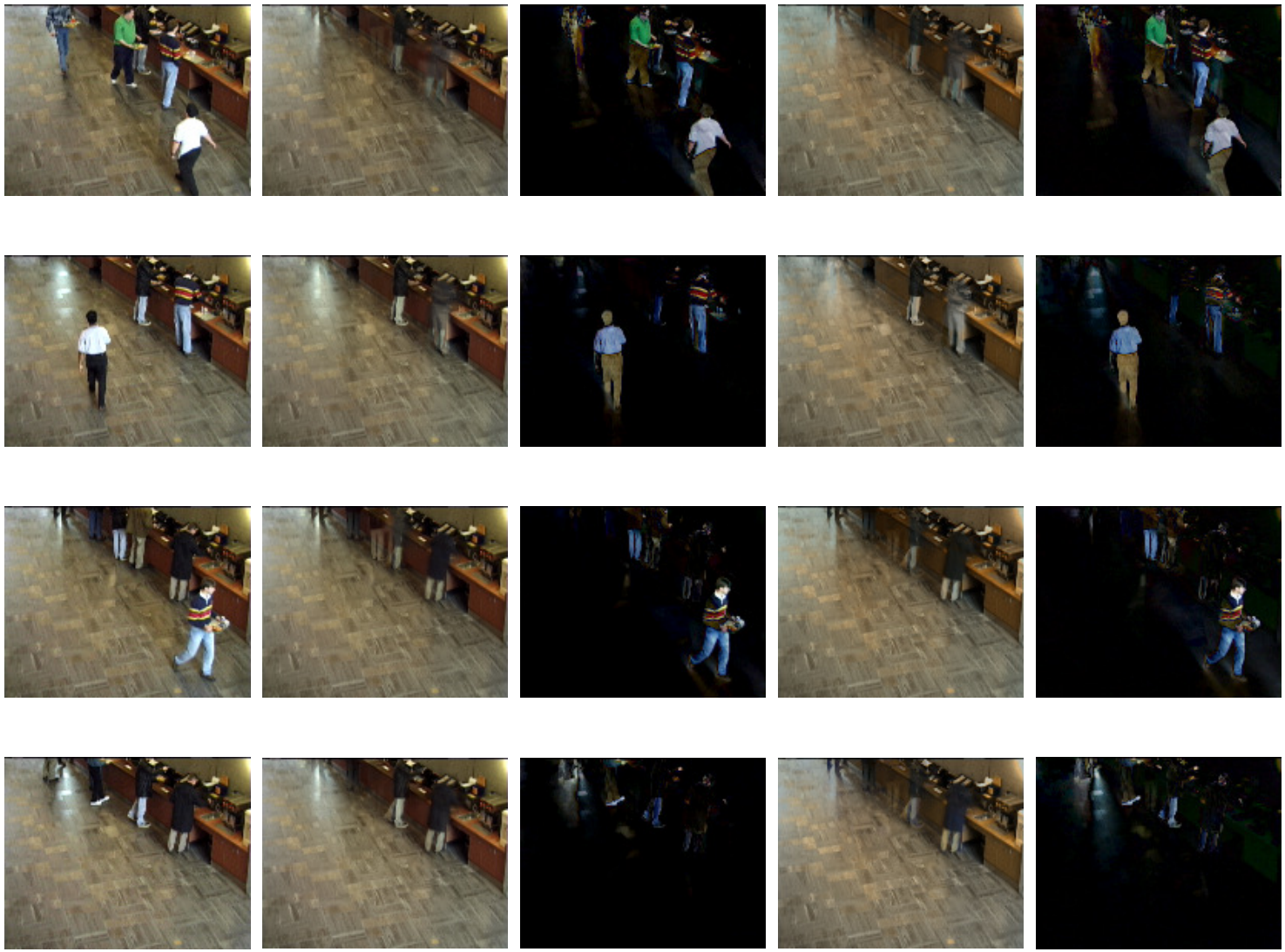
$\operatorname{rank}(X)$正好是矩阵X所有非零奇异值的个数

- 矩阵X的核范数（nuclear norm）为矩阵所有奇异值的和，即：$\|X\|_* = \sum_i \sigma_i(X)$:

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \|X\|_*, \\ \text{s.t.} \quad & X_{ij} = M_{ij},\ (i,j) \in \Omega. \end{aligned}$$

视频分离

- 将视频分成移动和静态的部分



稀疏和低秩矩阵分离

- 给定矩阵M，我们想找到一个低秩矩阵W和稀疏矩阵E，使得

$$W + E = M.$$

- 非凸模型：

$$\begin{aligned} \min_{W,E\in\mathbb{R}^{m\times n}} \quad & \operatorname{rank}(W) + \mu\|E\|_0, \\ \text{s.t.} \quad & W + E = M. \end{aligned}$$

- 凸松弛：

$$\min_{W,E} \ \|W\|_* + \mu\|E\|_1, \ \text{s.t.}\ W + E = M$$

- 鲁棒主成分分析

提纲

卷积神经网络Convolutional neural network (CNN)

- 给定二维图像$\boldsymbol{I} \in \mathbb{R}^{n\times n}$和卷积核$\boldsymbol{K} \in \mathbb{R}^{k\times k}$，定义卷积操作$\boldsymbol{S} = \boldsymbol{I} * \boldsymbol{K}$，它的元素是

$$\boldsymbol{S}_{i,j} = \langle \boldsymbol{I}(i : i+k-1, j : j+k-1), \boldsymbol{K} \rangle,$$

其中两个矩阵X, Y的内积是它们相应元素乘积之和

- 生成的结果$\boldsymbol{S}$可以根据卷积核的维数、$\boldsymbol{I}$的边界是否填充、卷积操作时滑动的大小等相应变化.

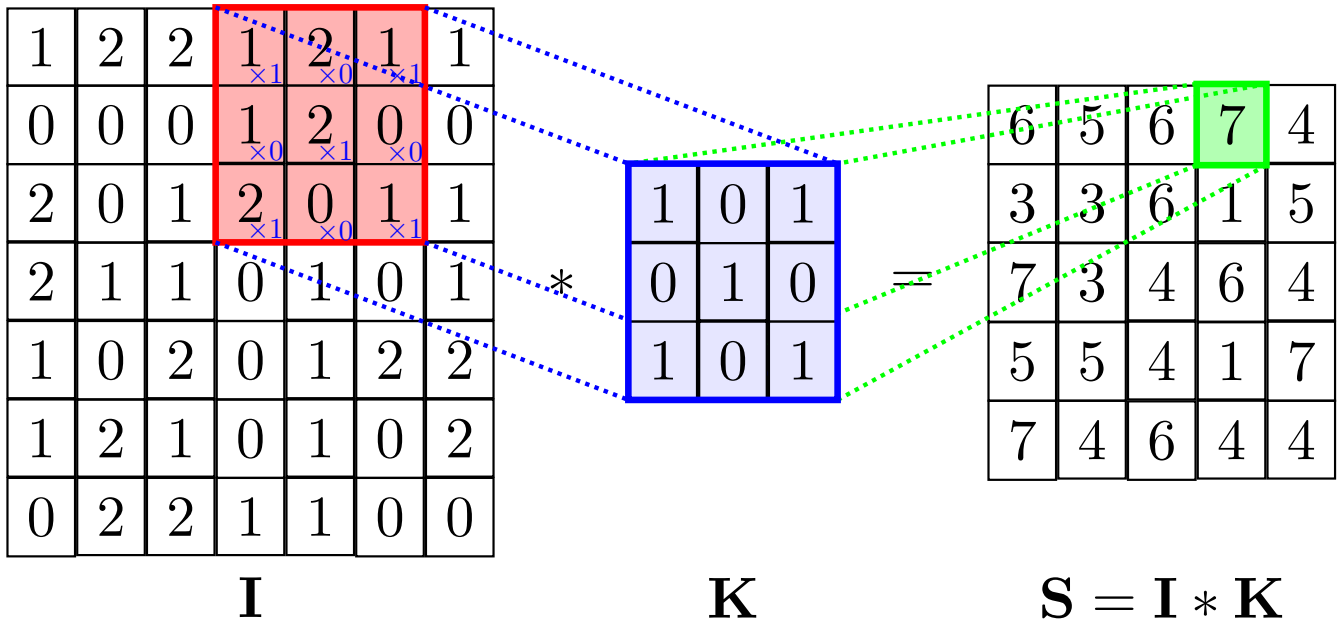
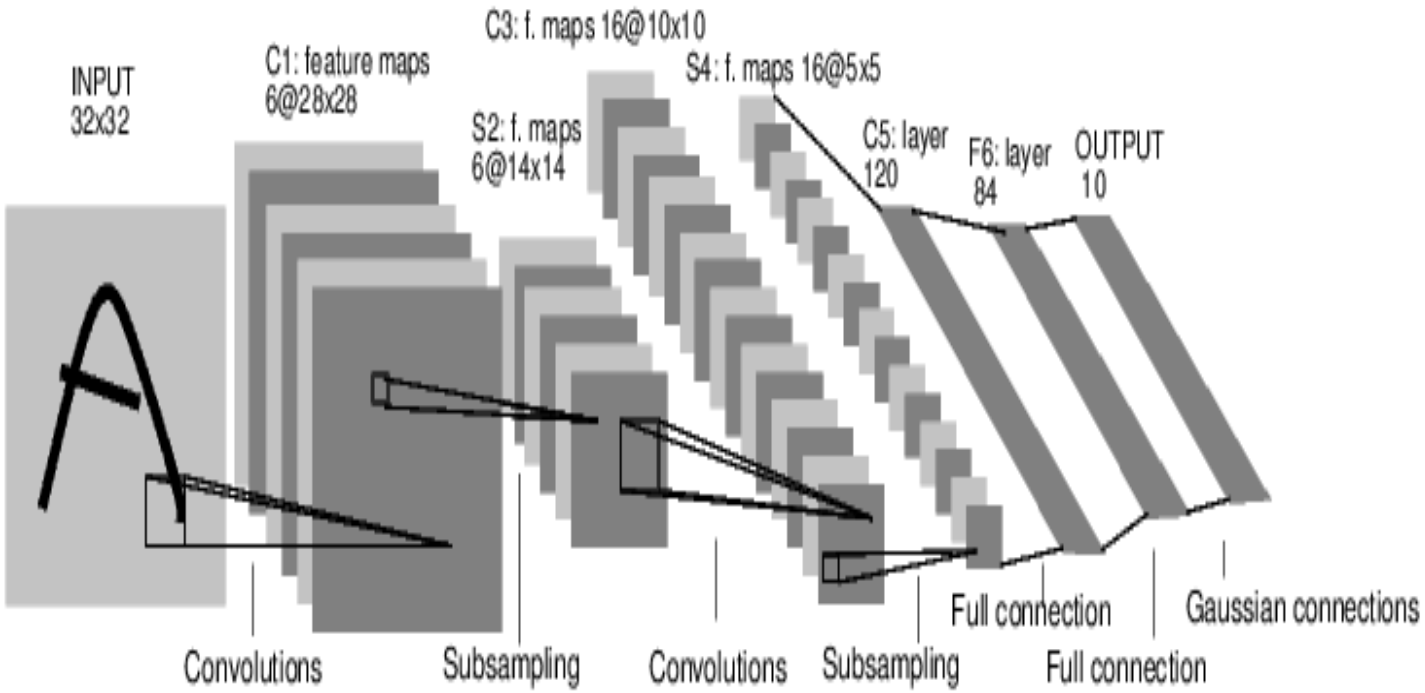


Figure: 卷积操作

卷积神经网络Convolutional neural network (CNN)

LeCun等人开创性的建立了数字分类的神经网络。几家银行使用它来识别支票上的手写数字。



Schrödinger equation

- The N-electron Schrödinger equation

$$H\Psi = E\Psi,$$

where H is the molecular Hamiltonian operator, Ψ is a N-electron antisymmetric wave function.

- Curse of dimensionality: computational work goes as 10^{3N}.
- Kohn-Sham total energy minimization (Manifold Optimization):

$$\min_{X^*X=I} E_{KS}(X) := E_{kinetic}(X) + E_{ion}(X) + E_{Hartree}(X) + E_{xc}(X),$$

where $\nabla E_{KS}(X) = H(X)X$.

- Kohn-Sham equation:

$$H(X)X = X\Lambda, \quad X^*X = I$$

FermiNet: 特征值优化问题+深度神经网络+SGD

- Minimizing the following Rayleigh quotient:

$$E_0 = \min_{\Psi} E[\Psi] = \frac{\int \Psi^*(\mathbf{r})\hat{H}\Psi(\mathbf{r})\mathrm{d}\mathbf{r}}{\int \Psi^*(\mathbf{r})\Psi(\mathbf{r})\mathrm{d}\mathbf{r}}$$

- Variational Quantum Monte Carlo:

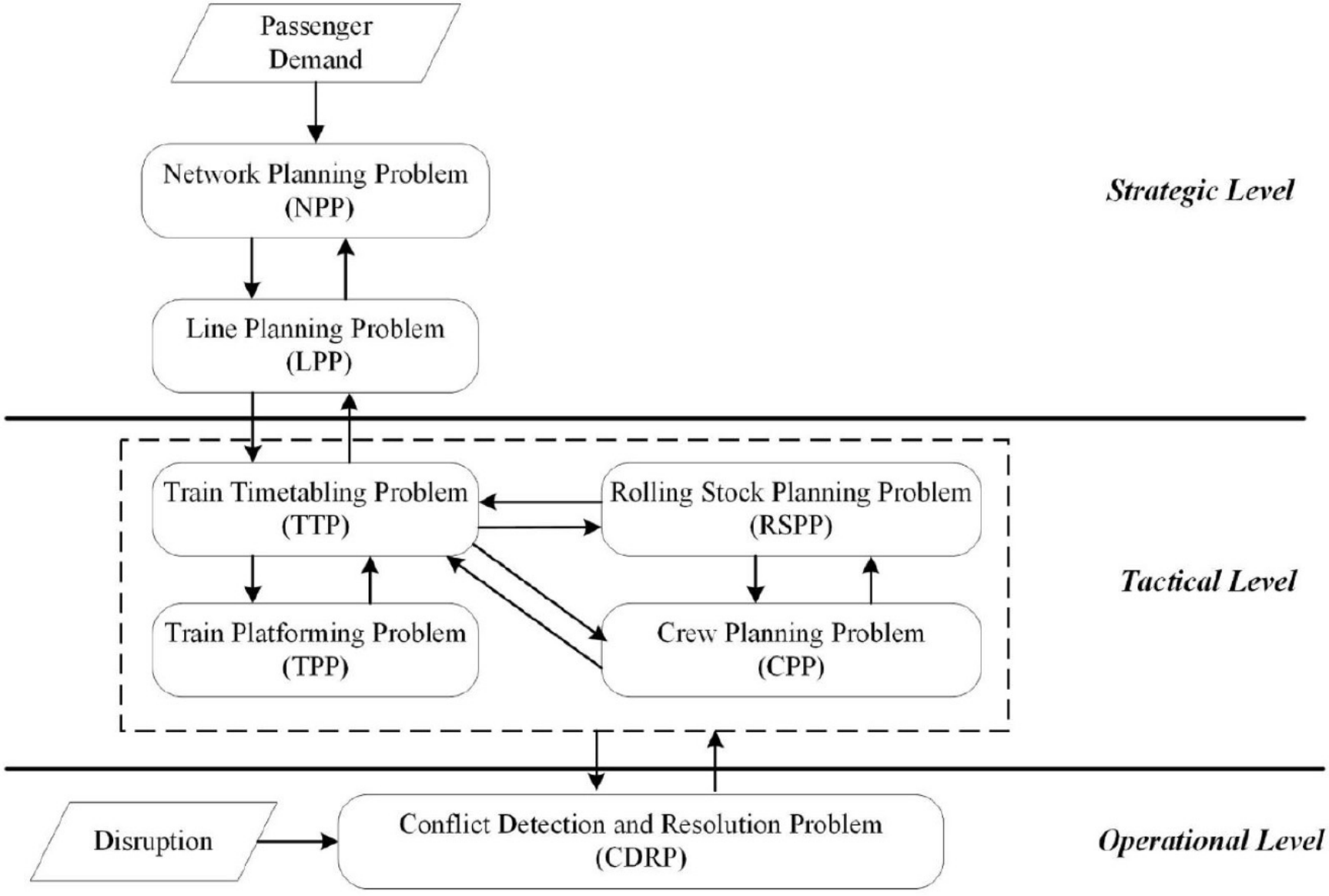
$$\begin{aligned}\min_{\theta} \mathcal{L}(\theta) &= \frac{\int \Psi_\theta^*(\mathbf{r})\hat{H}\Psi_\theta(\mathbf{r})\mathrm{d}\mathbf{r}}{\int \Psi_\theta^*(\mathbf{r})\Psi_\theta(\mathbf{r})\mathrm{d}\mathbf{r}} \\ &= \int \frac{|\Psi_\theta(\mathbf{r})|^2}{\int |\Psi_\theta(\mathbf{r})|^2 \mathrm{d}\mathbf{r}} \left(\Psi_\theta^{-1}(\mathbf{r})\hat{H}\Psi_\theta(\mathbf{r})\right) \mathrm{d}\mathbf{r} \\ &= \mathbb{E}_{P_\theta(\mathbf{r})}\left[E_L(\mathbf{r})\right] \Longrightarrow \text{ solved by SGD}\end{aligned}$$

- Local Energy: $E_L(\mathbf{r}) = \Psi_\theta^{-1}(\mathbf{r})\hat{H}\Psi_\theta(\mathbf{r})$,
  Probability distribution: $P_\theta(\mathbf{r}) = \frac{|\Psi_\theta(\mathbf{r})|^2}{\int |\Psi_\theta(\mathbf{r})|^2 \mathrm{d}\mathbf{r}}$

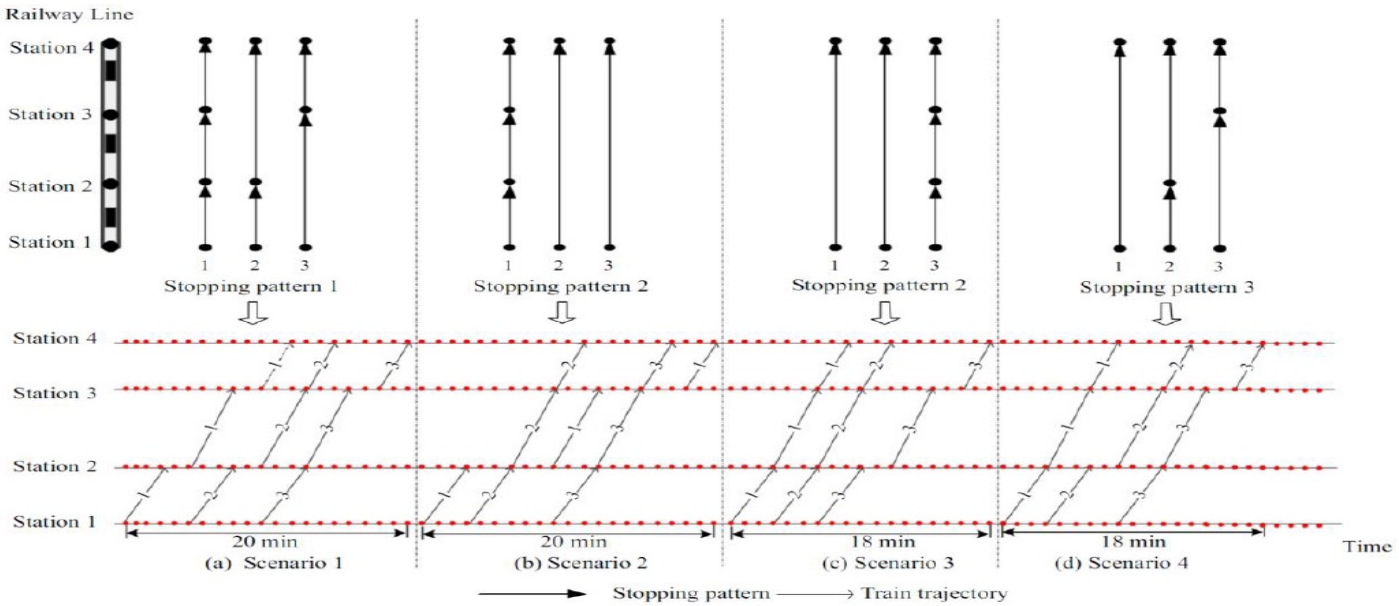
提纲

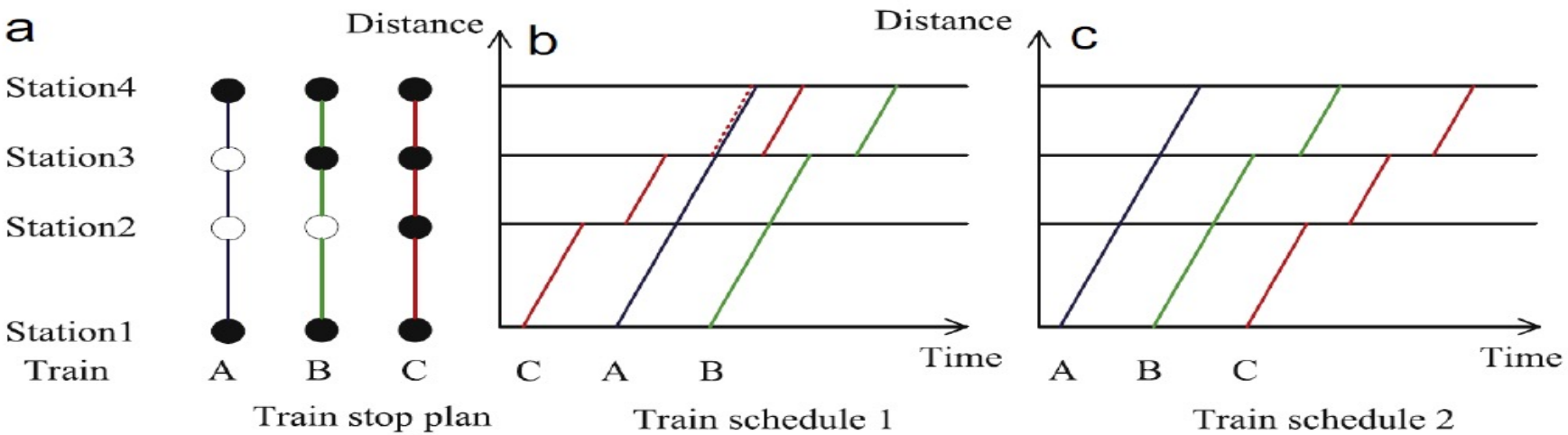
铁路运行图的地位

致谢：北京交通大学乐逸祥教授团队

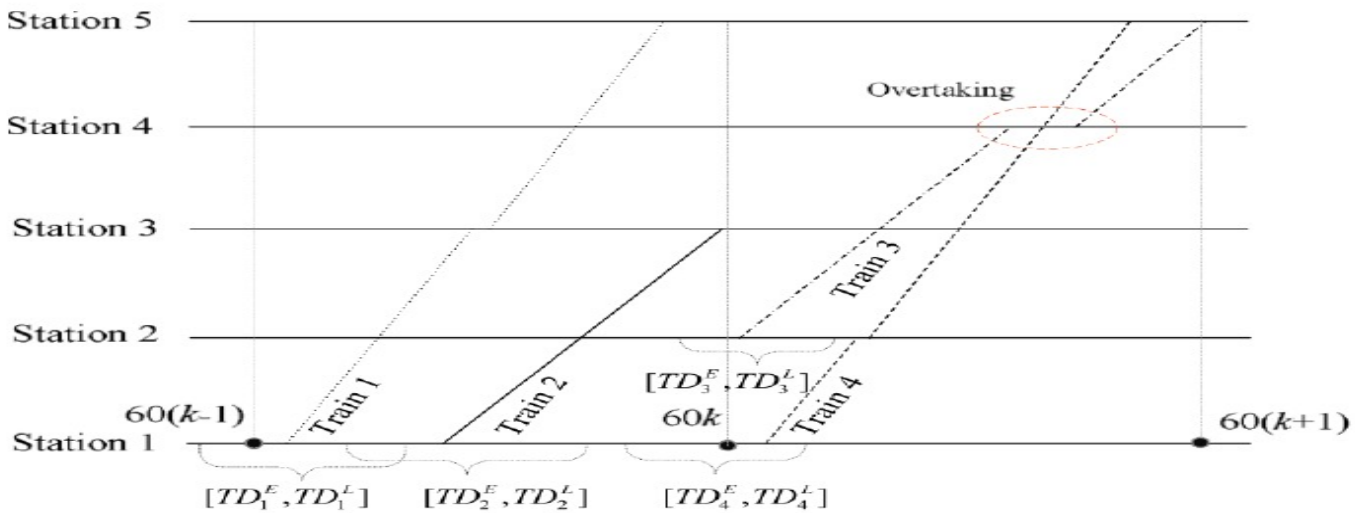


开行方案

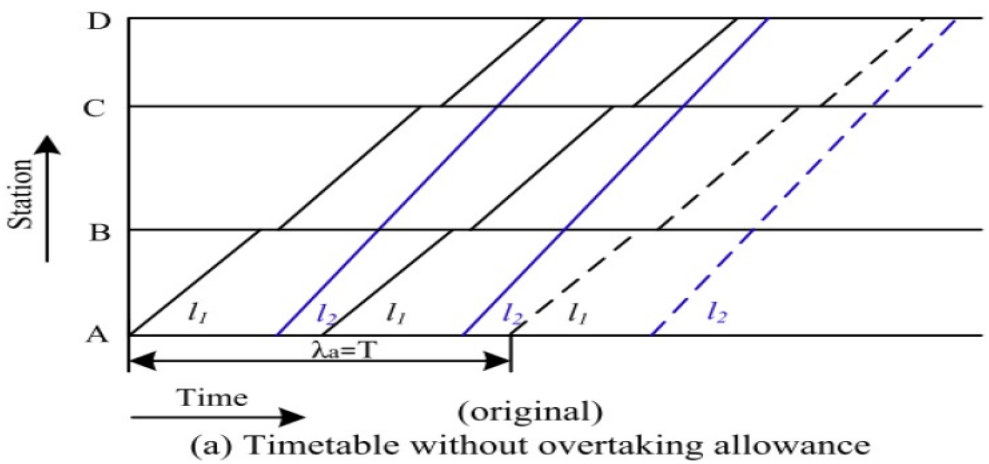

Railway Line
Station 4
Station 3
Station 2
Station 1
1 2 3
Stopping pattern 1
Stopping pattern 2
Stopping pattern 2
Stopping pattern 3
20 min
(a) Scenario 1
20 min
(b) Scenario 2
18 min
(c) Scenario 3
18 min
(d) Scenario 4
Time
Stopping pattern
Train trajectory


a
Station4
Station3
Station2
Station1
Train
A B C
Train stop plan
Distance
b
C A B
Time
Train schedule 1
Distance
c
A B C
Time
Train schedule 2

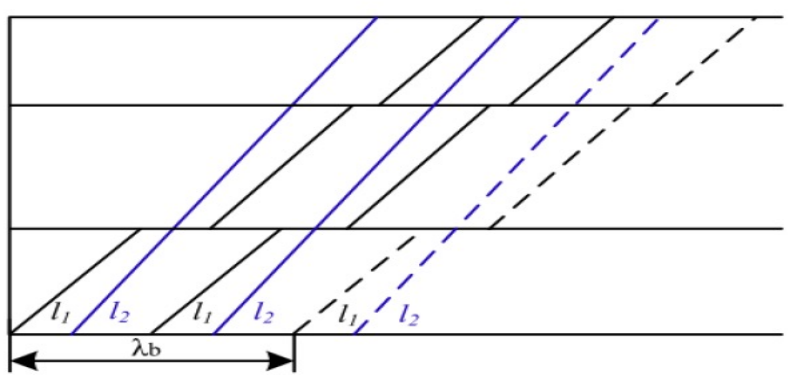
铁路运行图难点：速差、列车越行



(i) Traditional train departure time window

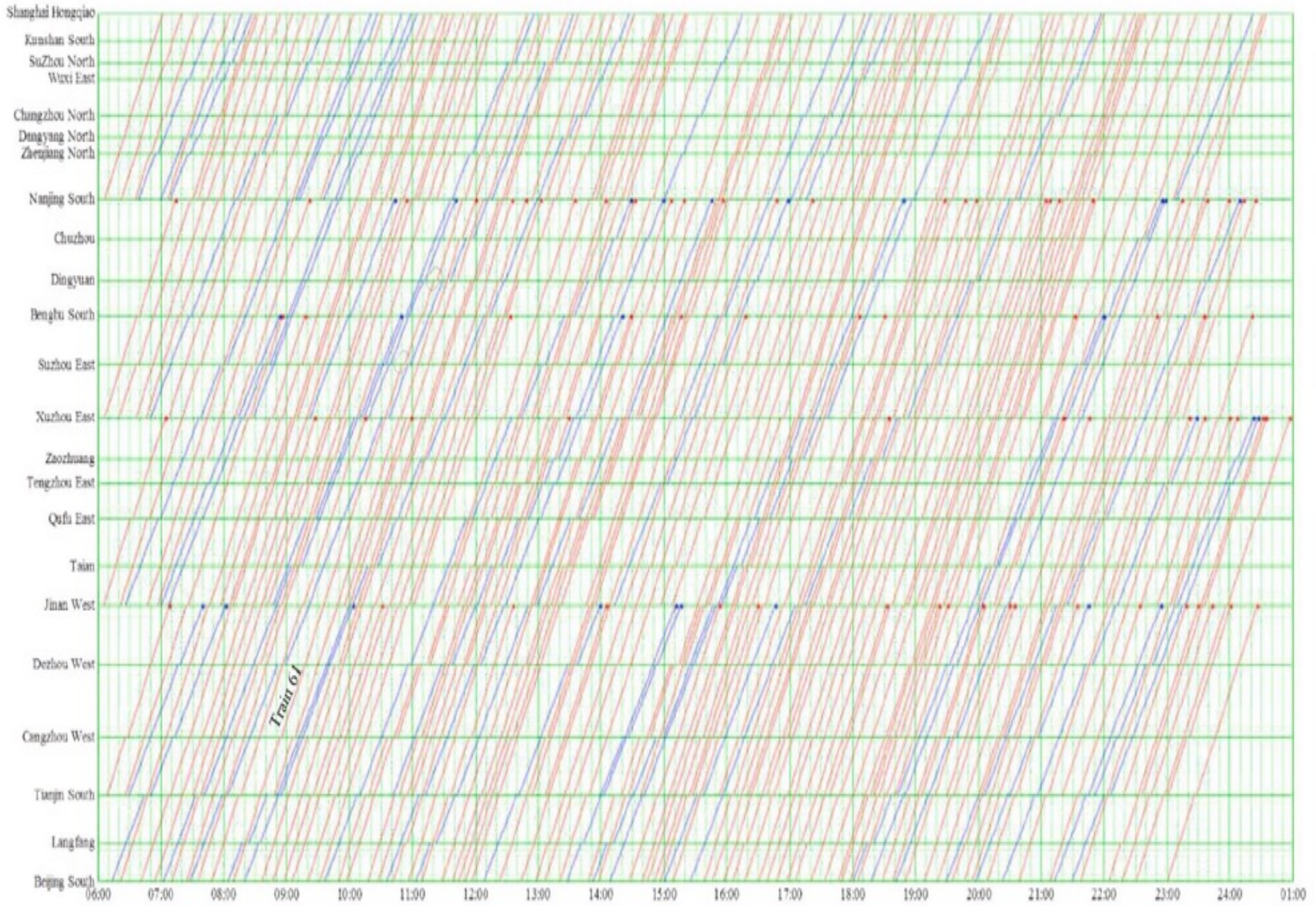


(a) Timetable without overtaking allowance



(b) Timetable with overtaking allowance

铁路运行图难点：高密度

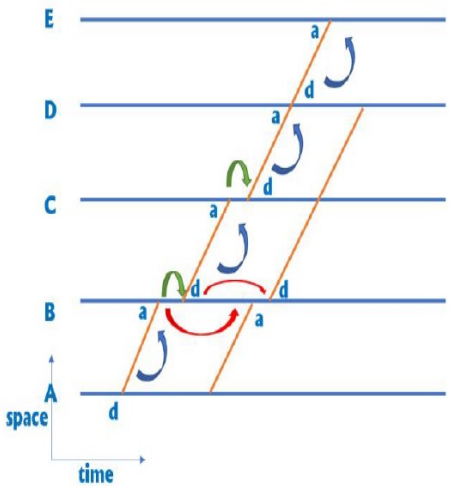

Shanghai Hongqiao
Kunshan South
SuZhou North
Wuxi East
Changzhou North
Dangyang North
Zhenjiang North
Nanjing South
Chuzhou
Dingyuan
Bengbu South
Suzhou East
Xuzhou East
Zaozhuang
Tengzhou East
Qufu East
Taian
Jinan West
Dezhou West
Train 61
Cangzhou West
Tianjin South
Langfang
Beijing South
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
24:00
01:00

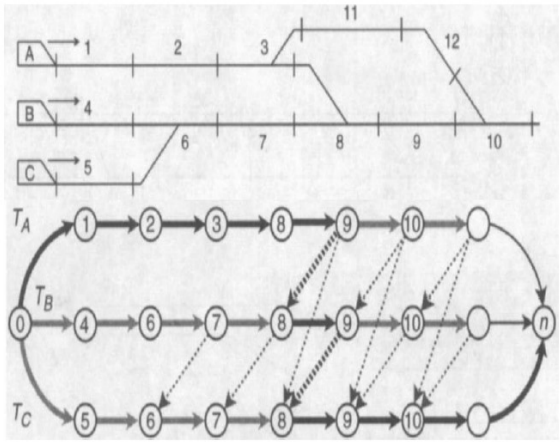
铁路运行图建模

难点：列车前后关系的确定，MILP

基于列车事件（到达、出发事件）进行建模

1. Job-shop like



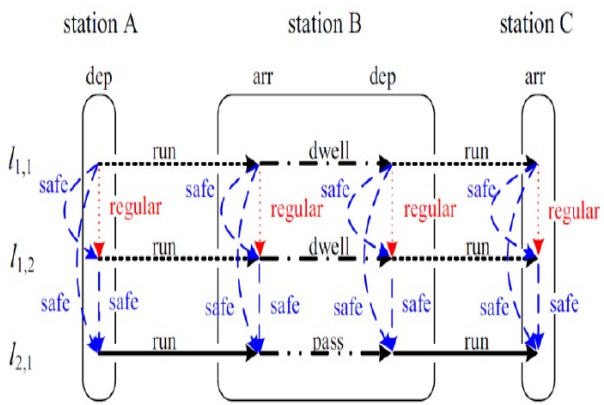


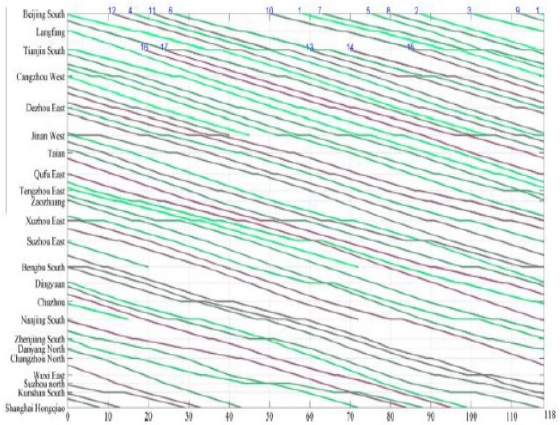
Commercial Solvers

B&B (Overall, Inserting)

Heuristics (GA,SA,TS,ACO)

2. PESP



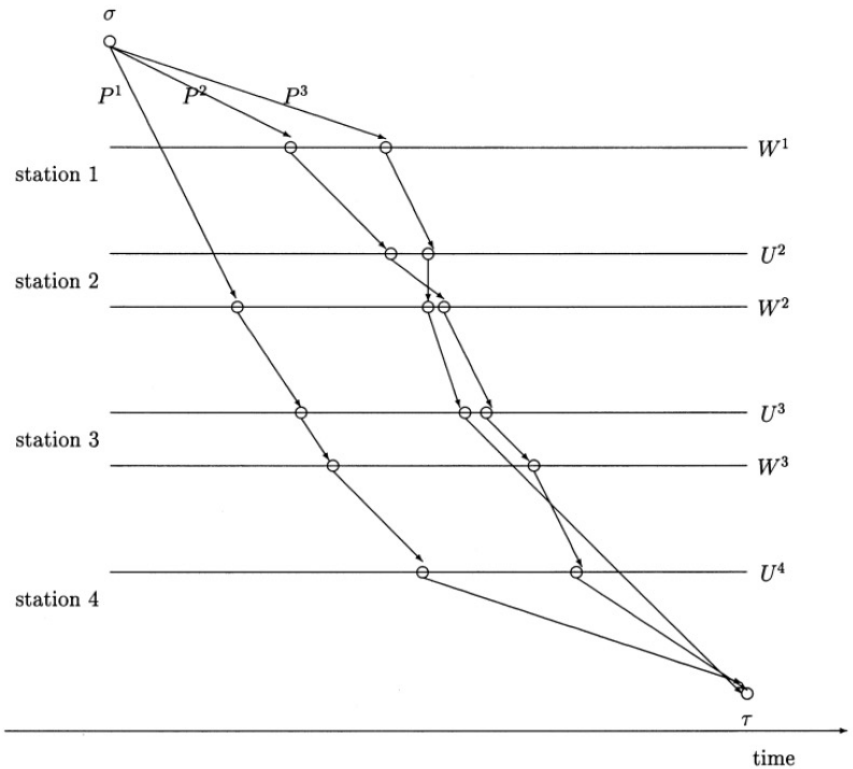


Commercial Solvers

Modulo Simplex Heuristic

铁路运行图建模

时空网络模型：



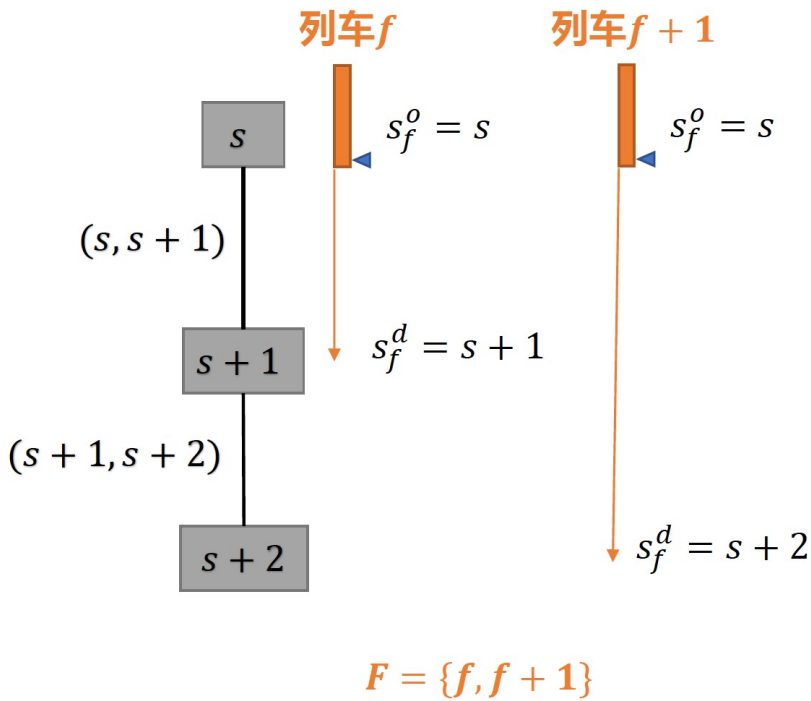
Lagrangian Relaxation

ADMM

Branch and Price

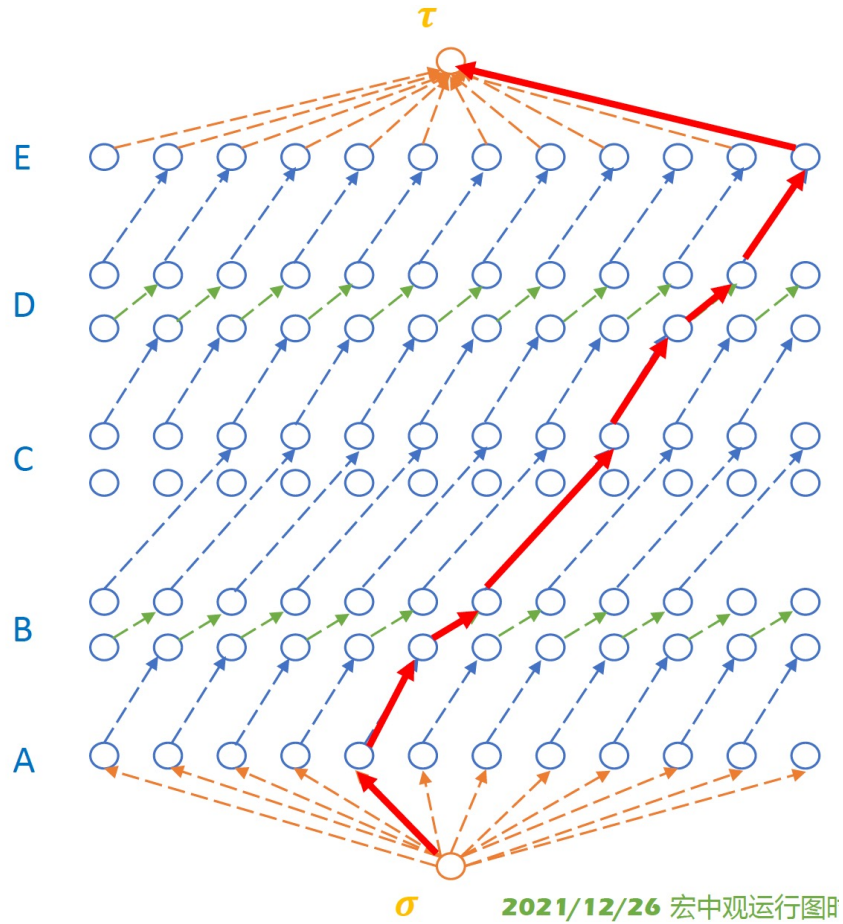
宏观运行图建模：时空网络模型

符号	解释
s	车站索引
SG	区间集合
(s_1, s_2)	区间索引
F	列车集合
f	列车索引
s_f^o	列车f的始发站索引
s_f^d	列车f的终到站索引
t	时间索引
$u, u_{s,t}$	通用时空节点
V_f	列车f可能经过的时空节点集合
σ	虚拟源节点
τ	虚拟汇节点



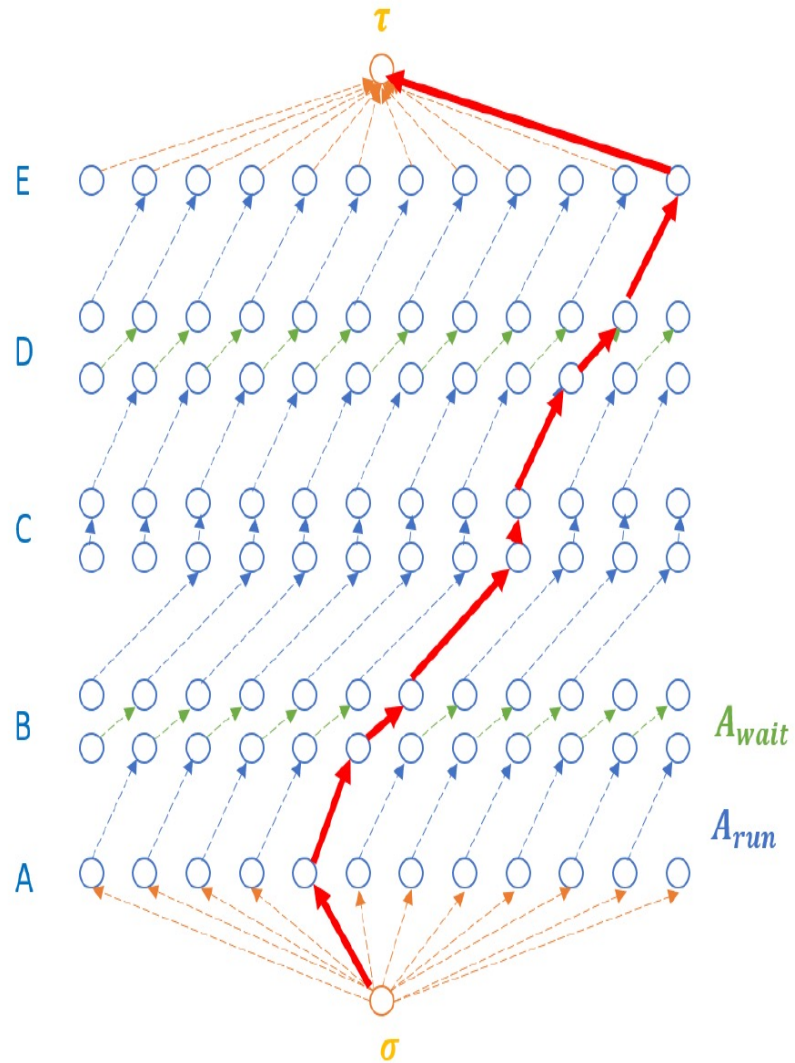
宏观运行图建模：时空网络模型

符号	解释
s	车站索引
SG	区间集合
(s_1, s_2)	区间索引
F	列车集合
f	列车索引
s_f^o	列车f的始发站索引
s_f^d	列车f的终到站索引
t	时间索引
$u, u_{s,t}$	通用时空节点
V_f	列车f可能经过的时空节点集合
σ	虚拟源节点
τ	虚拟汇节点



宏观运行图建模：时空网络模型

符号	解释
A	时空弧段集合
$A^+(u)$	流入时空节点u的弧段集合
$A^-(u)$	流出时空节点u的弧段集合
A_f	属于列车f的弧段集合
A_f^{wait}	属于列车f的停站弧段集合
A_f^{run}	属于列车f的区间运行弧段集合
A_f^{vt}	属于列车f的虚拟弧段集合 用于连接源节点与汇节点
a_f	属于列车f的弧段索引

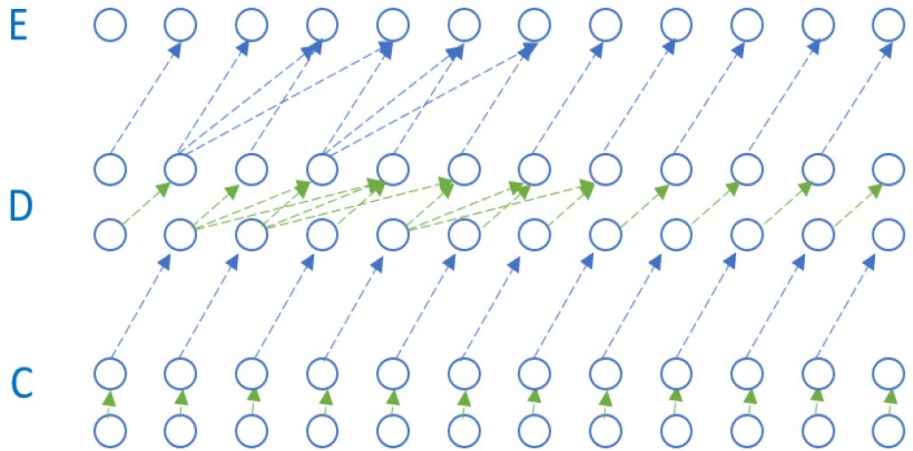


宏观运行图建模：目标函数

目标函数：列车总旅行时分最短

$$\min \sum_{f \in F} \sum_{a_f \in A_f} w_{a_f} x_{a_f}$$

$$w_{a_f} = \begin{cases} p_f(s_1, s_2) & \forall a_f \in A_f^{run} \cap (s_1, s_2) \\ \delta^{a_f} & \forall a_f \in A_f^{wait} \cap s \\ 0 & \forall a_f \in A_f^{vt} \end{cases}$$



$$\delta_{min}^{f,s} \leq \delta^{a_f} \leq \delta_{max}^{f,s}$$

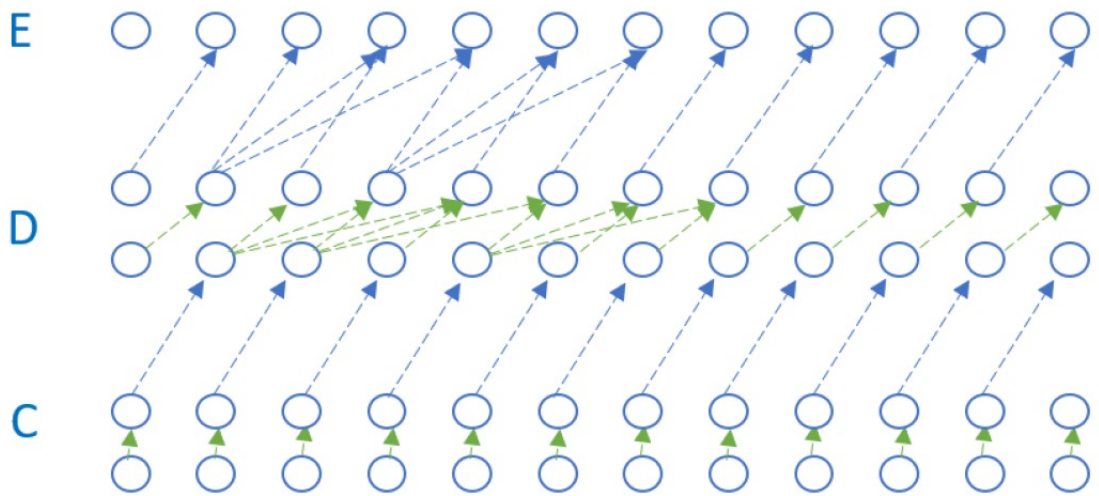
变量x_{af} 表示列车f是否在弧段a上运行:

$$x_{af} = \begin{cases} 1 & \text{if 列车} f \text{在弧段} a \text{上运行} \\ 0 & \text{o.w.} \end{cases}$$

宏观运行图建模：流平衡约束

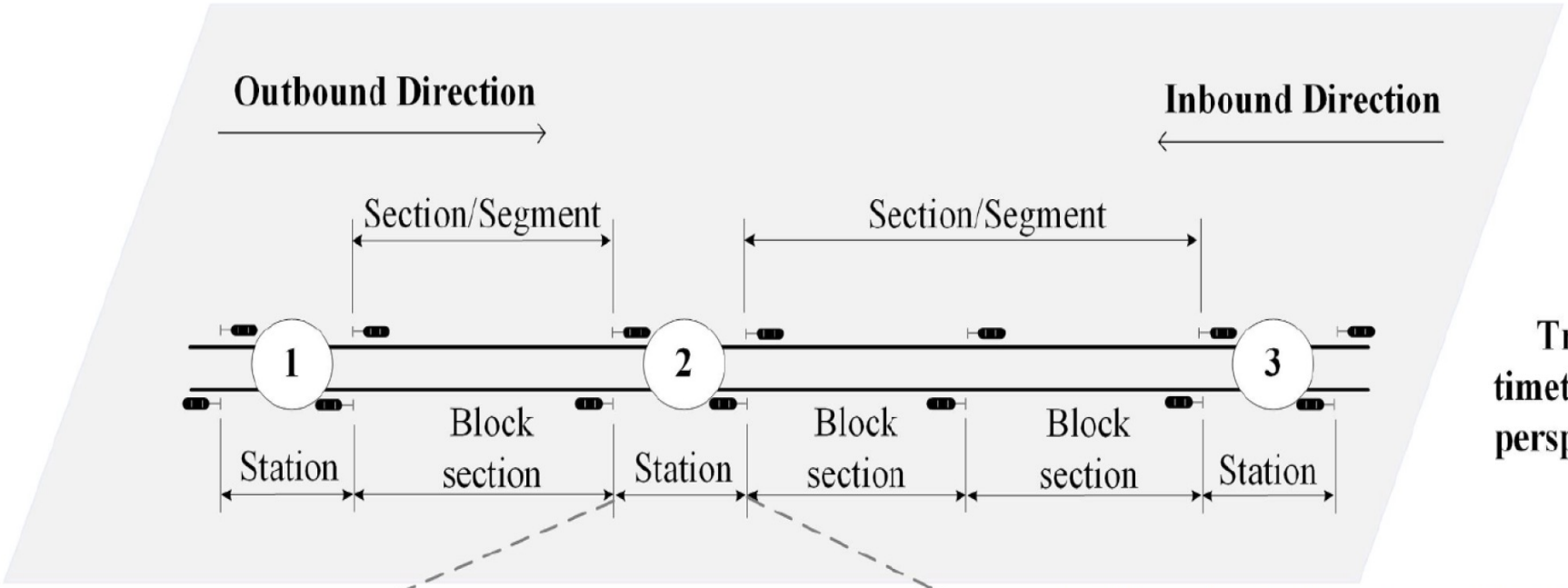
流平衡约束

$$\sum_{a_f \in A^+(u_f)} x_{a_f} - \sum_{a_f \in A^-(u_f)} x_{a_f} = \begin{cases} -1 & if\ u_f = \sigma \\ 1 & if\ u_f = \tau \\ 0 & otherwise \end{cases} \qquad \forall u_f \in V_f, \forall f \in F$$

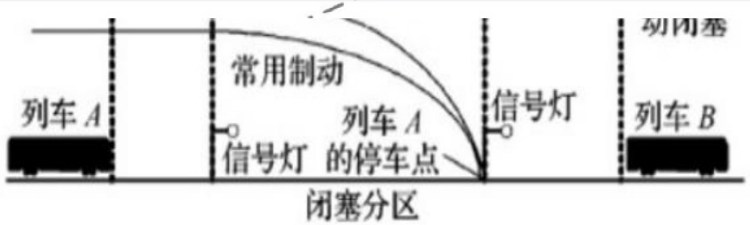


宏观运行图建模：轨道占用约束

轨道占用约束



Train timetabling perspective

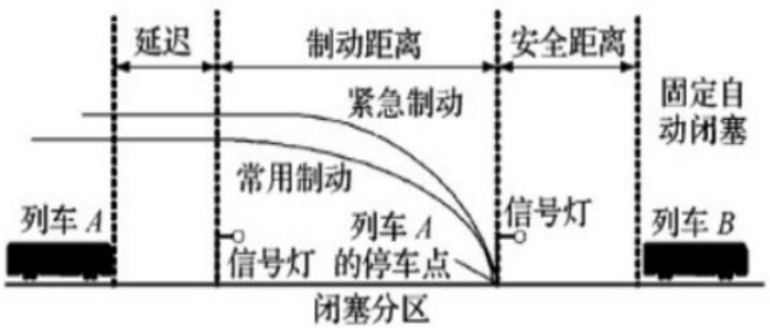


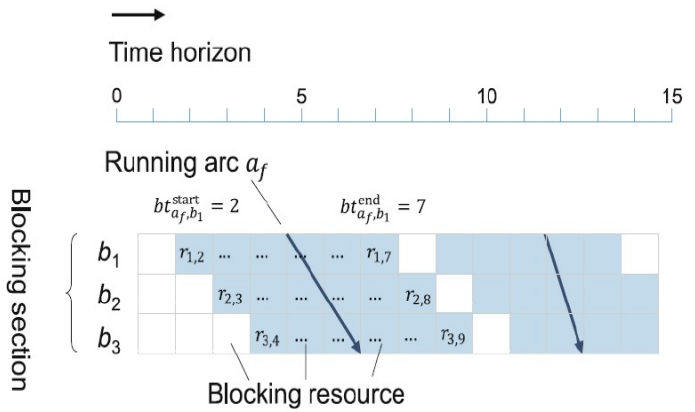
$$\sum_{f \in F} \sum_{a \in A_f^{run}: r \in R(a_f)} x_{a_f} \leq 1, \qquad \forall r \in R$$

宏观运行图建模：轨道占用约束

轨道占用约束

b	资源分区索引（实际闭塞分区）
$B_{(a)}$	弧段a占用的资源分区集合（实际闭塞分区）
$r, r_{b,t}$	时空资源，资源分区在各时段的资源体现
R	时空资源集合
$R_{(a)}$	弧段a占用的时空资源集合





$$R(a_f) = \{r_{1,2}, \dots r_{1,7}, r_{2,3}, \dots, r_{2,8}, r_{3,4}, \dots, r_{3,9}\}$$

Occupied blocking resource

$$R(a_f) = \left\{ r_{b,t} \middle| bt^{start}_{a_f,b} \le t \le bt^{end}_{a_f,b} \right\}, \qquad \forall b \in B(a_f)$$

$$\sum_{f \in F} \sum_{a_f \in A^{run}_f : r \in R(a_f)} x_{a_f} \le 1, \qquad \forall r \in R$$

宏观运行图建模：整体模型

总体模型

$$\min \sum_{f \in F} \sum_{a_f \in A_f} w_{a_f} x_{a_f}$$

$$w_{a_f} = \begin{cases} p_f(s_1, s_2) & \forall a_f \in A_f^{run} \cap (s_1, s_2) \\ \delta^{a_f} & \forall a_f \in A_f^{wait} \cap s \\ 0 & \forall a_f \in A_f^{vt} \end{cases}$$

$$\sum_{a_f \in A^+(u_f)} x_{a_f} - \sum_{a_f \in A^-(u_f)} x_{a_f} = \begin{cases} -1 & if\ u_f = \sigma \\ 1 & if\ u_f = \tau \\ 0 & otherwise \end{cases} \qquad \forall u_f \in V_f, \forall f \in F$$

$$\sum_{f \in F} \sum_{a \in A_f^{run}: r \in R(a_f)} x_{a_f} \leq 1, \qquad \forall r \in R$$

$$x_{a_f} \in \{0,1\}$$

References: Simultaneously re-optimizing timetables and platform schedules under planned track maintenance for a high-speed railway network, https://doi.org/10.1016/j.trc.2020.102823

课程项目

课程项目文件描述（持续更新）：
http://faculty.bicmr.pku.edu.cn/~wenzw/bigdata/trainsch.pdf

- （混合）整数规划建模: 从自然语言描述用大语言模型自动建模
- 调用Gurobi, Mosek或者SCIP等算法软件直接求解
- 基于ALM等连续优化算法+贪婪算法求解
- 实现AM系列算法求解
- 强化学习方法求解