优化理论与算法

第四章
网络流

郭加熠 | 助理教授

目录

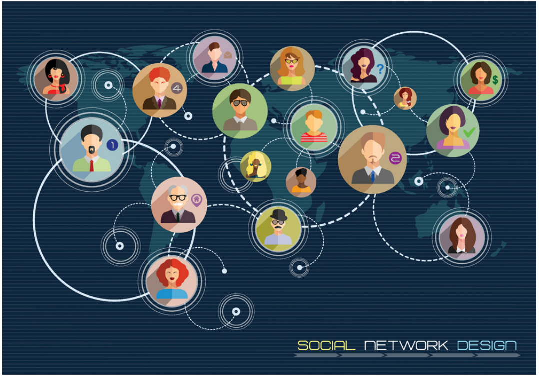
生活中的网络

南水北调的输水路线示意图

水管网络

油气网络

抽象意义上的网络



社交网络

神经网络

导航路径选择：交通网络中的最短路径

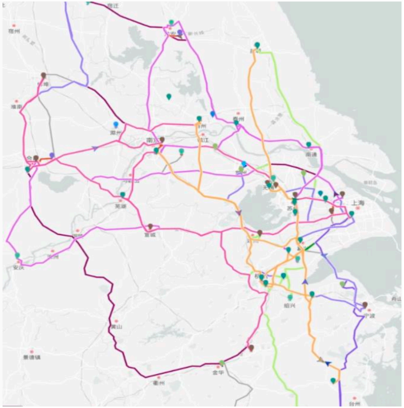
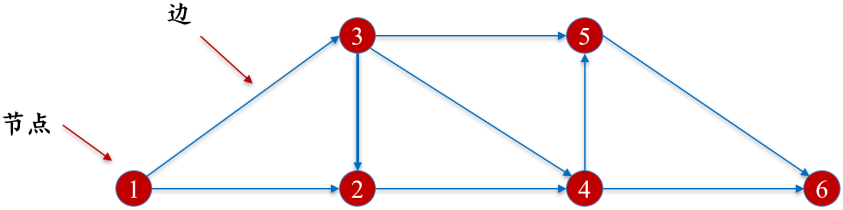


图 1: 算法给出推荐路线

怎样在复杂的交通网络快速找到

- 距离最短的路线
- 通行时间最少的路线
- 收费最低的线路

图的定义



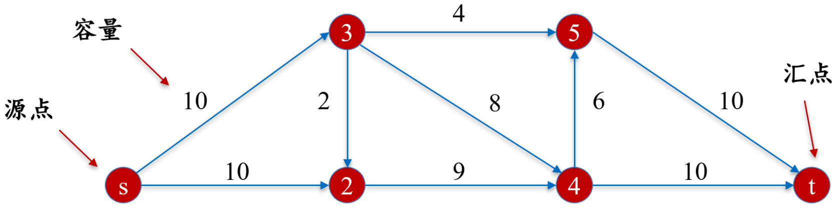
定义 4.1 图

图（有向图）由节点集 V 与边集 E 构成，表示为 $G = (V, E)$

1. $E \subseteq V \times V$ 边为点的**二元子集**
2. (i, j) 为**有向边**，即边 (i, j) 离开节点 i 进入节点 j

$$V = \{1, 2, 3, 4, 5, 6\}, \quad E = \{(1, 2), (1, 3), (3, 2), (2, 4), (3, 4), (3, 5), (4, 5), (4, 6), (5, 6)\}$$

流网络



定义 4.2 流网络

流网络是具有如下特点的有向图，记做 $G = (V, E, s, t, U)$

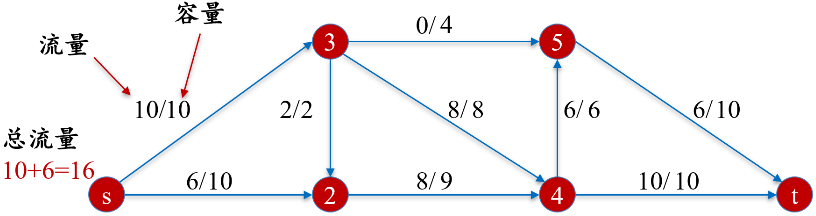
1. 每条边 (i, j) 均被赋予一个非负的 **容量 (capacity)** 参数 $u_{ij} \geq 0$
2. 存在唯一的源（source）节点 $s \in V$
3. 存在唯一的汇（sink）节点 $t \in V$

流网络的理解

以运输网络为例

- **源点**是物质运输的起点，**汇点**是运输的终点 (如导航运送乘客的出发点与终点)
- **边的容量**描述允许物质通过的限制 (如交通路线上车道的数量)
- **内部节点**描述物质在边之间的交互性质 (如路线上的十字路口)

$s-t$ 流的定义



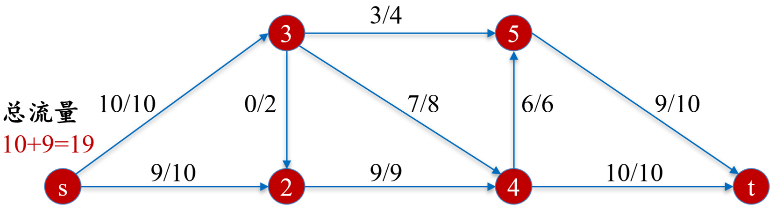
定义 4.3 $s-t$ 流

每条边 $(i,j) \in E$ 关联流量 $x_{ij} \geq 0$，并需要满足如下性质

- **容量约束**: $0 \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in E$
- **流平衡约束**：除源与汇节点外，任何节点 i 的进入流等于离开流

$$\sum_{(i,j)\in E} x_{ij} = \sum_{(k,i)\in E} x_{ki}, \quad \forall i \in V - \{s,t\}$$

最大流问题



定义 4.4 最大流问题

最大化总流量: $val(x) = \sum_{(s,j)\in E} x_{sj} - \sum_{(i,s)\in E} x_{is}$

$s - t$ 流满足如下约束条件:

- **容量约束:** $0 \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in E$
- **流平衡约束:** $\sum_{(i,j)\in E} x_{ij} = \sum_{(k,i)\in E} x_{ki}, \quad \forall i \in V - \{s,t\}$

算法复杂度

n 为节点数量，m 为边的数量，$|f_{max}|$ 为最大流量

算法	Fold-Folkerson	Edmonds-Karp	Dinic	BEST
复杂度	$O(m\|f_{max}\|)$	$O(nm^2)$	$O(n^2m)$	$O(mn)$

Orlin, James B. "**Max flows in O(nm) time, or better.**" Proceedings of the forty-fifth annual ACM symposium on Theory of computing. 2013.

ABSTRACT

In this paper, we present improved polynomial time algorithms for the max flow problem defined on sparse networks with n nodes and m arcs. We show how to solve the max flow problem in $O(nm + m^{31/16} \log^2 n)$ time. In the case that $m = O(n^{1.06})$, this improves upon the best previous algorithm due to King, Rao, and Tarjan, who solved the max flow problem in $O(nm \log_{m/(n \log n)} n)$ time. This establishes that the max flow problem is solvable in $O(nm)$ time for all values of n and m. In the case that $m = O(n)$, we improve the running time to $O(n^2/\log n)$.

目录

最小割问题

定义 4.5 $s-t$ 割（cut）

流网络 $G=(V,E,s,t,U)$ 的一个**割**（V_s,V_t）是将点集 V 划分为两个子集合 V_s 和 V_t，满足

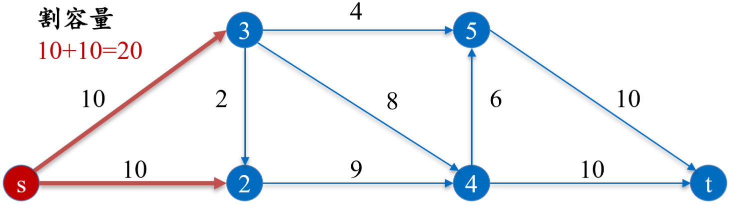
- $V_s\bigcap V_t=\Phi, V_s\bigcup V_t=V$
- $s\in V_s, t\in V_t$

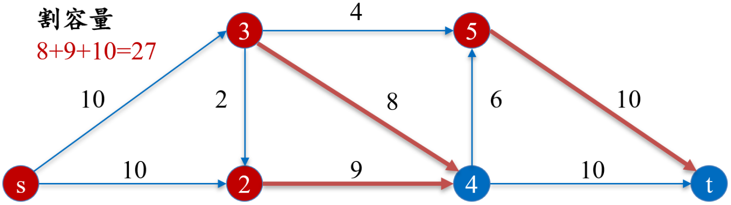
定义 4.6 割容量与最小割（min cut）问题

割（V_s,V_t）所对应的**割容量**：V_s 到 V_t 的容量，即 $\sum\limits_{\{(i,j)\in E:i\in V_s,j\in V_t\}} u_{ij}$

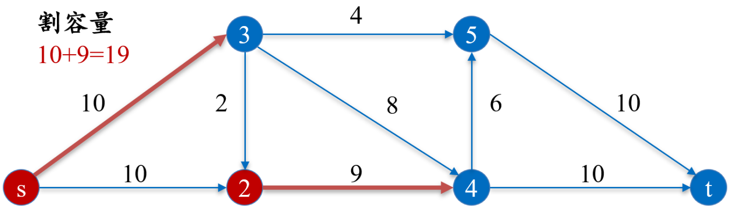
最小割问题：找到所有割中有最小割容量的割

割容量示例

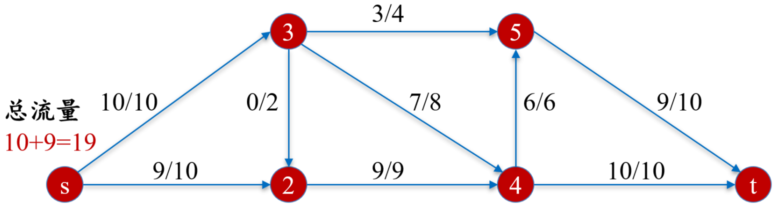

割容量
10+10=20
3
4
5
10
2
8
6
10
s
10
2
9
4
10
t


割容量
8+9+10=27
3
4
5
10
2
8
6
10
s
10
2
9
4
10
t

最小割示例



最小割与最大流一致!



建模最小割问题

$$z_i = \begin{cases} 1, & \text{if } i \in V_s. \\ 0, & \text{otherwise.} \end{cases} \qquad y_{ij} = \begin{cases} 1, & \text{if } i \in V_s, j \in V_t. \\ 0, & \text{otherwise.} \end{cases}$$

最小割问题建模如下：

$$\text{minimize} \sum_{(i,j)\in E} u_{ij} y_{ij}$$

$$\text{subject to } y_{ij} = \max\{z_i - z_j, 0\}, \quad \forall (i,j) \in E$$

$$z_s = 1, z_t = 0,$$

$$z_i \in \{0,1\}, \quad \forall i \in V$$

回顾最大流问题

为了简化，假设 s 点只有出流

$$
\begin{array}{lll}
\text{maximize} & \sum_{(s,j)\in E} x_{sj} & \\
\text{subject to} & x_{ij} \le u_{ij}, & \forall (i,j) \in E \\
& \sum_{(k,i)\in E} x_{ki} - \sum_{(i,j)\in E} x_{ij} = 0, & \forall i \in V - \{s,t\} \\
& x_{ij} \ge 0 &
\end{array}
$$

一个自然的疑问：最大流问题的**对偶**是什么含义？

最大流线性规划的对偶

分别引入 y_{ij} 与 z_i 作为容量限制约束与流平衡约束的对偶变量，

$$
\begin{aligned}
\underset{y_{ij},z_i}{\text{minimize}} \quad & \textstyle\sum_{(i,j)\in E} u_{ij}y_{ij} \\
\text{subject to} \quad & y_{sj} + z_j \geq 1 \\
& y_{ij} + z_j - z_i \geq 0 \\
& y_{it} - z_i \geq 0 \\
& y_{ij} \geq 0
\end{aligned}
\qquad
\begin{aligned}
\underset{y_{ij},z_i}{\text{minimize}} \quad & \textstyle\sum_{(i,j)\in E} u_{ij}y_{ij} \\
\text{subject to} \quad & y_{sj} \geq 1 - z_j & \forall(s,j)\in E \\
& y_{ij} \geq z_i - z_j, & \forall(i,j)\in E - \{(s,j),(t,i)\} \\
& y_{it} \geq z_i - 0, & \forall(i,t)\in E \\
& y_{ij} \geq 0, & \forall(i,j)\in E
\end{aligned}
$$

我们最终得到等价模型

$$
\begin{aligned}
\underset{y_{ij},z_i}{\text{minimize}} \quad & \textstyle\sum_{(i,j)\in E} u_{ij}y_{ij} \\
\text{subject to} \quad & y_{ij} \geq z_i - z_j, \quad y_{ij} \geq 0, \quad \forall(i,j)\in E \\
& z_s = 1, z_t = 0
\end{aligned}
$$

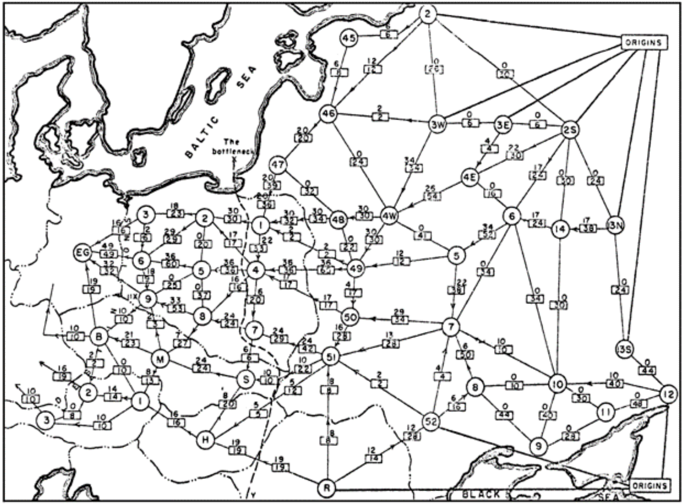
最大流与最小割

上述对偶问题可以进一步写为

$$\begin{aligned} \underset{y_{ij}, z_i}{\text{minimize}} \quad & \textstyle\sum_{(i,j)\in E} u_{ij} y_{ij} \\ \text{subject to} \quad & y_{ij} = \max\{z_i - z_j, 0\}, \quad \forall (i,j) \in E \\ & z_s = 1, z_t = 0 \end{aligned}$$

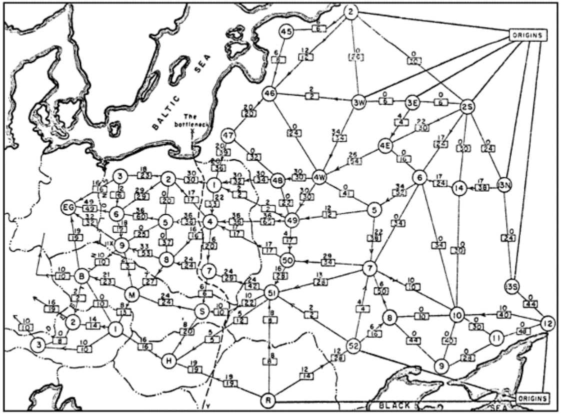
- 基于后面马上介绍的**全单模性质**，可知在最优解处有 $z_i \in \{0,1\}$
  进而，可构造对应流网络上的**最小割**： $V_s = \{i : z_i = 1\}, V_t = \{i : z_i = 0\}$
- 根据强对偶性质，最大流-最小割问题的最优值相等，即 **最大流-最小割** 定理

猜猜这张图片是哪里的铁路图



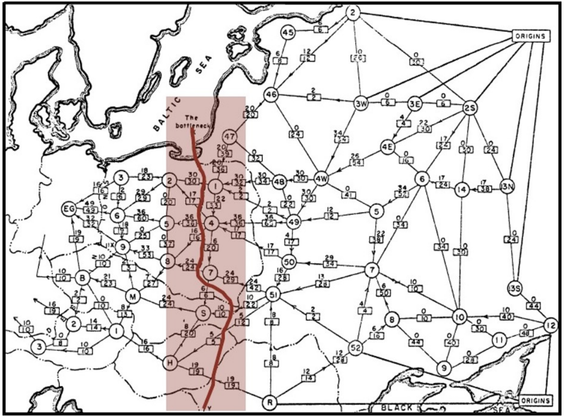
最大流的应用

苏联的目标： 寻找向东欧前线输送最多补给的方案



最小割的应用

盟军的目标： 利用最小代价切断铁路供给网



1999 年美国五角大楼解秘苏联与东欧国家的铁路图

目录

回顾：运输问题的数学模型

基本的供应链运输模型场景中，

给定二分图 $G = (V_s \bigcup V_d, E, C)$

- 供应商 i 的供应量 $s_i \in V_s$
- 需求方 j 的需求量 $d_j \in V_d$
- 供应商 i 到需求方 j 的**单位运输成本** c_{ij}
- 假设供需平衡 $\sum_{i \in V_s} s_i = \sum_{d_j \in V_d} d_j$
- (如果 $s_i = d_j = 1$，则被称为**匹配问题**)

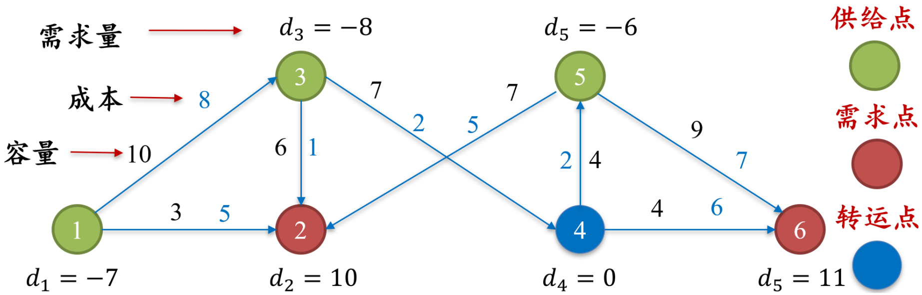
决策：供应商 i 到需求方 j 的运输量 x_{ij}

如何在满足供求关系的条件下，

最小化运输成本?

$$
\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{(i,j)\in E} c_{ij} x_{ij} \\
\text{subject to} \quad & \textstyle\sum_{(i,j)\in E} x_{ij} = s_i, \quad \forall i \in V_s, \\
& \textstyle\sum_{(i,j)\in E} x_{ij} = d_j, \quad \forall j \in V_d \\
& x_{ij} \geq 0, \quad \forall (i,j) \in E
\end{aligned}
$$

最小成本流问题要素



最小成本流问题**要素**：给定图 $G = (V, E, U, C, D)$

- 每条边 $(i, j) \in E$ 关联**容量**：u_{ij}
- 每条边 $(i, j) \in E$ 关联**单位成本**：c_{ij}
- 每个点 $i \in V$ 关联**流需求量**：d_i
- (流需求量可为负，代表流供给量)

最小成本流问题

最小化成本流问题：在满足流约束的条件下，最小化总成本

设每条边 $(i,j) \in E$ 上有流量 x_{ij}

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in E} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{(k,i)\in E} x_{ki} - \sum_{(i,j)\in E} x_{ij} = d_i, && \forall i \in V \\
& 0 \le x_{ij} \le u_{ij}, && \forall (i,j) \in E
\end{aligned}
$$

此线性规划有解的**前提**：供需平衡，即 $\sum_{i\in V} d_i = 0$

化归为最小成本流问题

大部分网络流问题都可以化为最小成本流问题，并设计了特定算法

- 运输问题 $\to$ 表上作业法
- 分配问题 $\to$ 匈牙利算法
- 最小成本最大流问题 $\to$ 最大流算法 $+$ 二分查找
- 最短路径 $\to$ Dijkstra 算法
- …

最小成本最大流问题

流网络中，每条边 $(i,j) \in E$ 关联**单位成本**参数 c_{ij}，记做 $G = (V, E, s, t, U, C)$

最小成本最大流问题：在所有实现最大流的 $s-t$ 流中，选取总成本最小的

先利用最大流问题求出最大流量 f^*，并构线性规划问题如下

$$
\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{(i,j)\in E} c_{ij} x_{ij} \\
\text{subject to} \quad & \textstyle\sum_{(s,j)\in E} x_{sj} - \sum_{(i,s)\in E} x_{is} = f^* \\
& \textstyle\sum_{(i,j)\in E} x_{ij} - \sum_{(k,i)\in E} x_{ki} = 0, \quad \forall i \in V - \{s,t\} \\
& 0 \le x_{ij} \le u_{ij}, \quad \forall (i,j) \in E
\end{aligned}
$$

最短路问题

给定流网络 $G=(V,E,s,t,U,C)$，其中 $u_{ij}=1,\forall(i,j)\in E$

求有一条从起点 s 到终点 t 的最短的路径（此时，c_{ij} 代表边 (i,j) 的长度）

$$
\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} \\
\text{subject to} \quad & \sum_{(s,j)\in E} x_{sj} = \sum_{(i,t)\in E} x_{it} = 1 \\
& \sum_{(i,j)\in E} x_{ij} = \sum_{(k,i)\in E} x_{ki}, \qquad \forall i \in V-\{s,t\} \\
& 0 \le x_{ij} \le 1, \qquad \forall (i,j)\in E
\end{aligned}
$$

根据接下来的**全单模性质**可知，在最优解处 $x_{ij}\in\{0,1\}$，并构建出最短路。

求解最小成本流算法

算法设计

- Cycle canceling algorithms
- Push/Relabel Algorithms

线性规划

- 网络单纯形法
- 内点法

目前最好方法：$O(m^{1+o(1)} \log I)$，其中 m 为边的个数，

I 为重新放缩后使所有边容量均为整数的最大值（无理数时是无穷）

Chen, Li, et al. **Maximum flow and minimum-cost flow in almost-linear time.** 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2022.

目录

全单模矩阵 Total Unimodularity Matrix (TUM)

定义 4.7 全单模矩阵

一个矩阵的任何子方阵是单模的，即该子矩阵的行列式等于-1,1 或 0

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 \end{bmatrix}$$

这两个矩阵是不是全单模矩阵呢?

全单模矩阵与整数解

定理 4.1 全单模矩阵

如果 A 是一个全单模矩阵，b 为整数向量，那么 $\{Ax = b\ x \geq 0\}$ 的所有顶点解均为整数解。

证明：每个基本可行解可用 $A_B x_B = b$ 计算，根据克莱默法则有 x_B 均为整数解

定理 4.2 克莱默法则 (Cramer's Rule)

对可逆方阵 $B \in \mathbb{R}^{n\times n}$，那么 $Bx = b$ 的解，可以表示为 $x_i = \frac{det(B_i)}{det(B)}$

其中，B_i 是将 B 中第 i 列，替换成 b

全单模矩阵的充分条件

定理 4.3 全单模矩阵的充分条件

如果一个 $\{0, 1, -1\}$ 矩阵，每列只有至多一个 1 和至多一个-1，那么该矩阵是全单模矩阵

证明：对子方阵 B 进行归纳

- Base case: 1×1 matrix
- Case 1: B 中某列全是 0
- Case 2: B 中某列只有一个 1 或-1
- Case 3: B 中所有列全有一个 1 和一个-1

点边关联矩阵矩阵

无向图中的点边关联矩阵为 $\{0,1\}$ 矩阵

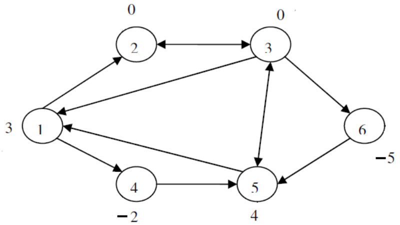
定义 4.8 有向图中点边关联矩阵（node-edge incidence matrix）

关联矩阵 $A \in \mathbb{R}^{n\times m}$, $A_{ij} = \begin{cases} -1 & \text{if edge } j \text{ leaves vertex } i \\ 1 & \text{if edge } j \text{ enters vertex } i \\ 0 & \text{otherwise} \end{cases}$

点边关联矩阵是全单模矩阵

点边关联矩阵举例

	x_{12}	x_{14}	x_{23}	x_{31}	x_{32}	x_{35}	x_{36}	x_{45}	x_{51}	x_{53}	x_{65}	d
1	−1	−1	0	1	0	0	0	0	1	0	0	=3
2	1	0	−1	0	1	0	0	0	0	0	0	=0
3	0	0	1	−1	−1	−1	−1	0	0	1	0	=0
4	0	1	0	0	0	0	0	−1	0	0	0	=−2
5	0	0	0	0	0	1	0	1	−1	−1	1	=4
6	0	0	0	0	0	0	1	0	0	0	−1	=−5



设点边关联矩阵为 A，那么最小成本流的约束条件

$$\sum_{(k,i)\in E} x_{ki} - \sum_{(i,j)\in E} x_{ij} = d_i, \quad \forall i \in V$$

可以改写为 $A\boldsymbol{x} = \boldsymbol{d}$

全单模矩阵的性质

A 矩阵是全单模矩阵

定理 4.4 全单模矩阵的性质

- 交换部分行或列保持全单模矩阵
- 某行或列乘 -1 保持全单模矩阵
- A^T 是全单模矩阵
- $[\,A \mid I\,]$ 和 $\left[\frac{A}{I}\right]$ 是全单模矩阵

回顾最小成本流约束：

流约束：$\sum_{(k,i)\in E} x_{ki} - \sum_{(i,j)\in E} x_{ij} = d_i, \ \forall i \in V$

容量约束：$0 \leq x_{ij} \leq u_{ij}, \qquad \forall (i,j) \in E$

最小成本流的整数定理

根据定理 4.1 和定理 4.4 的最后一条性质可得

定理 4.5 最小成本流的整数定理

若给定最小成本流问题 $G = (V, E, U, C, D)$，其中所有 $u_{ij} \in U$ 以及 $d_i \in D$ 均为整数，那么其对应线性规划问题的任意基本可行解均为整数解

注：整数定理与参数 c_{ij} 是否是整数无关

从而可知，最小成本流的一系列应用均满足整数定理：运输问题，匹配问题，最大流问题、最小割问题、最小成本最大流问题、最短路问题

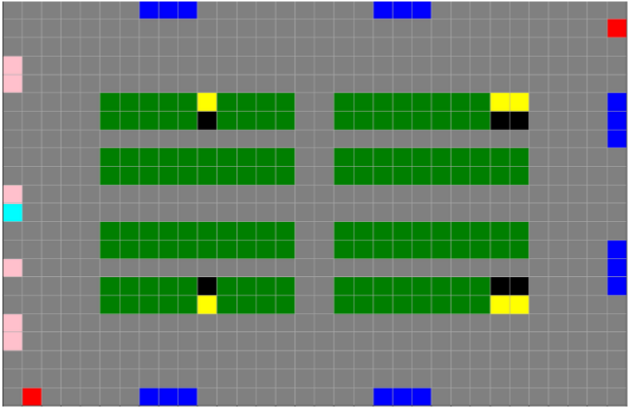
目录

背景介绍

无人仓管理：进货、存货、寻货、出货等

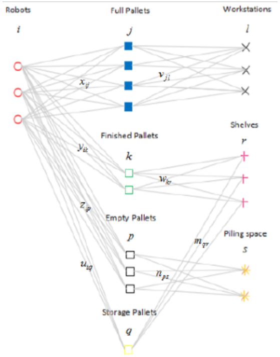
无人仓地图

- 路径节点（灰色）
- 储位节点（绿色）
- 充电桩节点（黄色）
- 柱子节点（黑色）
- 拣选工位节点（蓝色）
- 补货位节点（粉色）



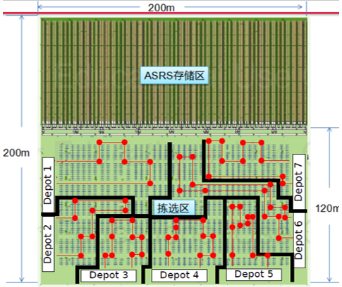
无人仓任务分解

智能拣货路径



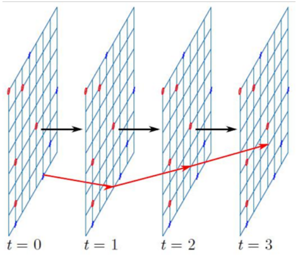
统筹安排任务

任务均衡区域划分



均衡工作台负载

智能避免拥堵碰撞



避免小车碰撞

无人仓拣货调度问题

无人车（AGV）需要完成如下的一些任务：

- 出库：搬运运载有商品的货架到空闲工位
- 回库：搬运货架从工位回到仓库储位
- 回收：搬运空货架从工位到货架回收处
- ...

无人仓的核心是统筹安排 AGV 的各种上述任务，安排 AGV 的行径路线满足全局最优、实时响应等等要求。

三分图建模思路

AGV 的路径规划，可以简化为一个**三分图匹配问题**，即

- 红色：空闲小车
- 绿色：储位或工位的货架
- 蓝色：终点（工位、储位）

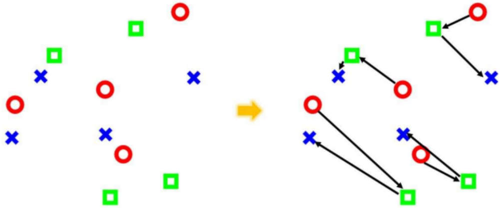
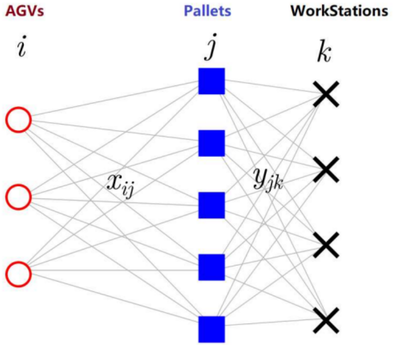


图 2: 无人车取货与送货示意图

三分图展示

三分图建模具有良好的扩展性，支持多新增种类的任务，只需要增加对应的节点即可。



利用最小成本最大流思路进行建模

无人仓 AGV 调度问题：符号定义

定义如下符号:

- 决策变量 $x_{ij} \in \{0, 1\}$: AGV i 是否运输货架 j
- 决策变量 $y_{jk} \in \{0, 1\}$: 货架 j 是否被运送到工位 k
- 参数 c_{ij}^1: AGV i 从当前位置移动到货架 j 的距离
- 参数 c_{jk}^2: 从货架 j 到工位 k 的距离

最大流模型

首先建模最大流问题

目标函数：最大化可以接到任务的 AGV 的数量

$$\text{maximize} \quad \text{MaxFlow} = \sum_{i,j} x_{ij}$$

约束条件：

- **AGV**: 每个 AGV 最多被指派一个任务，这里指获取一个货架

$$\sum_{j} x_{ij} \leq 1 \quad \forall i$$

最大流模型（继续）

约束条件:

- **货架**: 每个货架最多分配给一个 AGV

$$\sum_i x_{ij} \leq 1 \quad \forall j$$

- **货架**: 每个货架只能送到一个工位

$$\sum_k y_{jk} \leq 1 \quad \forall j$$

最大流模型（继续）

- **工位**: 每个工位最多接受三个货架

$$\sum_j y_{jk} \leq 3 \quad \forall k$$

- **货架**: 被分配了 AGV 的货架必须分配到一个工位

$$\sum_i x_{ij} = \sum_k y_{jk} \quad \forall j$$

最大流完整模型

最大流问题的完整模型如下

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i,j} x_{ij} \\
\text{subject to} \quad & \sum_{j} x_{ij} \le 1 \quad \forall i, \quad \sum_{i} x_{ij} \le 1 \quad \forall j \\
& \sum_{k} y_{jk} \le 1 \quad \forall j, \quad \sum_{j} y_{jk} \le 3 \quad \forall k \\
& \sum_{i} x_{ij} = \sum_{k} y_{jk} \quad \forall j, \\
& x_{ij} \in \{0,1\}, \quad y_{ij} \in \{0,1\}
\end{aligned}
$$

最小成本最大流模型

给定最大流 f^*，最小化 AGV 的总移动距离和

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j} c_{ij}^1 x_{ij} + \sum_{j,k} c_{jk}^2 y_{jk} \\
\text{subject to} \quad & \sum_{j} x_{ij} \le 1 \quad \forall i, \quad \sum_{i} x_{ij} \le 1 \quad \forall j \\
& \sum_{k} y_{jk} \le 1 \quad \forall j, \quad \sum_{j} y_{jk} \le 3 \quad \forall k \\
& \sum_{i} x_{ij} = \sum_{k} y_{jk} \quad \forall j, \quad \sum_{i,j} x_{ij} = f^* \\
& x_{ij} \in \{0,1\}, \quad y_{ij} \in \{0,1\}
\end{aligned}
$$

样例求解结果

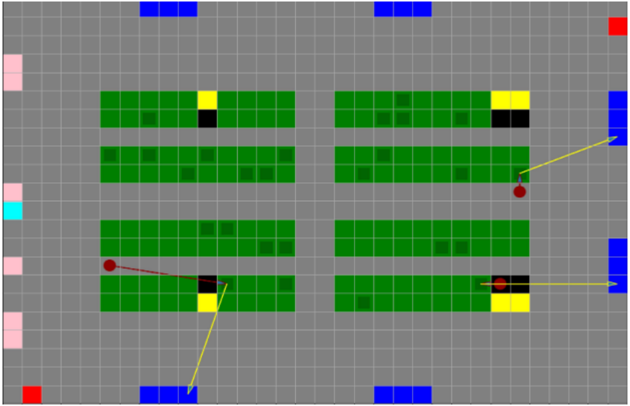


图 3: 样例求解结果

拓展思考

- **网络建模**：AGV 转弯一停二转三通过，网络拓展 (东西南北 4 个点替一个点)
- **避撞系统**：从流网络的不相交路径到三维时空网络的不相交路径
- **提速方式**：任务分区，利用两个二分图匹配问题替代三分图匹配

感谢聆听！

Thank You!