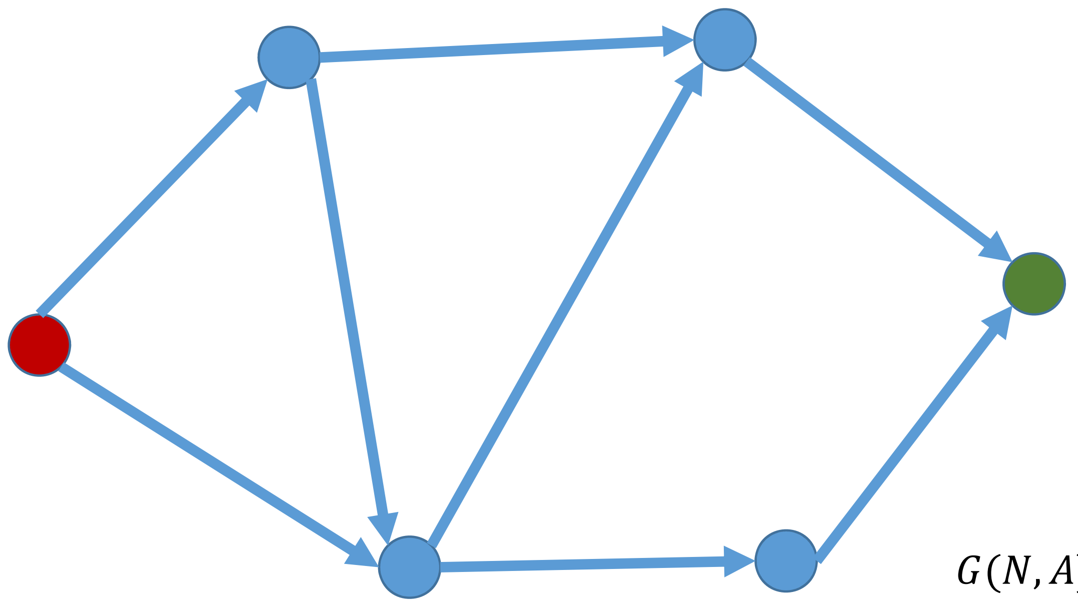
大规模整数规划分解算法介绍

以通信网络节点布局优化问题为例

廖正文

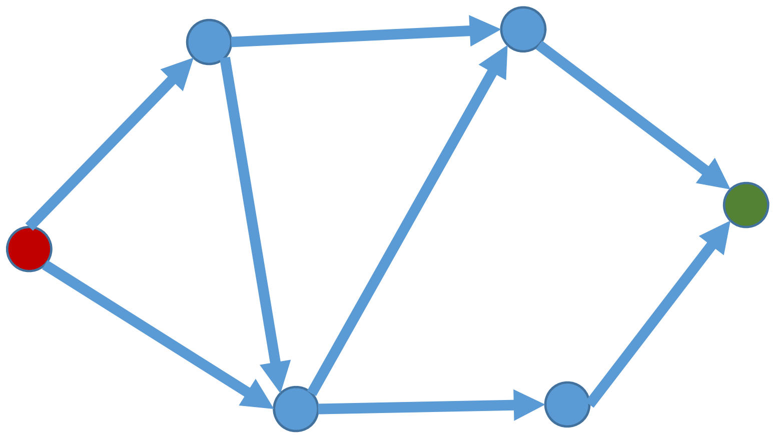
2018. 3. 22

从最短路到网络流

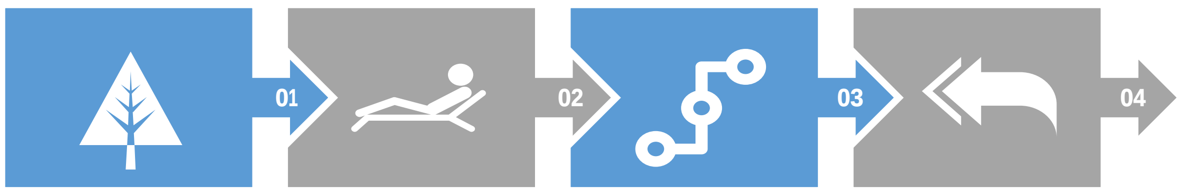


$G(N, A)$

从最短路到网络流

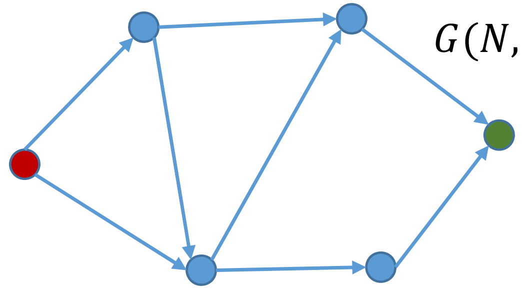


- 最短路径算法
  - Dijkstra 算法
  - B – F 算法
  - SPFA 算法
  - Forward DP 算法
  - Floyd 算法
  - …



构建搜索数据	**“松弛”操作**	**遍历节点 / 边**	**回溯**
构建搜索队列/树/栈	更新节点的路长标号	完成所有的可行域搜索	获取最短路径结果

从最短路到网络流



最短路问题：0-1规划模型

$$Minimize \sum_{a\in A} c_a x_a$$

$$s.t.$$

$$\sum_{a\in O^-} x_a = \sum_{a\in D^+} x_a = 1$$

$$\sum_{a\in n^-} x_a = \sum_{a\in n^+} x_a \quad \forall n \in N - \{O, D\}$$

$$x_a = \{0,1\} \ \forall a \in A$$

x_a - 0-1变量，最短路径是否包含边a

c_a - 边a上的出行成本

最小费用流问题：线性规划模型

$$Minimize \sum_{a\in A} c_a x_a$$

$$s.t.$$

$$\sum_{a\in O^-} x_a = \sum_{a\in D^+} x_a = V$$

流量约束
V为网络上的总流量

$$\sum_{a\in n^-} x_a = \sum_{a\in n^+} x_a \quad \forall n \in N - \{O, D\}$$

$$x_a \le C_a \ \ \forall a \in A$$

能力约束
C_a为边的能力

$$x_a \in \mathbb{R}^+ \ \ \forall a \in A$$

从最短路到网络流

一般多商品流模型

$$Minimize \sum_{(O,D)\in\mathcal{D}} \sum_{a\in A} c_a x_a^{OD}$$

$s.t.$

多起终点，流量分层

$$\sum_{a\in O^-} x_a^{OD} = \sum_{a\in D^+} x_a^{OD} = V_{OD} \quad \forall (O,D)\in\mathcal{D}$$

$$\sum_{a\in n^-} x_a^{OD} = \sum_{a\in n^+} x_a^{OD} \quad \forall (O,D)\in\mathcal{D}, n\in N-\{O,D\}$$

$$\sum_{(O,D)\in\mathcal{D}} x_a^{OD} \le C_a \quad \forall a\in A$$

能力，多层流量叠加

$$x_a^{OD} \in \mathbb{R}^+ \quad \forall a\in A, (O,D)\in\mathcal{D}$$

(O,D) – 一对流的起讫点

$$Min \sum_{a\in A} c_a x_a^1 \quad + \quad Min \sum_{a\in A} c_a x_a^2 \quad + \quad Min \sum_{a\in A} c_a x_a^{OD}$$

$s.t.$

$$\sum_{a\in O^-} x_a^1 = \sum_{a\in D^+} x_a^1 = V_1$$

$$\sum_{a\in n^-} x_a^1 = \sum_{a\in n^+} x_a^1 \quad \forall n\in N-\{O,D\}$$

$$\sum_{a\in O^-} x_a^2 = \sum_{a\in D^+} x_a^2 = V_2$$

$$\sum_{a\in n^-} x_a^2 = \sum_{a\in n^+} x_a^2 \quad \forall n\in N-\{O,D\}$$

$$\sum_{a\in O^-} x_a^3 = \sum_{a\in D^+} x_a^3 = V_3$$

$$\sum_{a\in n^-} x_a^3 = \sum_{a\in n^+} x_a^3 \quad \forall n\in N-\{O,D\}$$

$$x_a^1 \quad + \quad x_a^2 \quad + \quad x_a^3 \quad \le C_a \quad \forall a\in A$$

从最短路到网络流

一般多商品流模型

$$Minimize \quad \sum_{(O,D)\in\mathcal{D}} \sum_{a\in A} c_a x_a^{OD}$$

$s.t.$

多起终点，流量分层

$$\sum_{a\in O^-} x_a^{OD} = \sum_{a\in D^+} x_a^{OD} = V_{OD} \quad \forall (O,D)\in\mathcal{D}$$

$$\sum_{a\in n^-} x_a^{OD} = \sum_{a\in n^+} x_a^{OD} \quad \forall (O,D)\in\mathcal{D}, n\in N-\{O,D\}$$

$$\sum_{(O,D)\in\mathcal{D}} x_a^{OD} \le C_a \quad \forall a\in A$$

能力，多层流量叠加

$$x_a^{OD} \in \mathbb{R}^+ \quad \forall a\in A, (O,D)\in\mathcal{D}$$

(O,D) – 一对流的起讫点

同一支OD流路径不可拆分的多商品流模型

$$Minimize \quad \sum_{(O,D)\in\mathcal{D}} \sum_{a\in A} V_{OD} c_a x_a$$

$s.t.$

$$\sum_{a\in O^-} x_a^{OD} = \sum_{a\in D^+} x_a^{OD} = 1 \quad \forall (O,D)\in\mathcal{D}$$

$$\sum_{a\in n^-} x_a^{OD} = \sum_{a\in n^+} x_a^{OD} \quad \forall (O,D)\in\mathcal{D}, n\in N-\{O,D\}$$

$$\sum_{(O,D)\in\mathcal{D}} x_a^{OD} \le C_a \quad \forall a\in A$$

$$x_a^{OD} \in \{0,1\} \quad \forall a\in A, (O,D)\in\mathcal{D}$$

从最短路到网络流

单商品流模型

$$Minimize \sum_{a\in A} c_a x_a$$

$$s.t.$$

$$\sum_{n\in N} s_n = \sum_{n\in N} g_n = V$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

$$x_a \in \mathbb{R}^+ \quad \forall a \in A$$

$$s_n, g_n \in \mathbb{R}^+ \quad \forall n \in N$$

s_n、g_n – 节点n的流量需求和供给

- 单商品流 VS 多商品流
  - 多商品流：仅共享能力

    如：车流、快递包裹、通讯网络报文
  - 单商品流：共享能力的同时共享流量

    如：水、电、天然气

从最短路到网络流

网络流关键内容

流平衡约束

网络流问题的基础

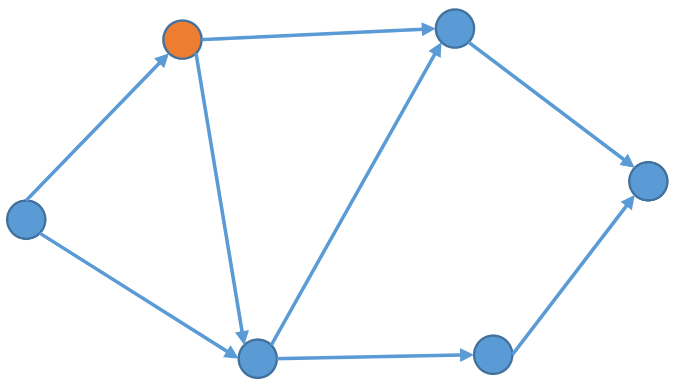
费用约束 / 目标

目标函数往往与费用紧密相关
部分约束条件存在费用限制

能力约束

网络流问题的核心难点

CDN网络



- CDN网络流量问题
  - 影视服务器部署位置已知（N_O）
  - 通讯网络的通道已知（$a \in A$）
  - 通讯网络的通道带宽（C_a）已知
  - 终端节点的信息量需求（s_n）已知
  - 已知求流在网络上的分布（x_a）

$$Minimize \sum_{a \in A} c_a x_a$$

$$s.t.$$

$$g_n = 0 \quad \forall n \notin N_O$$

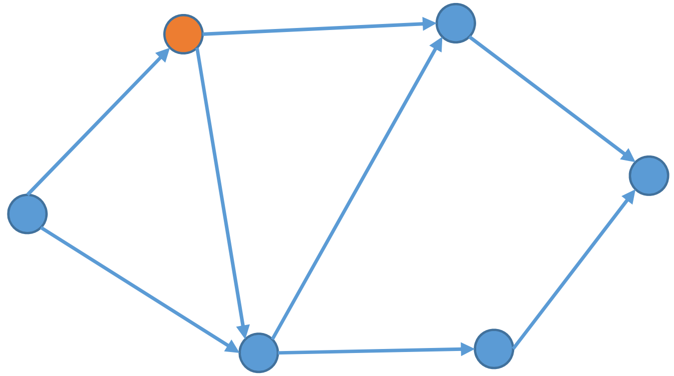
$$s_n + \sum_{a \in n^-} x_a = g_n + \sum_{a \in n^+} x_a \quad \forall n \in N$$

$$x_a \leq C_a \quad \forall a \in A$$

$$x_a \in \mathbb{R}^+ \quad \forall a \in A$$

$$g_n \in \mathbb{R}^+ \quad \forall n \in N$$

CDN网络



- CDN网络设计问题
  - 影视服务器部署位置未知，但存在有限的备选位置，存在部署成本
  - 通讯网络的通道已知、带宽已知
  - 信息流的起点未知，终点已知
  - 求服务器的部署位置，及流在网络上的分布

$$Minimize \sum_{n\in N} c_n y_n + \sum_{a\in A} c_a x_a$$

$$s.t.$$

$$g_n \le y_n \times M \quad \forall n \in N$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

$$x_a \in \mathbb{R}^+ \quad \forall a \in A$$

$$g_n \in \mathbb{R}^+ \quad \forall n \in N$$

$$y_n \in \{0,1\} \quad \forall n \in N$$

提纲

- 问题分析——从“资源”角度剖析网络设计问题
- “资源”问题的解决方案模型
  - “自由市场”模型——拉格朗日松弛算法
  - “部门协商”模型——Dantzig – Wolfe 分解与列生成算法
  - “主仆协作”模型——Benders分解算法
- 方法评述

问题分析——从“资源”角度剖析网络设计问题

$$Minimize \sum_{n\in N} c_n y_n + \sum_{a\in A} c_a x_a$$

$$s.t.$$

$$g_n \le y_n \times M \quad \forall n \in N$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

网络及用户

$$Minimize \sum_{a\in A} c_a x_a$$

$$s.t.$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

服务器

$$Minimize \sum_{n\in N} c_n y_n$$

$$\boxed{g_n} \le \boxed{y_n} \times M \quad \forall n \in N$$

问题分析——从“资源”角度剖析网络设计问题

服务器部署——背包问题

$$\text{Minimize} \sum_{n \in N} c_n y_n$$

$$s.t. \sum_{n \in N} c_n y_n \leq U$$

$$g_n \leq y_n \times M \quad \forall n \in N$$

流量生成位置 - 服务器部署
耦合关系

流量分配——最小费用流问题

$$\text{Minimize} \sum_{a \in A} c_a x_a$$

$$s.t.$$

$$s_n + \sum_{a \in n^-} x_a = g_n + \sum_{a \in n^+} x_a \quad \forall n \in N$$

$$x_a \leq C_a \quad \forall a \in A$$

服务器部署的角度

尽可能少地部署服务器

用户服务的角度

尽可能多地部署服务器

极端情况 – 所有节点都部署

问题分析——从“资源”角度剖析网络设计问题

$$Minimize \sum_{n \in N} c_n y_n + \sum_{a \in A} c_a x_a$$

$s.t.$

$$g_n \leq y_n \times M \quad \forall n \in N$$

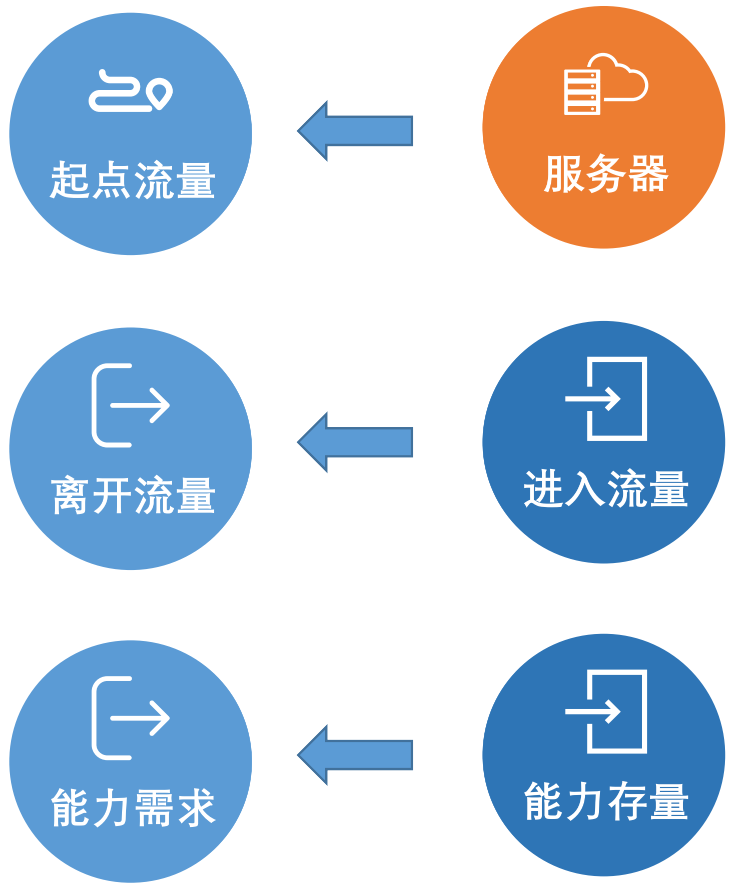
原生流量资源

$$s_n + \sum_{a \in n^-} x_a = g_n + \sum_{a \in n^+} x_a \quad \forall n \in N$$

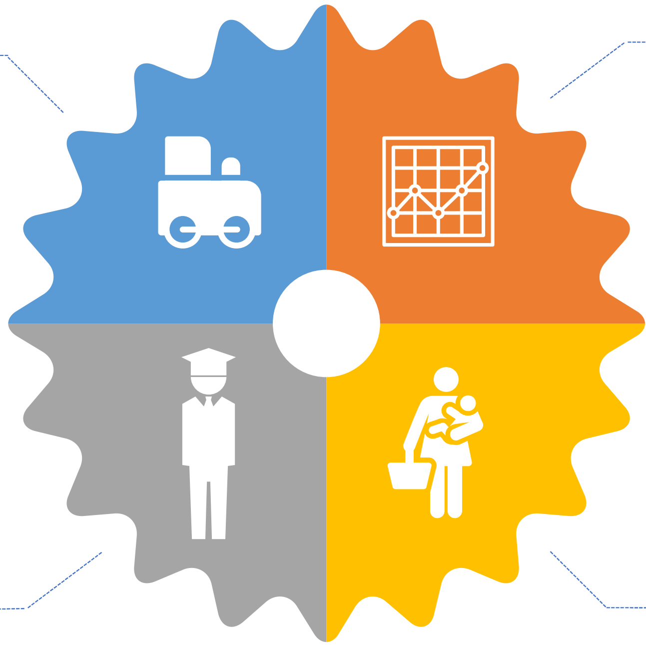
派生流量资源

$$x_a \leq C_a \quad \forall a \in A$$

能力资源



问题分析——从“资源”角度剖析网络设计问题



动车组交路

车底接续约束

车底作为移动实体资源，承载列车服务
到达前车作为出发后车车底的提供者

列车运行图

追踪间隔

$d_i - d_j \geq H \quad \rightarrow \quad d_j \leq d_i - H$

表示后车j若选择在d_j时刻出发，
则为前车预留了$d_j - H$这么多
的时间去规划自己的出发时刻。
隐式地表达了区间的**通过能力资源**。

乘务计划

乘务组的指派

人力资源
列车司乘、客舱服务的承担着

开行方案

流量分配

旅客是资源的请求者
列车服务是资源的承担者

每一条约束均代表一种“资源”

资源分配的机制模型

“自由市场”模型

“部门协商”模型

“主仆协作”模型

如何让资源分配更有效率？

探讨资源分配的机制

资源分配的机制模型

“自由市场”模型

拉格朗日松弛方法

“部门协商”模型

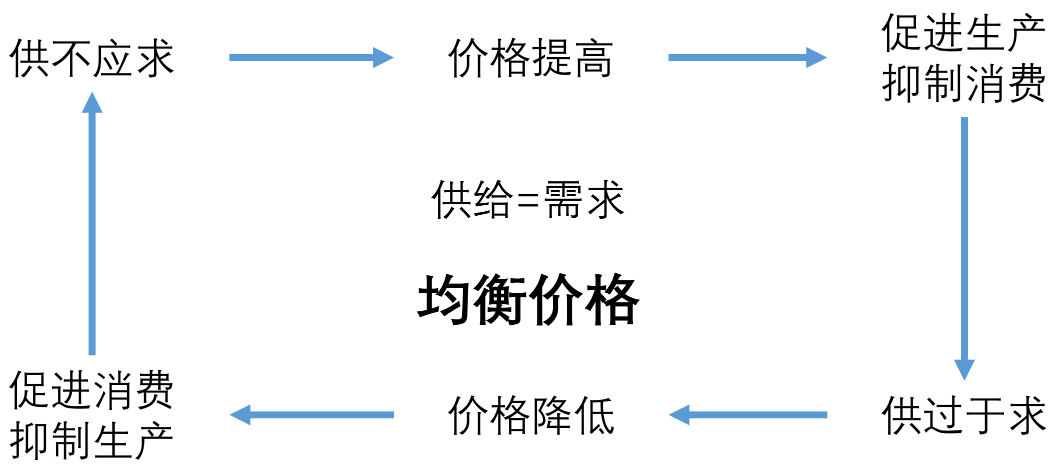
“主仆协作”模型

如何让资源分配更有效率？

探讨资源分配的机制

资源分配的机制模型（一）

■“自由市场”模型



拉格朗日松弛

$$Minimize \sum_{n\in N} c_n y_n + \sum_{a\in A} c_a x_a$$

$s.t.$

$$g_n \leq y_n \times M \quad \forall n \in N$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \leq C_a \quad \forall a \in A$$

方法的核心：资源价格

$$Minimize \sum_{n\in N} c_n y_n + \sum_{a\in A} c_a x_a + \sum_{n\in N} \boxed{\rho_n} \times (g_n - M y_n)$$

$s.t.$

$$g_n \leq y_n \times M \quad \forall n \in N$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \leq C_a \quad \forall a \in A$$

ρ_n - 对应约束的拉格朗日乘子

拉格朗日松弛

$$Minimize \sum_{n\in N}(c_n - M\rho_n)\times y_n + \sum_{a\in A} c_a x_a + \sum_{n\in N}\rho_n g_n$$

$s.t.$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n\in N$$

$$x_a \le C_a \quad \forall a\in A$$

考虑收益的服务器部署——背包问题

$$Minimize \sum_{n\in N}(c_n - M\rho_n)\times y_n$$

考虑流量生成成本的流量分配
——最小费用流问题

$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N}\rho_n g_n$$

$s.t.$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n\in N$$

$$x_a \le C_a \quad \forall a\in A$$

拉格朗日松弛

- “资源价格”的制定

价格调整的依据

$$\rho_n^{q+1} = \rho_n^q + \alpha^q \times (g_n - y_n \times M)$$

$$\rho_n^{q+1} \geq 0$$

回顾：资源分配的机制模型（一）

供不应求 → 价格提高 → 促进生产抑制消费

促进生产抑制消费 → 供过于求 → 价格降低 → 促进消费抑制生产

促进消费抑制生产 → 供不应求

供给=需求

均衡价格

$g_n > y_n \times M$ → ρ_n上升 → y_n变大 g_n变小

y_n变大 g_n变小 → $g_n < y_n \times M$ → ρ_n下降 → y_n变大 g_n变小

y_n变大 g_n变小 → $g_n > y_n \times M$

$g_n = y_n \times M$

最优解

资源分配的机制模型

“自由市场”模型

拉格朗日松弛方法

“部门协商”模型

列生成法
Dantzig – Wolfe 分解

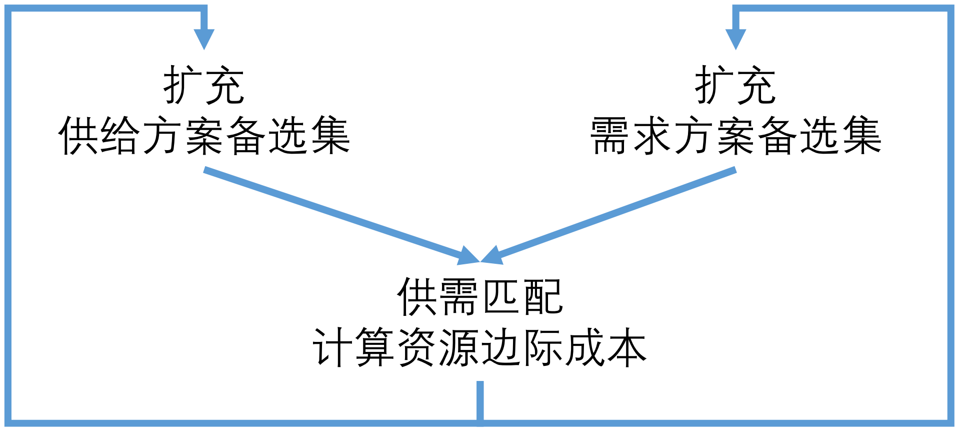
“主仆协作”模型

如何让资源分配更有效率？

探讨资源分配的机制

资源分配的机制模型（二）

- “部门协商”模型



列生成法

$$Minimize \sum_{n\in N} c_n y_n + \sum_{a\in A} c_a x_a$$

$s.t.$

$$g_n \leq y_n \times M \quad \forall n \in N$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \leq C_a \quad \forall a \in A$$

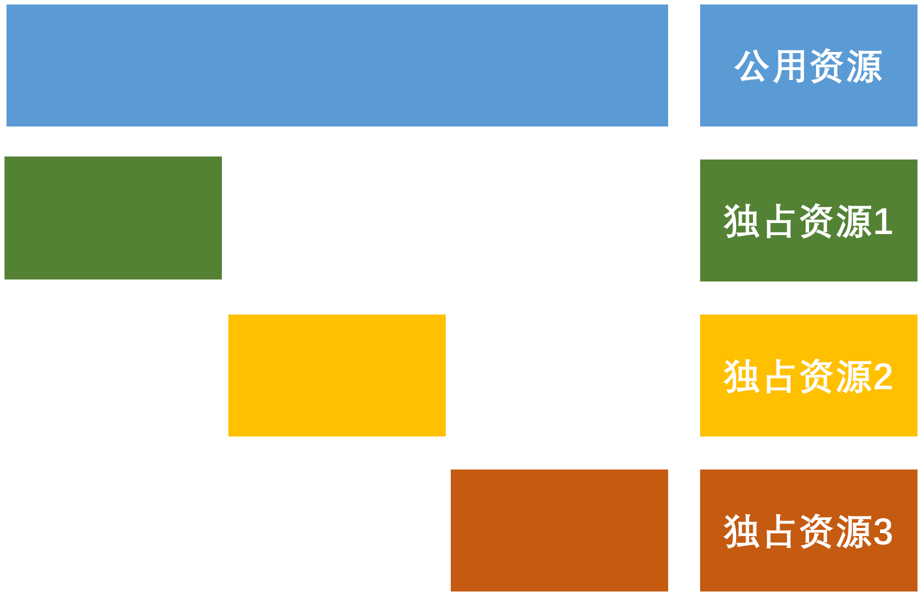
$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N} c_n y_n$$

$s.t.$

$$\boxed{g_n \qquad -y_n \times M} \leq 0 \quad \forall n \in N$$

$$\boxed{\begin{array}{c} -g_n - \sum_{a\in n^+} x_a + \sum_{a\in n^-} x_a \\ x_a \end{array}} \quad \begin{array}{l} = -s_n \quad \forall n \in N \\ \leq C_a \quad \forall a \in A \end{array}$$

列生成法



$$\text{Minimize} \sum_{a\in A} c_a x_a + \sum_{n\in N} c_n y_n$$

$$s.t.$$

$$g_n - y_n \times M \le 0 \quad \forall n \in N$$

$$-g_n - \sum_{a\in n^+} x_a + \sum_{a\in n^-} x_a = -s_n \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

子系统矩阵

列生成法

$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N} c_n y_n$$

$$s.t.$$

$$g_n - y_n \times M \le 0 \quad \forall n \in N$$

$$-g_n - \sum_{a\in n^+} x_a + \sum_{a\in n^-} x_a = -s_n \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

■资源分配策略

- ■内部资源：各自求最优解
- ■外部资源：价格化

消耗者：资源交易价格是其支出

$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N} g_n \times \pi_n$$

$$-g_n - \sum_{a\in n^+} x_a + \sum_{a\in n^-} x_a = -s_n \quad \forall n \in N$$

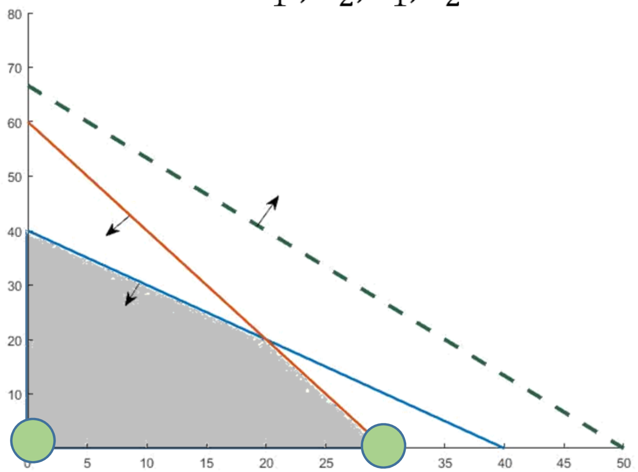
$$x_a \le C_a \quad \forall a \in A$$

生产者：资源交易价格是其收入

$$Minimize \sum_{n\in N} c_n y_n + \sum_{n\in N} (-y_n) \times M \times \pi_n$$

列生成法的原理 - 单纯型法

$$
\begin{aligned}
&\textit{maximize} && 4x_1 + 3x_2 \\
&\textit{subject to} && x_1 + x_2 + s_1 = 40 \\
& && 2x_1 + x_2 + s_2 = 60 \\
& && x_1, x_2, s_1, s_2 \geq 0
\end{aligned}
$$



0	x_1	x_2	s_1	s_2	θ_i
σ_j	4	3	0	0	$z = 0$
Row 1	1	1	1	0	40
Row 2	2	1	0	1	60

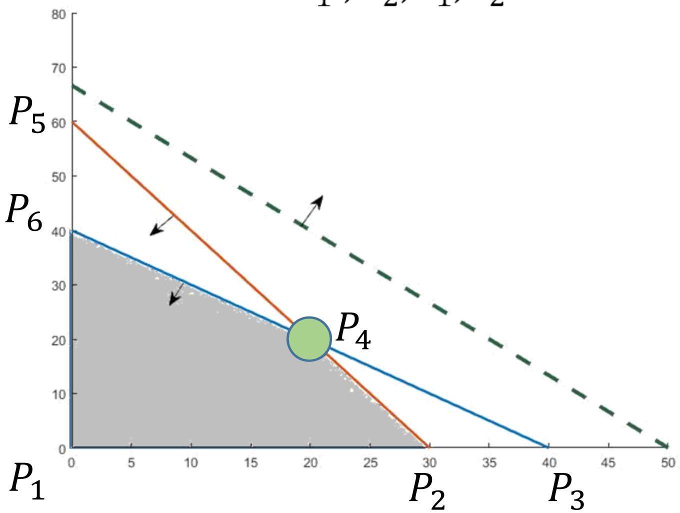
$x_1 = 0, x_2 = 0, s_1 = 40, s_2 = 60$　　目标函数值：0

	x_1	x_2	s_1	s_2	θ_i
σ_j	0	1	0	-2	$z = -120$
Row 1	0	½	1	- ½	10
Row 2	1	½	0	½	30

$x_1 = 30, x_2 = 0, s_1 = 10, s_2 = 0$　　目标函数值：120

列生成法的原理 - 单纯型法

$$\begin{aligned} maximize \quad & 4x_1 + 3x_2 \\ subject\ to \quad & x_1 + x_2 + s_1 = 40 \\ & 2x_1 + x_2 + s_2 = 60 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{aligned}$$

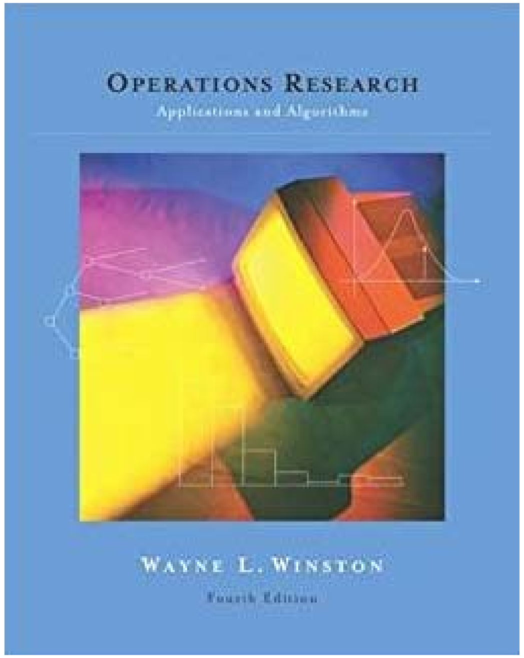


	x_1	x_2	s_1	s_2	θ_i
σ_j	0	0	-2	-1	z =-140
Row 1	0	1	2	-1	20
Row 2	1	0	-1	1	20

$x_1 = 20, x_2 = 20, s_1 = 0, s_2 = 0$　　目标函数值：140

基变量	x_1	x_2	s_1	s_2	对应的凸点
s_1, s_2	0	0	40	60	P_1
x_1, s_1	30	0	10	0	P_2
x_1, s_2	40	0	0	-10	P_3
x_1, x_2	20	20	0	0	P_4
x_2, s_1	0	40	-20	0	P_5
x_2, s_2	0	60	0	20	P_6

列生成法的原理 - 单纯型法



Winston, Wayne L., and Jeffrey B. Goldberg. *Operations research: applications and algorithms*: Thomson/Brooks/Cole, 2004.

列生成法

$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N} c_n y_n$$

$s.t.$

$$\boxed{g_n \quad -y_n \times M} \leq 0 \quad \forall n \in N$$

通过加入子问题的凸点，张成解空间

■资源分配策略

- ■内部资源：各自求最优解
- ■外部资源：参数化

$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N} g_n \times \pi_n$$

$$-g_n - \sum_{a\in n^+} x_a + \sum_{a\in n^-} x_a = -s_n \quad \forall n \in N$$

$$x_a \leq C_a \quad \forall a \in A$$

$$\sum_{n\in N} c_n y_n + \sum_{n\in N} (-y_n) \times M \times \pi_n$$

列生成法

$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N} c_n y_n$$

$s.t.$

$$\boxed{g_n \qquad -y_n \times M} \le 0 \quad \forall n \in N$$

$$\pi_n$$

$$Minimize \sum_{a\in A} c_a x_a + \sum_{n\in N} g_n \times \pi_n$$

$$-g_n - \sum_{a\in n^+} x_a + \sum_{a\in n^-} x_a = -s_n \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

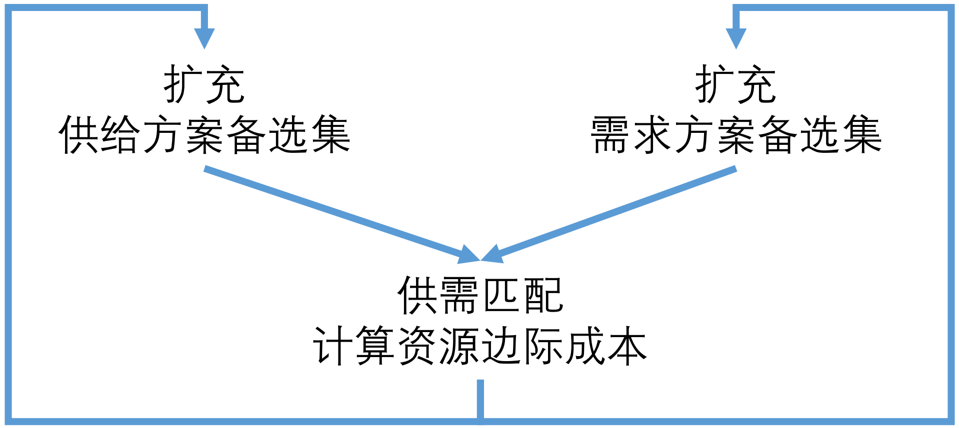
$$\sum_{n\in N} c_n y_n + \sum_{n\in N} (-y_n) \times M \times \pi_n$$

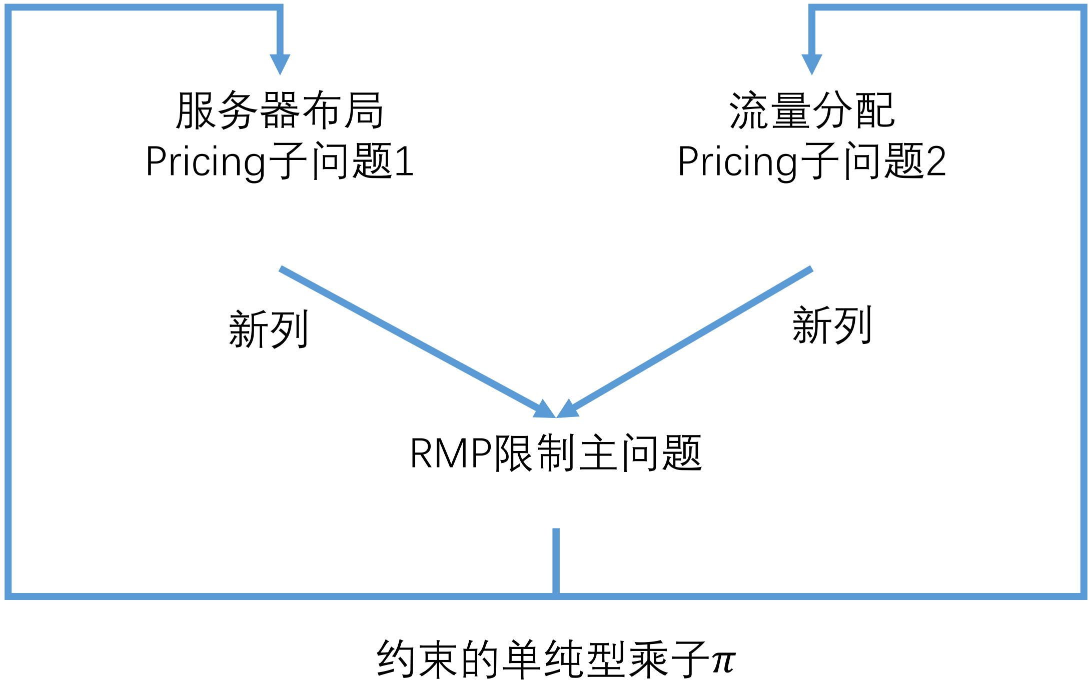
根据线性规划的**对偶理论**

- π_n是单纯性乘子、影子价格、对偶问题的解
- 其经济学含义是 资源的边际收益率
- 即多给一个单位的资源，能带来多大的收益的增加

回顾：资源分配的机制模型（二）

- “协商”模型





资源分配的机制模型

“自由市场”模型

拉格朗日松弛方法

“部门协商”模型

列生成法

Dantzig – Wolfe 分解

“主仆协作”模型

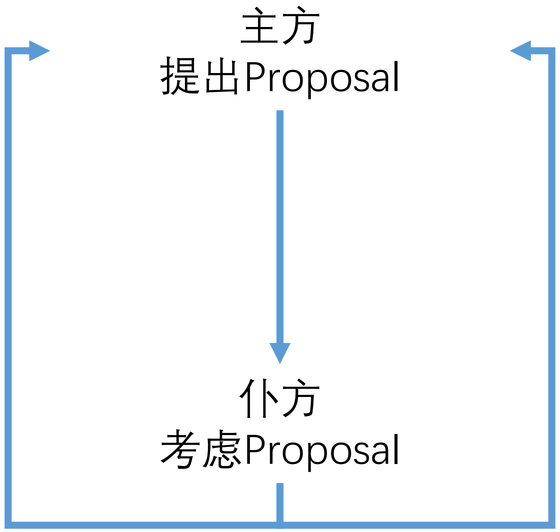
Benders 分解算法

如何让资源分配更有效率？

探讨资源分配的机制

资源分配的机制模型（三）

■“主仆协作”模型



Benders主问题和子问题的构造

原问题

$$minimize\ c^T x + f^T y$$

$$subject\ to\ Ax + By = b$$

$$x, y \geq 0$$

Bender's主问题

$$minimize\ f^T y + q(y)$$

$$subject\ to\ \ y \geq 0$$

Bender's子问题

$$minimize\ c^T x$$

$$subject\ to\ \ Ax = b - B\bar{y}$$

$$x \geq 0$$

主问题求解得到的y

Bender's子问题的对偶问题

$$maximize\ \alpha^T(b - B\bar{y})$$

$$subject\ to\ \boxed{A^T\alpha = c}$$

$$\alpha\ unrestricted.$$

Benders

$$\text{Minimize} \sum_{n\in N} c_n y_n + \sum_{a\in A} c_a x_a$$

$s.t.$

$$g_n \le y_n \times M \quad \forall n \in N$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

$$x_a \le C_a \quad \forall a \in A$$

- 情况1：求得最优解，主=子
- 情况2：不可行
- 情况3：可行，但主≠子

主问题：服务器部署

$$\text{Minimize} \sum_{n\in N} c_n y_n$$

子问题：流量分配

y_n给定的条件下

$$\text{Minimize} \sum_{n\in N} c_n \overline{y_n} + \sum_{a\in A} c_a x_a$$

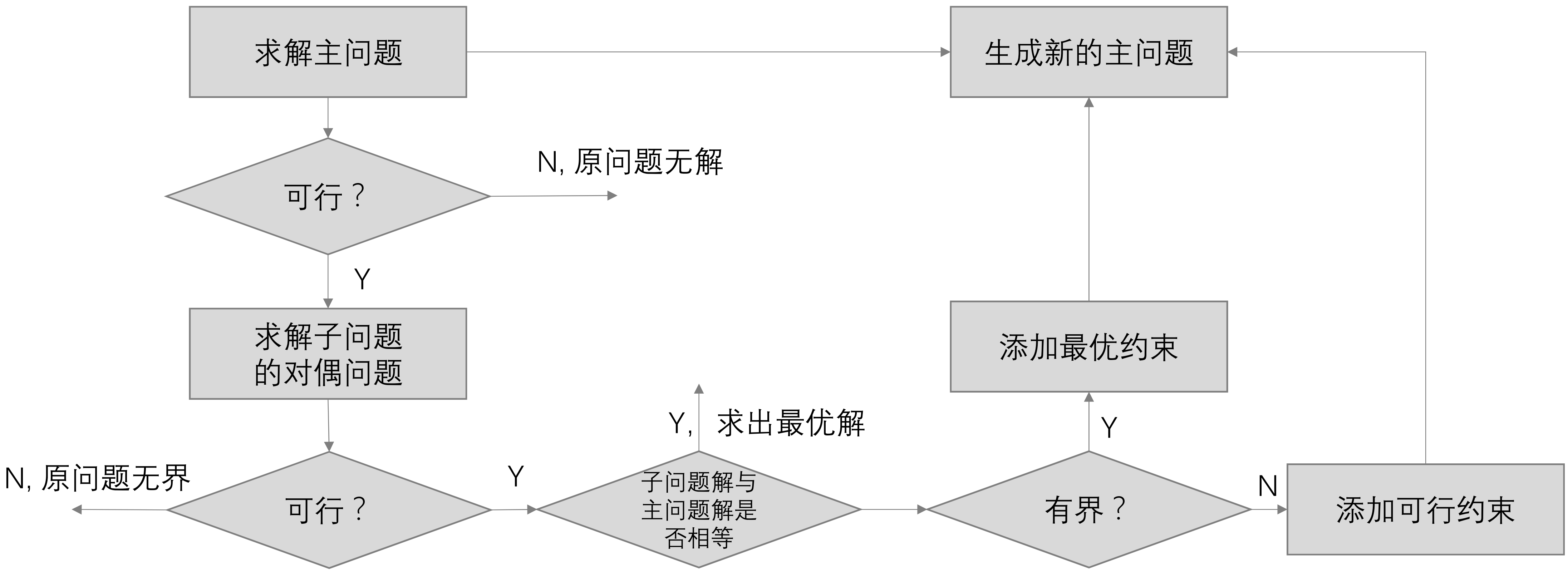
$s.t.$

$$g_n \le \overline{y_n} \times M \quad \forall n \in N$$

$$s_n + \sum_{a\in n^-} x_a = g_n + \sum_{a\in n^+} x_a \quad \forall n \in N$$

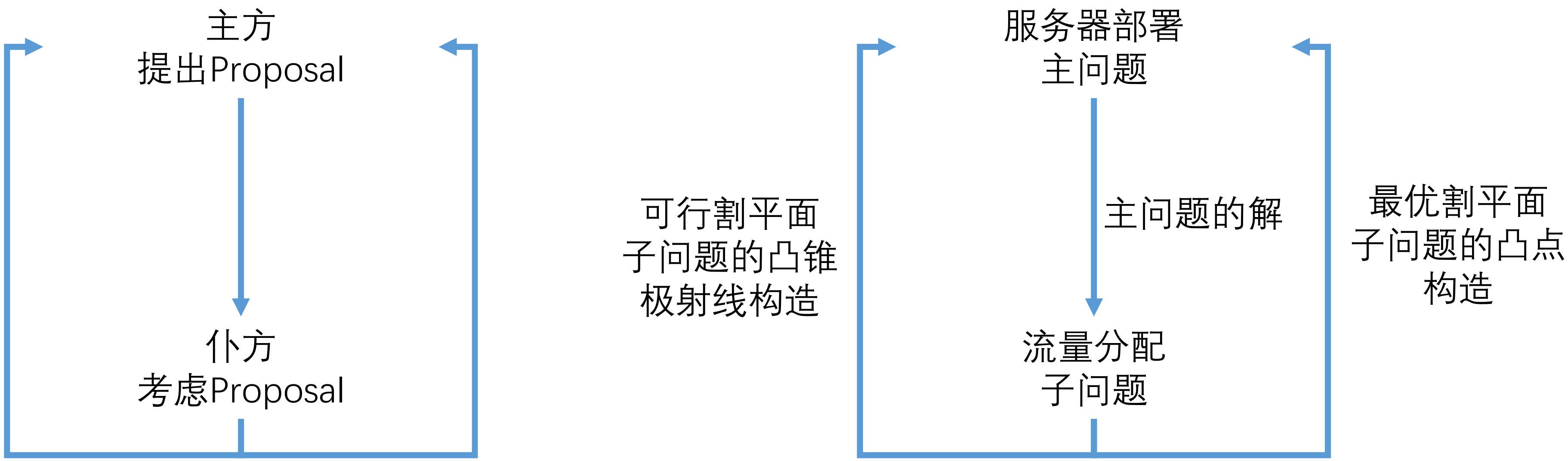
$$x_a \le C_a \quad \forall a \in A$$

Bender’s 分解的求解框架



回顾：资源分配的机制模型（三）

■“主仆协作”模型



计算机程序实现

- Original model （By invoking Gurobi）
- Lagrangian Relaxation
- Branch & Bound + Column Generation (**recursion** version)
- Benders decomposition

https://github.com/lzw37/DecompositionAlgorithm

方法评述：优点

LR
拉格朗日松弛方法

- 快
- 可分解为并行子问题
- 实现简单

CG
列生成算法

- 高、大、上
- 对于子问题边界特征明显的问题有奇效
- 形式严谨，逻辑性强

BD
Benders分解算法

- 理解相对容易
- 针对一些“主仆”特征明显的问题效果好

方法评述：缺点

LR
拉格朗日松弛方法

- 乘子更新难把握
- 求出的解多半是不可行的
- 对于大M问题，界极其不好

CG
列生成算法

- 要重新构造RMP，难度大
- 遇到退化解容易陷在一个地方很久
- 子系统较少时效果不明显的
- 整数规划的分支定界

BD
Benders分解算法

- 主问题如果离原问题太远，迭代很慢
- 主问题如果本身过于复杂，则分解无用

需要继续讨论的问题…

- 大 M 对求解的影响
  - 线性松弛提供的最优界不紧的问题
  - 大 M 的约束中的0-1变量是一个“开关”，起到“四两拨千斤”的作用
- 基于列生成法的分支定界
  - 原问题的线性松弛下界质量要非常好，才能避免分支定界困难
- 启发式策略在精确算法中的应用
  - 启发式方法 在 精确算法 的步骤中可以起到非常积极的作用（CPLEX、Gurobi等引擎已实践）
- **Benders** 分解算法的收敛速度
  - 主问题的最优解要尽可能接近原问题的最优解，才能避免反复地进行加割平面操作

可参考的文献

- Bradley, Stephen, Arnoldo Hax, and Thomas Magnanti. "**Applied mathematical programming**." (1977).
- Birge, John R. "**Linear Programming**: Foundations and Extensions." Iie Transactions 31.3 (1999): 278-278.
- Winston, Wayne L., and Jeffrey B. Goldberg. **Operations research**: applications and algorithms: Thomson/Brooks/Cole, 2004.
- Fisher, Marshall L. "The **Lagrangian relaxation** method for solving integer programming problems." *Management science*27.1 (1981): 1-18.
- Taşkin, Zeki Caner. "**Benders decomposition**." *Wiley Encyclopedia of Operations Research and Management Science* (2010).
- 朱道立. *大系统优化理论和应用*. 上海交通大学出版社, 1987.

一点思考——数学方法对于研究的意义

工　具

作为工具的数学方法

可以为我们解决各种实际问题
写论文、做算例用

作为抽象自现实的数学方法

更加深刻地认识我们原本以为熟悉的世界
跳出原有的领域看待问题
理论创新

灵　感

大规模整数规划分解算法介绍

以通信网络节点布局优化问题为例

廖正文

2018. 3. 22