

# 一范数下设施选址逆优化问题的求解方法

Discount

## 摘要

我们使用了经典的列生成算法对无容量和有容量设施选址的逆优化问题进行了计算，其子问题就是求解原问题，这样求解得到的就是对应设施选址的逆优化最优值。对子问题进行松弛得到非整数解计算可以得到逆优化问题的上界，而对子问题松弛得到的非整数解取临近整数可以得到逆优化问题的下界。最后我们将一种启发式算法推广至有容量设施选址的逆优化问题，并将其结果与最优值和列生成子问题松弛解进行了比较。

**关键词：**设施选址问题、逆优化、列生成算法、启发式算法

## 1 引言

### 逆优化综述

逆优化问题最初已经有很多人进行了研究同时设施选址由于设施选址问题本身是 NP 难问题，它的逆问题就变得更为复杂。若是

在现实生活中，由于设施已经建立在某些固定的地方，不容易更改，我们往往需要固定哪些设施开启以提供便利。对应到模型中，即给定了开设哪些设施，需要调整量达到最小。

而设施选址的逆优化的问题很少有人研究

## 2 无容量限制

设施选址问题通常被提出如下：假设有  $m$  个设施和  $n$  个客户。我们希望选择（1）哪些设施要打开，以及（2）哪些打开的设施要用于向哪些客户提供，以便以最低的成本满足某些固定需求。引入以下记号，令  $f_i$  表示开启设施  $i$  的固定成本， $i \in M, M = \{1, \dots, m\}$ 。  $r_{ij}$  表示运输商品从设施  $i$  到客户  $j$  的成本， $j \in N, N = \{1, \dots, n\}$ 。更进一步，令  $d_j$  表示客户  $j$  的需求，并且假设每一个设施有一个最大输出限制，令  $k_i$  表示设施  $i$  能够生产出的最大商品量，即表示设施  $i$  的容量。

则无容量限制设施选址问题定义如下：

$$\begin{aligned}
& \min \sum_{i=1}^n \sum_{j=1}^m r_{ij} u_{ij} + \sum_{i=1}^n f_i v_i \\
& \text{s.t.} \quad \sum_{i=1}^n u_{ij} = 1, \forall j \in N \\
& \quad \sum_{j=1}^m u_{ij} \leq T v_i, \forall i \in M \\
& \quad u_{ij} \in \{0, 1\}, \quad v_i \in \{0, 1\}
\end{aligned}$$

用  $c = (f, r)$  表示成本，并且用另一种方式改写包含足够大的  $T$  的上式，得到更简洁的表达形式：

$$\begin{aligned}
F(c) &= \min \sum_{i=1}^n \sum_{j=1}^m r_{ij} u_{ij} + \sum_{i=1}^n f_i v_i \\
& \text{s.t.} \quad \sum_{i=1}^n u_{ij} = 1, \forall j \in M, \forall j \in N \\
& \quad u_{ij} \leq v_i, \forall i \in M, \forall j \in N \\
& \quad u_{ij} \in \{0, 1\}, \quad v_i \in \{0, 1\}
\end{aligned}$$

类似的，有容量限制的设施选址问题定义如下：

$$\begin{aligned}
& \min \sum_{i=1}^n \sum_{j=1}^m c_{ij} u_{ij} + \sum_{i=1}^n f_i v_i \\
& \text{s.t.} \quad \sum_{i=1}^n u_{ij} = 1, \forall j \in N \\
& \quad \sum_{j=1}^m d_j u_{ij} \leq k_i v_i, \forall i \in M \\
& \quad 0 \leq u_{ij} \leq 1, \forall j \in N, \forall i \in M \\
& \quad v_i \in \{0, 1\}, \forall i \in M
\end{aligned}$$

逆优化问题分为最优值和最优解问题，为方便显示我们的结果，我们主要考虑最优解问题，而最优值问题也可用相同思路求解。给定原问题的一个可行解  $x^0$ ，通过调整原问题的成本  $c \rightarrow d$ ，使得  $x^0$  成为调整后问题的最优解。而逆优化则是求解在某个范数下使得这个调整量最小的问题，为方便以及更贴近现实，我们以一范数为例，即表示调整量之和最小。往往对于设施成本和运输成本的调整权重并不一样，可以分别设为  $w$

设施选址的逆优化问题定义如下：

$$\begin{aligned} & \min_d \|d - c\|_1 \\ \text{s.t. } & dx^0 = F(d) \\ & d \in \mathbb{R}^{m+n} \end{aligned}$$

由约束式可以看出， $x^0$  需要满足  $dx^0 \leq dx', x' \in \Upsilon$ 。其中  $\Upsilon$  是满足原问题的所有可行解  $x'$  的集合。若从逆优化的定义出发建立求解模型，上述集合中包含可行解有指数个。由此可以看出设施选址的逆优化问题比原问题更加复杂，是 NP-难问题。

### 3 列生成算法求解

从设施选址逆优化问题的定义过程中，我们可以看出原优化问题和逆优化问题有着一定的联系，即逆优化中的给定解对应的成本需要小于所有满足原问题约束条件的解对应的成本。列生成算法作为一种有效求解规模较大线性规划问题的方法。我们自然想到列生成算法中子问题即是求解满足原问题约束的，因而列生成算法可以用来求解设施选址的逆优化问题。

原逆优化模型为

$$\begin{aligned} & \min |c - c_0| \\ \text{s.t. } & V_0 = cx^0 \leq cx', x' \in \Upsilon \end{aligned}$$

将列生成算法应用于逆 UFL 问题，将上式对应的设施和运输部分写开可以得到该逆问题 UFL 的主问题

$$\begin{aligned} & \min \sum_{i=1}^m (a_i + b_i) + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} + d_{ij}) \\ \text{s.t. } & (a_i - b_i + f_i^0)(v'_i - v_i^0) + (c_{ij} - d_{ij} + r_{ij}^0)(u'_{ij} - u_{ij}^0) \geq 0 \\ & (v'_i, u'_{ij}) \in \mathbb{I} \end{aligned}$$

其中  $(v'_i, u'_{ij})$  属于一个限制集  $\mathbb{I}$ ，集合中的元素由子问题不断生成加入其中。而主问题得到的  $(a^*, b^*, c^*, d^*)$  带入到子问题中求解。

子问题如下：

$$\min \quad \sum_{i=1}^m (a_i^* - b_i^* + f_i^0)(v_i' - v_i^0) + \sum_{i=1}^m \sum_{j=1}^n (c_{ij}^* - d_{ij}^* + r_{ij}^0)(u_{ij}' - u_{ij}^0) \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^m u_{ij}' = 1 \quad \forall j \in L \quad (2)$$

$$u_{ij}' \leq v_i' \quad \forall i \in F, \forall j \in L \quad (3)$$

$$u_{ij}' \geq 0 \quad \forall i \in F, \forall j \in L \quad (4)$$

$$u_{ij}' \in \{0, 1\}, \quad v_i' \in \{0, 1\} \quad (5)$$

子问题求出得到  $(v^*, u^*)$ , 如果目标函数值小于 0, 将得到的  $(v^*, u^*)$  加入到主问题的限制集中。如果目标函数值大于 0, 程序结束。

注意到子问题实际上是 UFL 原问题, 由于 UFL 问题是 NP-难问题, 因此 UFL 逆问题也是 NP-难问题。

可以对子问题进行如下的处理方法:

1. 精确求解子问题 UFL, 则可以得到 UFL 逆问题的最优解。
2. 近似求解子问题 UFL, 设定循环次数, 提前结束子问题的求解可以得到 UFL 逆问题的下界。
3. 将子问题松弛得到分数解, 则可以得到 UFL 逆问题的上界。

### 3.1 减少约束得到下界

注意这部分得到的 gap 较大

### 3.2 启发式得到上界

在 2010 文章中, 定义了关于设施选址问题的

$$\mathbb{C}_{\text{UFL}}^x = \left\{ (v, u) \in \mathbb{R}^{m+mn} : \sum_{i \in M} u_{ik} = 1, \forall k \in V, v_i - u_{ik} \geq 0, \forall i \in M, \forall k \in V, u_{ik} \geq 0, \forall i \in M, \forall k \in V \right\}$$

$$\begin{aligned}
& \min \sum_{i \in M} (\tau_i^f + \eta_i^f) + \sum_{i \in M} \sum_{k \in V} (\tau_{ik}^r + \eta_{ik}^r) \\
& \text{s.t.} \quad \sum_{k \in V} \pi_k \geq \nu_{\text{UFL}}, \\
& \quad \sum_{k \in V} \varrho_{ik} = \bar{f}_i, \forall i \in M, \\
& \quad \pi_k - \varrho_{ik} + \zeta_{ik} = \bar{r}_{ik}, \quad \forall i \in M, \forall k \in V, \\
& \quad \sum_{i \in M} \bar{f}_i v_i^0 + \sum_{i \in M} \sum_{k \in V} \bar{r}_{ik} u_{ik}^0 = \nu_{\text{UFL}}, \\
& \quad \tau_i^f - \eta_i^f = \bar{f}_i - f_i, \forall i \in M \quad \tau_{ik}^r - \eta_{ik}^r = \bar{r}_{ik} - r_{ik}, \forall i \in M, \forall k \in V, \\
& \quad \nu_{\text{UFL1}} \leq \nu_{\text{UFL}} \leq \nu_{\text{UFL2}}, \varrho_{ik} \geq 0, \zeta_{ik} \geq 0, \forall i \in M, \forall k \in V.
\end{aligned} \tag{6}$$

其中  $\bar{f}_i$  和  $\bar{r}_{ik}$  为改变后的开启设施成本和运输成本。相对应的,  $f_i$  和  $r_{ik}$  为改变前的成本。 $\pi_k, \varrho_{ik}, \zeta_{ik}$  分别为原 UFL 约束

$$\sum_{i \in M} u_{ik} = 1, v_i - u_{ik} \geq 0, u_{ik} \geq 0$$

对应的对偶变量。

$v_i^0, u_{ik}^0$  则对应于给定的可行解使之成为改变成本后的最优解。 $\nu_{\text{UFL}}$  是原 UFL 的最优值, 需要在一定的范围内取值使得问题可行。

$\tau_i^f, \eta_i^f, \tau_{ik}^r, \eta_{ik}^r$  则对应于求解范数一下引入的变量。

这部分 gap 很小, 实际上对应到 (integrality gap)

## 4 求解 lower bound 的一些方法

### 4.1 利用邻域的想法

有如下约束限制可以帮助给定的可行解在一定范围内是最优的

约束一 (横向移动): 对于给定的可行解 (0-1) 对应的  $r_{ik} = 1$  需要小于其他所有  $r_{ik} = 0$

约束二 (0-1 纵向移动):  $f$  中为 0 的部分需要确保  $f_i + \min_i r_{ij} \geq \min_i r_{ik}$

约束三 (纵向移动): 对于仅有一个  $r_{ik} = 1$  的行, 考虑  $r_{ik} + f_i \leq \min_j r_{jk}$

约束四: 考虑  $f_i = 1$  中  $r_{ik} = 1$  个数大于 1 的行则有  $\max f_i + \sum r_{ik} \leq \min$  对应列的  $r_{ik}$

## 5 计算结果

### 5.1 简单的例子

为说明逆优化的效果

## 5.2 不同结果比较

计算结果表明, 稀疏度并不影响 gap 以及设施开设成本和运输成本的大小关系不影响 gap

## 6 有容量限制

### 6.1 列生成算法得到最优解

### 6.2 启发式得到上界

对于有容量限制的设施选址逆优化问题, 多了一个约束  $\sum_k d_k u_{ik} \leq k_i v_i, i \in M$ 。同样利用锥优化的方法, 添加对应上述约束的对偶变量  $t$ , 我们可以得到有容量限制的设施选址问题的启发式算法

$$\begin{aligned}
& \min \sum_{i \in M} (\tau_i^f + \eta_i^f) + \sum_{i \in M} \sum_{k \in V} (\tau_{ik}^r + \eta_{ik}^r) \\
& \text{s.t.} \quad \sum_{k \in V} \pi_k \geq \nu_{\text{UFL}}, \\
& \quad k_i t_i + \sum_{k \in V} \varrho_{ik} = \bar{f}_i, \forall i \in M, \\
& \quad \pi_k - \varrho_{ik} + \zeta_{ik} - d_k t_i = \bar{r}_{ik}, \quad \forall i \in M, \forall k \in V, \\
& \quad \sum_{i \in M} \bar{f}_i v_i^0 + \sum_{i \in M} \sum_{k \in V} \bar{r}_{ik} u_{ik}^0 = \nu_{\text{UFL}}, \\
& \quad \tau_i^f - \eta_i^f = \bar{f}_i - f_i, \forall i \in M \quad \tau_{ik}^r - \eta_{ik}^r = \bar{r}_{ik} - r_{ik}, \forall i \in M, \forall k \in V, \\
& \quad \nu_{\text{UFL}1} \leq \nu_{\text{UFL}} \leq \nu_{\text{UFL}2}, \varrho_{ik} \geq 0, \zeta_{ik} \geq 0, \forall i \in M, \forall k \in V.
\end{aligned} \tag{7}$$

注意这部分可以通过改变需求和供给的大小来退化得到无容量限制

## 7 进一步改进的想法

对于列生成的方法, 实际上是限制集的选取的问题, 限制集越接近