

无容量限制设施选址逆优化问题的求解方法

Discount

摘要

逆优化问题是给定一个可行解 (值) 通过最小化目标函数中参数的改变量使得该可行解 (值) 成为改变后问题的最优解 (值) 的问题。对于本是 NP-难问题的设施选址问题, 其逆优化问题变得更为复杂。研究了使用经典的行生成算法对无容量设施选址的逆优化问题进行计算, 并给出了得到逆优化问题上下界的启发式方法。该方法基于对子问题的线性松弛求解给出上界和利用邻域搜索的方式给出下界。数值结果表明可行上界得到的结果与最优值差距极小, 极大地提高了求解该逆优化问题的效率。所提到的启发式算法还可以推广至有容量设施选址逆优化问题的计算。

关键词: 设施选址问题、逆优化、行生成算法、启发式算法

Abstract

The inverse optimization problem is to give a feasible solution (value) by minimizing the change of parameters in the objective function so that the feasible solution (value) becomes optimal after the change. For the facility location problem which is NP-hard, its inverse optimization problem becomes more complex. In this paper, the classical row generation algorithm is used to calculate the inverse optimization problem of uncapacitated facility location, and two heuristic methods are given to get the upper and lower bounds of the inverse optimization problem. One gives the upper bound based on the linear relaxation of the subproblem and the lower bound is given by the way of local search. The numerical results show that the difference between the feasible upper bound and the optimal value is very small, which greatly improves the efficiency of solving the inverse optimization problem. The proposed heuristic algorithm can also be extended to the calculation of the inverse optimization of the capacitated facility location problem.

Keywords: uncapacitated facility location problem, inverse optimization problem, row generation method, heuristic

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

本文研究了

本文安排如下：第一节

2 UFL 逆优化问题的模型

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， $j \in N, N = \{1, \dots, n\}$ 。

则无容量限制设施选址问题定义如下：

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

用 $c = (f, r)$ 表示成本，并且用更为简洁的方式改写包含足够大容量 T 的上式，得到下式：

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

逆优化问题通常分为最优值和最优解问题，当给定原问题的一个可行解 x^0 ，通过调整原问题的成本 c 为 d ，使得 x^0 成为调整后问题的最优解。而逆优化最优解问题则是求解在某个范数下使得这个调整量最小的问题。当给定原问题的一个可行值 V_0 ，称此时对应的问题为逆优化最优值问题。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调整量的权重并不一样，可以分别设为 w_i^f 和 w_{ij}^r 。

设施选址的逆优化问题定义如下：

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

由约束式可以看出, x^0 需要满足 $dx^0 \leq dx', x' \in \Upsilon$ 。其中 Υ 是满足原问题的所有可行解 x' 的集合。对于每一个客户 $j \in N$ 可以被任一设施服务, 共有 $|M| = m$ 种选择。因而对于所有客户而言一共会有 m^n 种不同情况。即满足原问题约束条件的可行解有 m^n 个。因此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的个数是指数个。由此可以看出设施选址的逆优化问题是 NP-难问题。事实上, 设施选址逆优化问题是 NP-难问题。

3 行生成算法求解

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。行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。我们自然想到行生成算法中子问题即是求解满足原问题约束的, 因而行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

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

为方便显示我们的结果, 我们主要考虑最优解问题, 而对于最优值问题, 给定 V^0 而不是 x^0 , 根据该式 $V^0 = cx^0$ 也可用相同思路求解。

将行生成算法应用于 UFL 的逆优化问题, 将上式对应的设施和运输部分用 (f, c) 分别表示。根据 2001 年文献 Ahuja 在一范数下可以得到 UFL 逆优化问题的主问题为

$$\begin{aligned} & \min \quad \sum_{i=1}^m w_i^f(a_i + b_i) + \sum_{i=1}^m \sum_{j=1}^n w_{ij}^r(c_{ij} + d_{ij}) \\ \text{s.t. } & (a_i - b_i + f_i^0)(v_i' - v_i^0) + (c_{ij} - d_{ij} + r_{ij}^0)(u_{ij}' - u_{ij}^0) \geq 0 \\ & a_i \geq 0, \quad b_i \geq 0, \quad \forall i \in M \\ & c_{ij} \geq 0, \quad d_{ij} \geq 0, \quad \forall i \in M, \forall j \in N \\ & (v_i', u_{ij}') \in \mathbb{I} \end{aligned}$$

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：

$$\begin{aligned}
T = \min \quad & \sum_{i=1}^m (a'_i - b'_i + f_i^0)(v'_i - v_i^0) + \sum_{i=1}^m \sum_{j=1}^n (c'_{ij} - d'_{ij} + r_{ij}^0)(u'_{ij} - u_{ij}^0) \\
\text{s.t.} \quad & \sum_{i=1}^m u'_{ij} = 1, \quad \forall j \in N \\
& u'_{ij} \leq v'_i, \quad \forall i \in M, \forall j \in N \\
& u'_{ij} \geq 0, \quad \forall i \in M, \forall j \in N \\
& u'_{ij} \in \{0, 1\}, \quad v'_i \in \{0, 1\}
\end{aligned}$$

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。如果目标函数值不小于 0, 程序结束。注意到子问题实际上是无容量限制设施选址的原问题, 因此该设施选址的逆问题复杂度要大于原问题, 也是 NP-难问题。

求解设施选址问题的行生成算法详细步骤如下：

输入：1) 可行解： $x^0 = (v_i^0, u_{ij}^0)$ ；2) 原设施成本： $c^0 = (f_i^0, r_{ij}^0)$ ；

输出：1) 最优解 (a^*, b^*, c^*, d^*) ；2) 最小的总成本改变量为 $\sum_{i=1}^m (a_i^* + b_i^*) + \sum_{i=1}^m \sum_{j=1}^n (c_{ij}^* + d_{ij}^*)$ ；
3) 对应原成本的改变量为 $f_i - f_i^0 = a_i^* - b_i^*, r_{ij} - r_{ij}^0 = c_{ij}^* - d_{ij}^*$ 。

初始化：初始限制集包含对于原问题的可行解即可。为方便且不再重新生成可行解, 可取初始限制集为给定的可行解即 $\mathbb{I} = \{(v_i^0, u_{ij}^0)\}$ 。

步骤 1：求解此时的主问题得到最优解 (a', b', c', d') 。

步骤 2：将步骤 1 得到的最优解 (a', b', c', d') 带入到子问题中, 求解子问题这一整数规划得到最优解 $x'^* = (v'^*, u'^*)$, 和目标函数值 T 。

步骤 3：如果目标函数值 $T < 0$, 将得到的 x'^* 加入到限制集中 $\mathbb{I} = \mathbb{I} \cup x'^*$, 回到步骤 1, 重新求解主问题; 如果目标函数值 $T \geq 0$, 则结束算法。此时得到最优解 (a^*, b^*, c^*, d^*) 。

由上述的行生成方法, 我们可以知道要求得逆优化的最优解就需要得到尽可能精确的主问题的限制集, 即主问题中的限制条件。由于主问题的限制集是由子问题不断生成得到的, 可以看出行生成方法的核心是子问题的效率, 即子问题求解越精确, 逆优化的最优解越精确。但由于子问题是设施选址的原问题, 因而可以通过对子问题进行松弛, 牺牲一定求解精度的情况下简化求解过程。

当对子问题进行如下几种处理方法时, 我们可以相应得到行生成方法的几点推论：

处理方法 1：严格求解子问题

推论 1. 当严格求解子问题时, 循环结束时此时得到 UFL 逆优化问题的最优解。

说明：由于子问题是不断从 UFL 问题的约束中筛选出满足条件的解 (v'^*, u'^*) , 严格求解子问题直到循环结束, 筛选出来的解集 \mathbb{I} 满足 UFL 所有的约束条件, 根据逆优化问题的定义, 根据此时的解集求解主问题得到的即是逆优化问题的最优解。

处理方法 2: 严格求解子问题但设定一定的循环次数

推论 2. 严格求解子问题时, 设定一定的循环次数, 此时求解得到 UFL 逆优化问题的下界。

说明: 当循环一定次数时, 此时子问题得到的限制集较最优时少, 因而主问题的约束条件较最优时少。实际上减少了一些所必须的约束, 此时主问题求解得到的目标函数值小于最优值, 因而得到逆优化问题的下界。

处理方法 3: 对子问题中得到的松弛解再利用邻域搜索得到

推论 3. 对子问题的约束条件进行线性松弛求解得到非整数解, 将该非整数解四舍五入为整数解, 再利用邻域搜索的方法得到较好

对于子问题

对于邻域搜索采用增加, 交换和减少的方式得到整数

说明:

处理方法 4: 直接对子问题进行线性松弛求解

推论 4. 将无容量限制设施选址的子问题松弛得到分数解, 则可以得到对应逆优化问题的上界。

说明: 当对子问题的整数规划问题进行线性松弛时, 可以得到非整数解。此时会得到子问题的下界, 对应的主问题会得到约束更为严格的限制集。主问题对应的可行域变小, 使得最终得到的结果大于最优解, 此时得到逆优化问题的上界。

3.1 启发式得到上界

有学者利用合作博弈的思想和锥优化的方法得到了一种无容量限制设施选址逆优化的启发式算法, 我们将主要阐述他的方法并且显示了我们得到上界的启发式算法与之等价, 从而可以得到我们的启发式算法在多项式时间内得到较为优异的结果。

在 2010 文章中, Letchford 定义了关于设施选址问题的最小整数 (Integer Minimization) 博弈, 并且给出了无容量限制设施选址问题的锥可行域在 x 上的投影。

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

事实上上述可行域只是将原 UFL 的整数条件进行了松弛

Liu 利用该可行域和锥优化的方法对 UFL 的逆优化问题得到了如下的启发式算法

$$\begin{aligned}
& \min \sum_{i \in M} (\tau_i^f + \eta_i^f) + \sum_{i \in M} \sum_{j \in V} (\tau_{ij}^r + \eta_{ij}^r) \\
& \text{s.t.} \quad \sum_{j \in V} \pi_j \geq \nu_{\text{UFL}}, \\
& \quad \sum_{j \in V} \varrho_{ij} = \bar{f}_i, \forall i \in M, \\
& \quad \pi_j - \varrho_{ij} + \zeta_{ij} = \bar{r}_{ij}, \quad \forall i \in M, \forall j \in V, \\
& \quad \sum_{i \in M} \bar{f}_i v_i^0 + \sum_{i \in M} \sum_{j \in V} \bar{r}_{ij} u_{ij}^0 = \nu_{\text{UFL}}, \\
& \quad \tau_i^f - \eta_i^f = \bar{f}_i - f_i^*, \forall i \in M \quad \tau_{ij}^r - \eta_{ij}^r = \bar{r}_{ij} - r_{ij}, \forall i \in M, \forall j \in V, \\
& \quad \varrho_{ij} \geq 0, \zeta_{ij} \geq 0, \forall i \in M, \forall j \in V.
\end{aligned} \tag{1}$$

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

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

对应的对偶变量。

v_i^0, u_{ij}^0 则对应于给定的可行解使之成为改变成本后的最优解。 ν_{UFL} 是原 UFL 的最优值。

$\tau_i^f, \eta_i^f, \tau_{ij}^r, \eta_{ij}^r$ 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 计算结果

本文使用 Windows 10 操作系统, 内存 8 GB 处理器为 Intel Core i7-8700 的 PC 电脑上进行了所有的数值实验。所有算法均通过 Matlab R2019a 调用 Gurobi 求解器去实现。

通过对按照规模 (m, n) 大小 m 和 n 之间比例关系稀疏矩阵还有设施成本与运输成本之间关系

为方便起见, 在接下来的计算中取 $w^f = 1, w^r = 1$ 。在实际不能改变设施成本时, 可以设置相应的权重 $w^f = \infty$ 。

在数值实验中, 我们对于 5 种不同规模随机生成了对应规模下 50 种可能的实例。

$(m, n) = (20, 20) (30, 30) (40, 40) (50, 50)$

$f_i = [100, 200] \quad r_{ij} = [1, 100] [100, 200]$

分别 $(20, 40) (30, 60) (40, 80) (50, 100)$

$(20, 60) (30, 30) (40, 40) (50, 50)$

我们采用当最优改变量不为 0 时, (上界改变量-最优改变量)/最优改变量 (%) 的方式来比较 gap 的大小同时当给定可行解刚好是最优解时, 此时最优改变量为 0, 可以利用总的改变量来衡量 gap 或者利用总成本改变量除以总的成本

1 离散随机变量 ξ 的分布

情景	1	2	3	4	5	6
ξ_1^i	0	0.5	0.6	0.8	1	2
ξ_2^i	1	2	0.7	0.5	0	0.4
p_i	1/6	1/6	1/6	1/6	1/6	1/6

在表中, (m,n) 表示 P 的问题规模。例如 LB1 表示使用 P 的下界。OS 表示严格求解子问题时得到的逆优化问题的最优值

4.1 简单的例子

为说明逆优化的效果

4.2 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

5 有容量限制

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。

类似的, 有容量限制的设施选址问题定义如下:

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

我们同样可以使用行生成算法得到最优解

5.1 启发式得到上界

尽管原文章中并未提到有容量限制的设施选址逆优化问题的启发式算法, 但我们依然沿用 2010 中定义以及 2019 中提到的锥优化方法。

对于有容量限制的设施选址逆优化问题, 需要添加一个约束, 此时尽管 $\sum_j d_j u_{ij} \leq k_i v_i, i \in M$ 。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

$$\begin{aligned}
& \min \sum_{i \in M} (\tau_i^f + \eta_i^f) + \sum_{i \in M} \sum_{j \in V} (\tau_{ij}^r + \eta_{ij}^r) \\
& \text{s.t.} \quad \sum_{j \in V} \pi_j \geq \nu_{\text{UFL}}, \\
& \quad k_i t_i + \sum_{j \in V} \varrho_{ij} = \bar{f}_i, \forall i \in M, \\
& \quad \pi_j - \varrho_{ij} + \zeta_{ij} - d_j t_i = \bar{r}_{ij}, \quad \forall i \in M, \forall j \in V, \\
& \quad \sum_{i \in M} \bar{f}_i v_i^0 + \sum_{i \in M} \sum_{j \in V} \bar{r}_{ij} u_{ij}^0 = \nu_{\text{UFL}}, \\
& \quad \tau_i^f - \eta_i^f = \bar{f}_i - f_i, \forall i \in M \quad \tau_{ij}^r - \eta_{ij}^r = \bar{r}_{ij} - r_{ij}, \forall i \in M, \forall j \in V, \\
& \quad \nu_{\text{UFL1}} \leq \nu_{\text{UFL}} \leq \nu_{\text{UFL2}}, \varrho_{ij} \geq 0, \zeta_{ij} \geq 0, \forall i \in M, \forall j \in V.
\end{aligned} \tag{2}$$

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
 对于行生成的方法，实际上是限制集的选取的问题，限制集越接近
 因此可以采用

6 结论与展望

本文针对无容量限制和有容量限制的设施选址逆优化问题进行了建模分析，提出了使用行生成算法得到最优解以及通过对整数规划进行线性松弛而得到启发式方法。分析表明对于行生成算法中的子问题采取不同的计算方法可以分别得到逆优化问题的最优解以及解的上下界。同时我们借助已有的一个启发式算法说明了两者的等价关系，从而说明了启发式算法的计算复杂度。数值实验结果说明了得到的启发式算法在上界上的优异表现，而在牺牲一定计算资源的情况下，下界也可以得到较为不错的结果。这些结果充分说明了该启发式算法的有效性。实际上，尽管上界非常接近最优值，但下界的结果可以进一步优化改进。由于子问题是设施选址的原问题，可以将求解设施选址更为高效的近似算法用于其中，从而提高下界解的质量。同时，有容量限制的设施选址问题中需求和供给值的选取以及之间的关系会对逆优化问题解产生的影响也是值得思考的问题。

子问题是最具有优化空间的地方，可以结合启发式算法一次找出多个目标函数小于 0 的解，并一次性全部加入主问题中进行求解。