

设施选址逆优化问题的计算

Discount

摘要

逆优化问题

我们使用了经典的列生成算法对 UFL 的逆优化问题进行了计算，其子问题就是求解原问题 UFL，这样求解得到的就是 UFL 的逆优化最优值。当对子问题进行优化时，例如对子问题进行松弛得到非整数解这样可以得到与启发式方法一样的上界，而对子问题得到的非整数解取临近整数可以得逆问题的下界

关键词：逆优化、列生成算法、启发式算法、设施选址问题

1 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

2 设施选址逆优化

3 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， $i \in M, M = \{1, \dots, m\}$ 。 c_{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 u_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：

它的逆优化问题定义如下：

3.1 列生成算法得到最优解

3.2 减少约束得到下界

注意这部分得到的 gap 较大

3.3 启发式得到上界

在 2010 文章中，定义了关于设施选址问题的

$$\begin{aligned}
\mathbb{C}_{\text{UFL}}^x = & \left\{ (v, u) \in \mathbb{R}^{m+mv} : \sum_{i \in M} u_{ik} = 1, \forall k \in V, v_i - u_{ik} \geq 0, \forall i \in M, \forall k \in V, u_{ik} \geq 0, \forall i \in M, \forall k \in V \right\} \\
\min & \sum_{i \in M} (\tau_i^f + \eta_i^f) + \sum_{i \in M} \sum_{k \in V} (\tau_{ik}^r + \eta_{ik}^r) \\
\text{s.t.} & \sum_{k \in V} \pi_k \geq \nu_{\text{UFL}}, \\
& \sum_{k \in V} \varrho_{ik} = \bar{f}_i, \forall i \in M, \\
& \pi_k - \varrho_{ik} + \zeta_{ik} = \bar{r}_{ik}, \quad \forall i \in M, \forall k \in V, \\
& \sum_{i \in M} \bar{f}_i v_i^0 + \sum_{i \in M} \sum_{k \in V} \bar{r}_{ik} u_{ik}^0 = \nu_{\text{UFL}}, \\
& \tau_i^f - \eta_i^f = \bar{f}_i - f_i, \forall i \in M \quad \tau_{ik}^r - \eta_{ik}^r = \bar{r}_{ik} - r_{ik}, \forall i \in M, \forall k \in V, \\
& \nu_{\text{UFL}1} \leq \nu_{\text{UFL}} \leq \nu_{\text{UFL}2}, \varrho_{ik} \geq 0, \zeta_{ik} \geq 0, \forall i \in M, \forall k \in V.
\end{aligned} \tag{1}$$

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。 ν_{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，考虑在原问题上改进得到更好的上界

4 有容量限制

4.1 列生成算法得到最优解

4.2 减少约束得到下界

4.3 启发式得到上界

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