Numerical Optimization, 2020 Fall

Homework 6

Due on 23:59, Nov. 06, 2020

(NOTE: Homework will not be accepted after this due for any reason.)

1 Warehousees for MX

某西 (MX) 是一个美国邮寄物品租赁公司. 顾客在该公司的网站上租赁物品, 然后 MX 会将它邮寄给顾客. 当顾客完成使用后, 需要将其邮寄回 MX. MX 的商业模式依赖于快速的发货时间 (否则, 顾客没有耐心等待很长时间). 但是, 像某东 (MD) 等公司的半日达速递服务不在我们的问题讨论范围之内, 因为他们的邮寄成本可能会相对较高.

在早期, MX 只有几个配送中心 (Distriution Centers, DC for short), 并且一般需要 2 天的时间才能把物品邮寄到顾客手中. MX 很快意识到提供快速的送货服务对于公司业务提升至关重要, 然后他们决定尝试为大约 90% 的客户提供 1 日达的速递服务. 这项政策在很大程度上导致了几年前利润的大幅增长.

在这道问题中, 您将进行数学建模并求解, 来确定 MX 应该把 DC 放置在何处, 以确保所需的顾客在 1 天邮寄范围内, 同时将建立 DC 的固定成本降至最低. (假定每个 DC 的物品处理和运输的单位成本都是相同的).

a) 将以下问题建模为整数规划 (Integer Programming, IP for short) 问题: 给定一组城市, 以及每个城市的人口和在该城市开设 DC 的固定成本. 目标是决定将 DC 设在哪些城市, 以便将总的固定成本降至最低, 同时还确保至少有 α 人口在 1 天邮寄范围内.

注意: 确保清晰地定义您所使用的符号, 并指明哪些是参数 (输入), 哪些是决策变量. 然后, 使用文字描述清楚您的每一个约束条件. [20pts]

b) 使用 AMPL 实现您创立的模型. 请查看.zip 文件 warehouse.xlsx 中的数据集来解决这个问题. 该数据集提供了全美 150 个大城市的位置和人口数 (根据 2000 年的美国人口普查), 以及在相应城市开设一家 DC 的年均固定成本. 文件中还包含数据集中每对城市之间的距离 (以英里为单位). 若两个城市之间的距离不超过 150 英里, 即可认定为它们在 1 天的邮寄范围内.

另外, 覆盖率 $\alpha = 90\%$, 请找到 MX DC 位置问题的最佳解决方案. 在您提交的文件中须包括两部分: PDF 文件和 .zip 文件 (请您分离提交). 其中模型文件 (.mod), 数据文件 (.dat), 运行文件 (.run) 以及报告您的解决方案的总成本和要开放的 DC 总数: 用截图的方式提交在 PDF 中; 所提交的 .zip 文件中包含您编写的.mod, .dat, .run 等文件. [25pts]

c) MX 目前大约有 35 个 DC 开放. 这是否接近您在 b) 中找到的 DC 数量? 如果不是, 您的解决方案与 MX 的解决方案可能不同的原因是什么? [5pts]

1. 构建下述整数规划模型：

    - Sets
        - I: 城市集合。
        - J: DC 位置的集合。
    - Parameters:
        - p_i: 城市 i 的人口。
        - d_{ij}: 城市 i 和城市 j 之间的距离是否在 150 英里内。
        - c_i: 在城市 i 建立 DC 的固定成本。
        - α: 计划达到的覆盖率。
    - Decision variables:
        - x_i: 是否在城市 i 处开设 DC。
        - g_i: 城市 i 是否被覆盖到。

    优化模型为：

$$
\begin{aligned}
\min_{x,g} \quad & \sum_{i=1}^{n} c_i x_i \\
s.t. \quad & \sum_{i=1}^{n} g_i p_i \geq \alpha \sum_{i=1}^{n} p_i \quad (\text{覆盖率达到 } \alpha) \\
& g_i \leq \sum_{j=1}^{n} x_j d_{ij}, \ \forall i \\
& g_i \geq \frac{\sum_{j=1}^{n} x_j d_{ij}}{1000}, \ \forall i \quad (\text{这两个约束使得 } g_i = \min\{1, \sum_{j=1}^{n} x_j d_{ij}\}) \\
& x_i \in \{0,1\}, g_i \in \{0,1\} \quad (\text{整数约束})
\end{aligned} \tag{1}
$$

2. 由求解器解得：最优的开设的 DC 数量为 23，最优的 cost 为 2970000。

```
# Define set: begin city and destination
set begin;
set dest;
# Define parameters: cost, population, distance<=150
param c {dest};
param p {begin};
param d {begin, dest};
param alpha;
# Define decision variables: DC_location, being_covered_or_not
var x{dest} binary;
var g{begin} binary;

# objective
minimize COST:
    sum {i in dest} c[i] * x[i];
# constraint 1: alpha cover rate
subject to COVER_RATE:
    (sum {i in begin} g[i] * p[i]) >= (alpha * sum {j in begin} p[j]);
# constraint 2 & 3: Switching constraint
subject to SWITCH_1 {i in begin}:
    g[i] <= sum {j in dest} x[j] * d[i, j];
subject to SWITCH_2 {i in begin}:
    g[i] >= sum {j in dest} x[j] * d[i, j] / 1000;
```

图 1: .mod

```
data;
set begin := city1  city2   city3   city4   city5   city6   city7   city8   city9   city10  city11  city12  city13  city14  city15  city16
set dest := dest1   dest2   dest3   dest4   dest5   dest6   dest7   dest8   dest9   dest10  dest11  dest12  dest13  dest14  dest15  dest16

param alpha:= 0.9;

param p:=
city1   8103677
city2   3805387
city3   2934854
city4   2028533
city5   1522870
city6   1395817
city7   1260010
city8   1234150
city9   1188531
city10  960279
city11  921724
city12  804020
city13  799977
city14  761895
city15  714871
city16  681734
city17  665829
city18  663740
city19  605315
city20  597674
city21  585273
city22  583680
city23  583008
city24  578298
city25  577817
city26  559884
city27  555194
city28  542334
city29  517995
city30  514247
city31  513913
city32  488633
city33  480690
city34  475333
city35  455452
city36  448042
city37  440449
city38  437232
city39  435548
city40  419198
city41  418810
city42  411438
city43  399091
city44  394285
city45  392054
city46  379351
city47  375418
city48  353313
city49  348770
city50  348091
city51  345735
city52  337830
city53  335736
city54  332869
city55  315118
city56  314287
city57  296134
city58  294233
```

图 2: .dat

```
reset;
model /Users/taou/Documents/Learning/Course/数值最优化/Homework6/hw6.mod;
data /Users/taou/Documents/Learning/Course/数值最优化/Homework6/hw6.dat;
option solver gurobi;
solve;
display sum {i in dest} x[i];
display sum {i in begin} g[i] * p[i] /  sum {j in begin} p[j];
display COST;
```

图 3: .run

```
ampl: reset;
ampl: include '/Users/taou/Documents/Learning/Course/数值最优化/Homework6/hw6.run';
Gurobi 9.0.3: optimal solution; objective 2970000
71 simplex iterations
1 branch-and-cut nodes
sum{i in dest} x[i] = 23

sum{i in begin} g[i]*p[i]/(sum{j in begin} p[j]) = 0.900764

COST = 2970000
```

图 4: result

3. 不接近。这个现象是非常正常的，因为在实际的建模过程中，我们的的目标函数与约束都会更加复杂，例如目标函数中我们还需要考虑运输成本，在约束中会加入 DC 容量的限制，等等。因此 DC 的数量和最低成本在直观上都会增加。

2 Solving the MX Problem

在上述问题 **1** 中, 您制定了一个 IP 模型来解决 MX 的 DC 定点问题, 以确保给定的人口分数 (α) 在距离其最近的 DC 的 1 天邮寄范围内. 在此问题中, 您将开发一种基于 Lagrangian 松弛来求解此 IP 的方法.

作为出发点, 您要根据您的 IP 模型松弛某个约束, 并定义一个相应的乘子.

a) 写出这个松弛产生的 Lagrangian 子问题. [20pts]

b) 子问题应分解为两个分离的问题, 一个仅包含 $\boldsymbol{x}$ 变量, 另一个仅包含 $\boldsymbol{y}$ 变量. 写出这两个不同的问题. [20pts]

c) 请解释如何求解两个子问题, 即 $\boldsymbol{x}$-子问题和 $\boldsymbol{y}$-子问题. 您的求解方法可能不依赖于使用单纯型法或任何其他一般的 LP 或 IP 算法. [10pts]

1. 引入拉格朗日乘子 $\lambda_i, \mu_i \geq 0$.

$$\begin{aligned}
\min_{x,g} \quad & \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \lambda_i (g_i - \sum_{j=1}^{n} x_j d_{ij}) + \sum_{i=1}^{n} \mu_i (\frac{\sum_{j=1}^{n} x_j d_{ij}}{1000} - g_i) \\
s.t. \quad & \sum_{i=1}^{n} g_i p_i \geq \alpha \sum_{i=1}^{n} p_i \quad (\text{覆盖率达到 } \alpha) \\
& x_i \in \{0,1\}, g_i \in \{0,1\} \quad (\text{整数约束}) \\
& \lambda, \mu \geq 0
\end{aligned} \tag{2}$$

2. 将原问题分解为包含 x 和 g 的子问题:

- x-子问题

$$\begin{aligned}
\min_{x} \quad & \sum_{i=1}^{n} c_i x_i - \sum_{i=1}^{n} (\lambda_i - \frac{\mu_i}{1000}) \sum_{j=1}^{n} x_j d_{ij} \\
s.t. \quad & x_i \in \{0,1\}, \lambda, \mu \geq 0
\end{aligned} \tag{3}$$

  该形式可进一步化简为:

$$\begin{aligned}
\min_{x} \quad & \sum_{i=1}^{n} \left(c_i - \sum_{j=1}^{n} (\lambda_j - \frac{\mu_j}{1000}) d_{ji} \right) x_i \\
s.t. \quad & x_i \in \{0,1\}, \lambda, \mu \geq 0
\end{aligned} \tag{4}$$

- s-子问题

$$\begin{aligned}
\min_{x} \quad & \sum_{i=1}^{n} (\lambda_i - \frac{\mu_i}{1000}) g_i \\
s.t. \quad & \sum_{i=1}^{n} g_i p_i \geq \alpha \sum_{i=1}^{n} p_i \quad (\text{覆盖率达到 } \alpha) \\
& g_i \in \{0,1\}, \lambda, \mu \geq 0
\end{aligned} \tag{5}$$

3.
- 对于 x-子问题:
  (a) 若 $c_i - \sum_{j=1}^{n} (\lambda_j - \frac{\mu_j}{1000}) d_{ji} < 0$, 则令 $x_i = 1$。
  (b) 若 $c_i - \sum_{j=1}^{n} (\lambda_j - \frac{\mu_j}{1000}) d_{ji} \geq 0$, 则令 $x_i = 0$。
- 对于 s-子问题:

(a) 若 $\lambda_i - \frac{\mu_i}{1000} < 0$，则令 $g_i = 1$，否则 $g_i = 0$。

(b) 如果得到的 x 满足约束，则该 x 即为最优解。

(c) 如果得到的 x 不满足约束，那么就依据 $\lambda_i - \frac{\mu_i}{1000}, \forall g_i = 0$ 的大小，从小到大依次将 $g_i = 0 \rightarrow g_i = 1$，直到约束满足为止。最终的 x' 即为最优解。