

情報特別演習 最終報告書

航空ネットワークのハブ配置最適化

筑波大学情報学群情報科学類 2 年 岡部 純弥

2022 年 1 月 27 日

概要

本演習を通して、航空ネットワークのハブ配置問題に関する一連の研究の流れを把握し、非ハブ同士で接続可能な新たなモデルを提案した。その後、このモデルを最適化問題として定式化し最適化ソルバー Gurobi Optimizer を用いて最適解を導出した。モデルの定式化にあたっては、Campbell[3] が定式化した p-hub median モデルをベースし、さらに特定の条件下で非ハブの空港同士の接続を可能とした。

1 はじめに

1.1 本演習への動機

以前旅行で航空機を使用した際、目的地への直行便が飛んでいなかったため、トランジットを余儀なくされた経験があった。また直行便よりトランジットをした方が時間こそかかるものの、安く目的地まで辿り着けることも多々ある。しかし主要都市へ行く際、多くの場合は直行便を利用することになる。

このように直行便を開設すべきか否か、またどのようにして実際の航空ネットワークが形成されるのかを確かめたいと考えた。さらに、実際の航空ネットワークをコストや時間の観点からより良いものに改善することはできないかと思い、本演習を行うことにした。

1.2 航空ネットワークとは

世界中には数多くの空港が存在し、空港間に路線が敷かれている。空港をノード、路線をエッジとした際に、航空網を有向グラフと見做すことができる。以降この有向グラフを航空ネットワークと呼ぶことにする。簡単な航空ネットワークの例として、Peach Aviation^{*1} の就航路線網を図 1 に記載する。



Fig.1: Peach Aviation の航空ネットワーク
(Peach Aviation 公式 HP より引用)

1.3 ハブ空港とは

航空ネットワークにおいて、多くの路線が就航する空港のことはハブ空港という言葉で知られている。そしてハブ空港を中心に航空ネットワークが形成される模様は、ハブ・アンド・スポークと呼ばれている。近年では、ハブ・アンド・スポーク方式の輸送形態からポイント・トゥ・ポイント方式の輸送形態へと変化しているが、依然としてシンガポールのチャンギ国際空港、韓国の仁川国際空港、さらには日本の羽田空港などのようなハブ空港も多く存在する。

^{*1} <https://www.flypeach.com>

1.4 ハブ配置問題とは

ハブ空港に関して、航空ネットワークに複数のハブ空港を配置し、航空ネットワーク全体における総輸送費用を最小化する問題は一般に**ハブ配置問題** (Hub Location Problem) と呼ばれている。

そもそもハブ配置問題は、O’Kelly[6] がハブ空港を配置するモデルを考案したことから始まる。O’Kelly は、ネットワーク上に1つのハブ空港を配置することで、ネットワーク全体の総輸送費用を最小化する問題を定式化した。その翌年には O’Kelly[7] がネットワーク上に複数のハブを設置するモデルを定式化し、合わせてヒューリスティックな解法を提案した。その後、Campbell[3] が目的関数を線形緩和したモデルを提案した。

これらのハブ配置問題の求解に関して、 n 個のハブ空港候補から P 個のハブ空港を選択する組合せは ${}_nC_P$ 通り、さらにハブでない $n - P$ 個の空港が P 個のハブ空港に接続するか否かの総数は $(n - P)^{2^P}$ 通りである。すなわち、 n や P の値が大きくなったとき、直ちに全列挙によって解を求められなくなる。したがってハブ配置問題は一般的に、ヒューリスティックな解法や近似解法を用いて良い上界を求めることが主流とされていた。この流れにあわせて、より強い仮定を用いることで厳密解を求める研究も行われた。

このように、ハブ配置問題は O’Kelly の考案を基に、1990 年代から 2000 年代にかけて大きく発展した。ハブ配置問題に対する一連の研究の流れは、やや古いものになるが O’Kelly ら [4]、Alumur ら [2] や佐々木 [11] など書かれている。

1.5 本演習の目的

本演習では、ハブ配置問題の既存のモデルを拡張し、数理最適化ソルバー Gurobi Optimizer ^{*2} を用いて、その厳密解を導出した。概要で述べた通り既存のモデルとして、Campbell[3] が定式化した P-Hub-Median モデルを選択した。

2 モデルの作成

2.1 既存のモデル

ハブ配置問題では一般に、

1. ノード数が n の航空ネットワークを表す有向グラフ $G = (V, E)$ ^{*3}
2. 全ての OD ^{*4} 間のフロー
3. 各出発地点から各到着地点への輸送コスト、及びハブ空港間のディスカウントファクター α

の全てが与えられる。ここで、ディスカウントファクター α とは O’Kelly[7] によれば、

These inter-hub costs are multiplied by a parameter $\alpha \leq 1$ to reflect the scale effects in inter-facility flows. The idea is that the flows between hubs might enjoy a discounted transportation rate arising from the greater volume on these connections.

と書かれている。すなわち、ハブ空港間の路線のように大量の需要がある場合、スケーリングエコノミーによって輸送コストが減少する。この割合を α と定義している。ただし、後に O’Kelly[8] で、ディスカウントファクターは定数ではなく需要に依存して決定すると主張がされているが、簡単のため本演習ではディスカウントファクター α は定数とした。

本演習で実装したモデルは、Campbell[3] が定式化した P-Hub-Median モデル (HMP) の拡張である。HMP は O’Kelly[7] が提唱した single allocation P-Hub-Median モデル (HMP-S) の目的関数を線形にしたモデルで、^{*5} 目的関数が線形であるがゆえに線形最適化問題として解くことができる。

まず、HMP-S は次のように定式化される。

^{*3} すなわち、各ノードが実際の空港に対応する

^{*4} 出発地-目的地

^{*5} O’Kelly[7] のモデルは目的関数に二次の項が含まれていた。

^{*2} <https://www.gurobi.com/products/gurobi-optimizer/>

(HMP-S)

$$\min \sum_i \sum_j \sum_k \sum_m h_{ij} (Z_{ik} C_{ik} + Z_{jm} C_{jm} + \alpha Z_{ik} Z_{jm} C_{km})$$

s.t.

$$\begin{aligned} \sum_k x_k &= P \\ \forall k; \sum_k Z_{ik} &= 1 \\ \forall i, k; 0 &\leq Z_{ik} \leq x_k \\ \forall i, k; 0 &\leq Z_{ik} \leq 1 \\ \forall k; x_k &= 0, 1 \end{aligned}$$

ただし,

- P : ハブ空港の数
- $h_{ij} :=$ ノード i からノード j への需要量
- $C_{ij}^{km} :=$ ハブノード k, m を経由するときのノード i からノード j へのコスト*6
- $Z_{ij} :=$ ノード i からノード j へのフロー
- $x_k :=$ ノード k がハブ空港であれば 1, そうでなければ 0

とする。HMP-S は目的関数に二次の変数項が含まれているため、問題のサイズが大きくなったときに全探索による求解が難しくなってしまう。この問題を解決するために、目的関数を線形緩和した HMP が提案された。HMP は次のように定式化される。

(HMP)

$$\min \sum_i \sum_j \sum_k \sum_m C_{ij}^{km} h_{ij} Z_{ij}^{km}$$

s.t.

$$\begin{aligned} \sum_k x_k &= P \\ \forall i, j; \sum_k \sum_m Z_{ij}^{km} &= 1 \\ \forall i, j, k, m; Z_{ij}^{km} &\leq x_k \\ \forall i, j, k, m; Z_{ij}^{km} &\leq x_k \\ \forall i, j, k, m; Z_{ij}^{km} &\geq 0 \\ \forall k; x_k &= 0, 1 \end{aligned}$$

*6 ノード A からノード B へのコストを C_{AB} と表せば,
 $C_{ij}^{km} = C_{ik} + \alpha C_{km} + C_{mj}$

ただし, Z_{ij}^{km} をハブノード k, m を経由するときのノード i からノード j へのフローを表すものとし, その他の変数は HMP-S と同じとする。

2.2 提案するモデル

本演習で提案するモデルは, HMP を拡張し条件付きでハブでない空港同士での接続を可能にしたモデルである。この条件とは

1. ハブでない 2 つの空港間の需要がしきい値 q 以上である
2. いかなるハブを経由するよりも, 直行便を開設する方がコストが安くなる

の 2 つが成り立つこととした。すなわち, ハブでない空港 i とハブでない空港 j が接続するには,

1. $h_{ij} \geq q$
2. $\forall k \forall m; C_{ij} \leq C_{ij}^{km}$

の 2 つが成り立てばよい。さらに, 空港 i と空港 j が直接接続されているとき, 明らかにハブを経由しない方が総費用が安くなる。

2.3 最適化問題の作成

2.2 節で提案した条件, 及び HMP をもとに拡張したモデルを定式化する。ただし, 変数の定義は 2.1 節で述べた通りとする。

(提案するモデル)

$$\min \sum_i \sum_j \left(\sum_k \sum_m C_{ij}^{km} h_{ij} Z_{ij}^{km} + C_{ij} h_{ij} Z_{ij} \right)$$

s.t.

$$\begin{aligned} \sum_k x_k &= P \quad (1) \\ \forall i, j; \sum_k \sum_m Z_{ij}^{km} + Z_{ij} &= 1 \quad (2) \\ \forall i, j, k, m; Z_{ij}^{km} &\leq x_k \quad (3) \\ \forall i, j, k, m; Z_{ij}^{km} &\leq x_k \quad (4) \\ \forall i, j, k, m; Z_{ij}^{km} &\geq 0 \quad (5) \\ \forall k; x_k &= 0, 1 \quad (6) \end{aligned}$$

目的関数の一項目*⁷ で、ハブを経由する総費用を、二項目*⁸ で直行便の総費用を表している。制約条件に関して、(1) 式でハブの総数が P である制約、また、制約条件の (2) 式によって、各 OD ペアについて、ハブを経由するフローと直行便のフローの総和が 1 になるようになっている。さらに (3) 式から (5) 式で、 Z_{ij}^{km} に関する整合性制約を記述している。(6) 式では x_k がバイナリ*⁹変数であることを表している。

3 最適化問題の解析

3.1 データセット

本演習では、この最適化問題に対して乱数を用いたデータを作成し、そのデータセットを用いて最適化問題を解く。ただし、 C_{ij} は対称行列とし、 h_{ij} は対称行列ではないように作成する。¹⁰ C_{ij} のデータセットを乱数によって作成するプログラムを Fig.2 に記載する。¹¹ また、 h_{ij} は制約がないため、簡単

```
def init_list(n):
    res = [[0 for _ in range(n)] for _ in range(n)]
    for index in range(n):
        for col in range(n):
            if index > col:
                continue
            tmp = randint(1, 10)
            res[index][col] = tmp
            res[col][index] = tmp
    return res
```

Fig.2: 乱数を用いて最適化問題のデータセットを作成するプログラム

な二重ループによって容易に作成ができる。本演習では、 C_{ij} は 1 から 10 までの一様乱数、 h_{ij} は 1 から 10 までの一様乱数の値に 10 をかけたもの¹²を

*⁷ $\sum_k \sum_m C_{ij}^{km} h_{ij} Z_{ij}^{km}$

*⁸ $C_{ij} h_{ij} Z_{ij}$

*⁹ 0-1

*¹⁰ 実際問題としては、空港 A から空港 B へのコストと空港 B から空港 A へのコストは異なることが多いが、簡単のため対称行列を仮定した。この仮定はなくても求解への影響はない。

*¹¹ ただし、Fig.2 で返される二次元配列を l とすると、任意の i, j に対して $l[i][j] = l[j][i]$ である。

*¹² すなわち、 h_{ij} は 10, 20...100 のいずれかの値を取る。

用いた。

概要で述べた通り、この最適化問題の厳密解の導出のために、数値最適化ソルバー Gurobi Optimizer を用いた。したがって、ここではヒューリスティックな解法の提案や実装は行わない。

3.2 実装

実装したコードのうち、最適化問題を記述する部分のみを抜き出し、Fig.3 に記載する。Fig.3 では制

```
# --- Constraint ---
# Constraint that the number of hubs will be three.
model.addConstr(sum(x) == 3)

# Constraint that the sum of the flows is 1 for each i, j
for i in range(node_num):
    for j in range(node_num):
        tmp = 0
        if z_ij_index[i][j]:
            tmp = 1
        model.addConstr(sum(z_ijkm[i][j][k][m]
                             for k in range(node_num)
                             for m in range(node_num))
                        + tmp == 1)

# --- Objective Function ---
model.setObjective(
    grb.quicksum((solve_c_ijkm(c, alpha, i, j, k, m) * h[i][j] * z_ijkm[i][j][k][m]
                    for i in range(node_num)
                    for j in range(node_num)
                    for k in range(node_num)
                    for m in range(node_num))) +
    grb.quicksum(c[i][j] * h[i][j] if z_ij_index[i][j] else 0
                  for i in range(node_num)
                  for j in range(node_num)),
    grb.GRB.MINIMIZE
)

# --- Perform Optimization ---
model.optimize()
```

Fig.3: 最適化問題を記述する箇所の抜粋

約条件の記述、及び目的関数の設定を行い、実際に最適化問題を実行している。コード全体は GitHub 上¹³に掲載したので、こちらを参照されたい。ただし、コードの実行には、最適化ソルバー Gurobi Optimizer のライセンスおよび Python ライブラリの NetworkX ¹⁴ が必要である。

*¹³ User:Okabe-Junya, Repo:Special-Seminar.

あるいは https://github.com/Okabe-Junya/Special-Seminar/tree/main/Implementation/my_def からアクセスできる。

*¹⁴ <https://networkx.org>

3.3 実行結果

3.2 節で実装したコードを実行し、得られた解から航空ネットワークを構成する。ただし $P = 3$, ディスカウントファクター α を 0.6, しきい値 q を 60 として実行した。構成したネットワークを Fig.4 に記載する。Fig.4 に関して簡潔に補足する。赤色で出

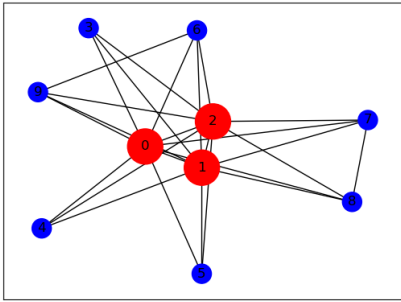


Fig.4: 最適化問題の最適解の可視化 ($\alpha = 0.6, q = 60$)

力されたノードがハブ空港、青色で出力されたノードがハブでない空港である。また、エッジによって空港間の接続（路線）を表している。Fig.4 を確認すると、確かに非ハブの空港同士での接続が行われつつも、ネットワークが出力されていることが確認できる。

3.4 考察

3.3 節の実行結果に基づいて考察を行う。まず、ディスカウントファクター α の値を変更したときの、ネットワークの構成の変化を確認する。このときのしきい値の値は 60 で固定している。例として、ディスカウントファクター α を 0.3 にしたときのネットワークを Fig.5 に記載する。 $\alpha = 0.3$ のときは $\alpha = 0.6$ のときと比べてハブ間の割引が大きいため、よりハブ間のフローが大きくなる。すなわち、非ハブ間の接続が減るはずである。Fig.5 を見ると確かに Fig.4 より非ハブ間の接続が少ないことが確認できる。

次に、しきい値 q の値を変更したときの、ネットワークの構成の変化を確認する。このときのディスカウントファクター α の値は 0.6 で固定している。

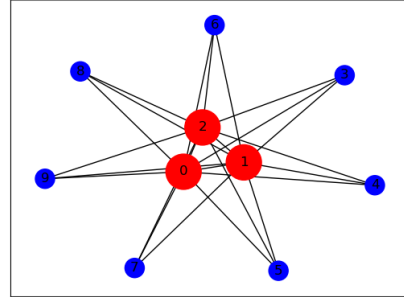


Fig.5: 最適化問題の最適解の可視化 ($\alpha = 0.3, q = 60$)

例として、しきい値を 30 にしたときのネットワークを Fig.6 に記載する。しきい値が小さくなると、非ハブの空港同士での接続が行いやすくなるため、ネットワークのエッジが増えることが予測される。Fig.6 を見ると、Fig.4 と比べて、確かに非ハブの

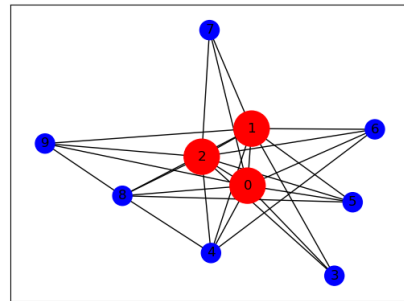


Fig.6: 最適化問題の最適解の可視化 ($\alpha = 0.6, q = 30$)

空港同士での接続が多くなっていることが確認できる。しきい値を変更したときの、非ハブの空港同士での接続の本数を可視化する。しきい値を 10 から 90 まで 10 刻みで変更したときの、非ハブの空港同士での接続の本数の変化を Fig.7 に記載する。Fig.7 を見ると、しきい値が大きくなると、非ハブの空港同士での接続が少なくなることが確認できる。線形回帰によって Fig.7 を近似することを考える。非ハブの空港同士での接続の本数の総和を m とすると

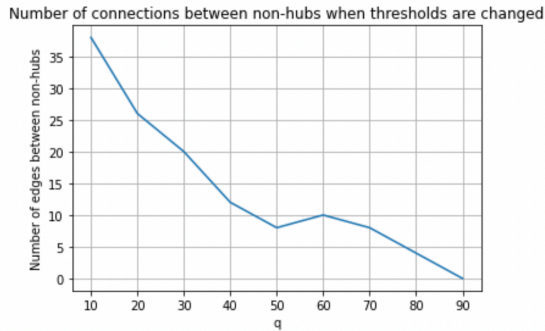


Fig.7: しきい値を変更した際の非ハブの空港同士の接続の本数

$m = -0.41q + 34.3$ という回帰直線が求められる。空港の数が 10, ハブの数が 3 のとき, 路線の総和は $m + 24$ であるため, m の値を調整することで, 総路線数を適切に調整することが可能である。

4 まとめ

本演習を通して, ハブ配置最適化問題について既存の主要な手法を調査し, その拡張を行った。具体的には, 条件付きで非ハブの空港同士の接続を可能にしたモデルを提案し, 最適化ソルバーを用いてその求解を行った。このモデルは, 非ハブ同士での接続が可能であるため, 現実の航空ネットワークにより近いネットワークとなっていることが確認できた。

5 今後の課題

本演習を通して, 既存のモデルの拡張およびその厳密解の導出を行ったが, このモデルの実データへの応用を行うことができなかった。実データへの応用を行う際には, 問題サイズがより大きくなることが考えられるので, コードをより高速にする必要がある。例えば今回の実装では, Python の繰り返し構造 (for ループ) を用いて実装してしまっている箇所があるため, 該当箇所を C++ や Cython で再実装する, あるいは Numpy のようなライブラリを用いて高速化することが考えられる。

また, 乗り換え回数や遅延などの要素を取り入れ

たモデルの提案も行いたい。さらに, Marianov ら [5], Wagner[9] などに挙げられる 2 つ以上のアライアンスがあり, ゲーム理論の要素を加味した競合モデルの提案なども今後の課題である。

6 謝辞

本演習のアドバイザー教員を担当していただきました長谷部 浩二先生をはじめご協力いただきました方々へ, この場を借りて感謝申し上げます。

参考文献

- [1] Python で学ぶネットワーク分析 Colaboratory と NetworkX を使った実践入門. オーム社, 2019.
- [2] Sibel Alumur and Bahar Y Kara. Network hub location problems: The state of the art. *European journal of operational research*, Vol. 190, No. 1, pp. 1–21, 2008.
- [3] James Campbell. Hub location and the p-hub median problem. *Operations Research*, Vol. 44, pp. 923–935, 12 1996.
- [4] James F Campbell and Morton E O’Kelly. Twenty-five years of hub location research. *Transportation Science*, Vol. 46, No. 2, pp. 153–169, 2012.
- [5] Vladimir Marianov, Daniel Serra, and Charles ReVelle. Location of hubs in a competitive environment. *European journal of operational research*, Vol. 114, No. 2, pp. 363–371, 1999.
- [6] Morton O’Kelly. The location of interacting hub facilities. *Transportation Science*, Vol. 20, pp. 92–106, 05 1986.
- [7] Morton O’Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, Vol. 32, pp. 393–404, 12 1987.
- [8] Morton E O’Kelly and DL Bryan. Hub location with flow economies of scale. *Transportation Research Part B: Methodological*,

Vol. 32, No. 8, pp. 605–616, 1998.

- [9] Bernd Wagner. A note on “location of hubs in a competitive environment”. *European Journal of Operational Research*, Vol. 184, No. 1, pp. 57–62, 2008.
- [10] 幹雄久保, ペドロペドロソジョア, 正和村松, アブドルレイス. あたらしい数理最適化: Python 言語と Gurobi で解く. 近代科学社, 2012.
- [11] 佐々木美裕. ハブ空港の配置モデル. オペレーションズ・リサーチ: 経営の科学=Operations research as a management science [Research], Vol. 45, No. 9, pp. 437–443, 2000.
- [12] 誉人久野, 麻衣子繁野, 順哉後藤. IT Text 数理最適化. IT Text / 情報処理学会編集. オーム社, 2012.

付録 A 使用書籍

本演習を通して、学習のために用いた書籍を以下に記載する。

- IT Text 数理最適化 [12]
- あたらしい数理最適化 [10]
- Python で学ぶネットワーク分析 [1]

付録 B 実行環境

本演習にて、プログラムを実行した環境は

- macOS Monterey 12.1
- Python 3.9.7
- Gurobi Optimizer 9.5
- gurobipy 9.5.0
- matplotlib 3.5.1
- NetworkX 2.3

となっている。上記環境にてプログラムが正しく動作することを確認している。