**MÔ TẢ FLEET ASSIGNMENT MODEL**

1. **Objectives (đề bài đưa cho model giải quyết)**

Maximize lợi nhuận VTHK =

+ Tổng doanh thu VTHK của từng chuyến bay được lựa chọn

- Tổng chi phí không bị ảnh hưởng bởi cơ cấu mạng bay của từng chuyến bay được lựa chọn

- chi phí SCBD (tính dựa trên FH/Cycle và FH)

- chi phí tàu (tính dựa trên số tàu lớn nhất cần sử dụng trong các khung thời gian x chi phí thuê mua tàu)

- các chi phí cố định khác (dự toán, ước tính)

Assign\_fleet\*BH\*CP đơn vị BH + Assign\_fleet\*FH\*CP đơn vị FH + Assign\_fleet\*CP đơn vị chuyến bay

1. **Parameters (các số liệu đầu vào giúp model giải quyết objectives)**

* Doanh thu VTHK/chuyến (tính chi tiết từ khách, hàng hoá, khác) của sector theo loại tàu bay (giả định DTTB và LF ko bị ảnh hưởng bởi khung tgian và ngày trong tuần, chỉ bị ảnh hưởng bởi loại tàu bay).
* Các chi phí biến đổi không bị ảnh hưởng bởi cơ cấu mạng bay của từng chuyến bay: nhiên liệu (tính từ định mức tiêu hao nhiên liệu các dòng tàu bay), PVCB, PVHK, nhân công biến đổi, chi phí bán.
* CP SCBD theo bảng matrix FH/Cycle của từng đội tàu bay.
* CP tàu bay theo từng dòng tàu bay. Bảng chi tiết số tàu bay của VNA năm 2019.
* Các chi phí cố định còn lại: chi phí quản lý chung, nhân công cố định, các chi phí cố định khác.
* Các thông số khai thác: BH theo từng chuyến bay – loại tàu bay, minimum groundtime của từng sân bay – loại tàu bay, distance của từng chuyến bay.
* Dataframe thông tin chuyến bay như BH, FH, seat, LF, pax/chuyến, số tần suất 1 chiều/tuần, ETD, ETA (đánh giá ETD và ETA nên được thể hiện dưới dạng timestamp có đầy đủ giờ, ngày (theo lịch bay 1 tuần cụ thể bất kỳ) để tiện cho các bước tiếp theo, có thể tính ETA dựa trên ETD và pd.to\_timedelta) theo lịch bay 2019.

Timetable (có thể bổ sung thêm một số option với revenue khác nhau đối với schedule design optimization)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | Sector | From | To | Số hiệu | DOW | AC | ETD | ETA |
| 0011350 | HAN-SGN | HAN | SGN | 001 | 1 | 350 | 4,25 | 6,5 |
| 0012321 | SGN-HAN | HAN | SGN | 001 | 2 | 321 | 5,5 | 7,5 |
|  | ….. |  |  |  |  |  |  |  |

Balance constraint: điều kiện trong khoảng time node T đến T+1 thì có những ETA nào => lọc dataframe dựa trên những ETA đó + “to” == airport, sau đó sum lại decision variable dựa trên index có trong filtered dataframes.

1. **Index sets (mức độ chi tiết ảnh hưởng đến số variables và constraints)**

* Sector (HAN-SGN, SGN-HAN,…).
* Loại tàu bay (321, 320, 78A, 78B, 78C, 350, AT7).
* Ngày trong tuần (1234567).
* Giờ+phút (0-23 giờ).

1. **Variables (các biến số mà model được phép thay đổi nhằm đạt được objectives)**

* Assign aircraft: Bay = 1, không bay = 0. Mẫu output. Assign aircraft dùng để tính số tàu bay đến/bay đi từ từng sân bay trong các khung giờ. Ngoài ra, assign aircraft cũng dùng để tính doanh thu, chi phí trong objective function.

|  |  |
| --- | --- |
| Index | Decision variable |
| 0011350 | 1 |
| 0012321 | 0 |
| ….. | …… |

|  |  |  |  |
| --- | --- | --- | --- |
| Số hiệu | DOW | AC | Decision variable |
| 001 | 1 | 321 | 1 |
| 001 | 1 | 320 | 0 |
|  |  |  |  |

Model.assign\_fleet = Var(flight no., dow, ac\_type, within = Binary, initialize = lịch bay hiện tại)

* Số lượng tàu bay đậu đỗ theo từng time node tại từng sân bay. Mẫu output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sân bay | DOW | hour | AC | Số tàu |
| HAN | 4 | 6 | 321 | 15 |
| SGN | 4 | 6 | 321 | 23 |
| ……. |  |  |  |  |

Model.park\_ac = Var(airport, dow, hour, fleet, domain = NonNegativeIntegers)

Có cần thiết phải đặt park\_ac là variable ko? Hay có thể để dưới dạng mutable parameter (ko để dưới dạng mutable parameter được: https://stackoverflow.com/questions/61997012/pyomo-coupling-variables)

1. **Constraints (các giới hạn đặt ra mà model phải tuân theo)**

* Balance constraint: tại mỗi mốc thời gian, số tàu đậu đỗ theo từng loại tàu của mỗi mốc thời gian + số tàu bay hạ cánh phải tương đương với số tàu đậu đỗ của mốc thời gian kế tiếp + số tàu bay cất cánh (Node1 + inwards = Node2 + outwards). Ngoài ra, để đảm bảo giả định sản phẩm tần suất các tuần giống nhau, time node cuối cùng trong tuần (23:45 Chủ Nhật) phải balance với node đầu tiên trong tuần (0:15 Thứ Hai).

Constraints để thể hiện LHS (left-hand side) và RHS (right-hand side) nhằm đảm bảo balance constraint. Tạo indexed constraints:

|  |  |
| --- | --- |
| Left-hand side | Right-hand side |
| Tàu đang đậu đỗ tại khung giờ T  +  Tàu bay đến sân bay (inwards) tại khung giờ T | Tàu đang đậu đỗ tại khung giờ T+1  +  Tàu bay đi từ sân bay (outwards) tại khung giờ T |

Model.balance\_constraint = Constraint(airport, dow, hour, ac\_type, rule =)

Def rule: for each fleet type in each airport:

Balance constraint cho từng sân bay theo các khung giờ trong ngày:

for a in airport\_param:

for ac in ac\_type:

for DOW in range(1,8):

for hour in range(0,24):

Model.park\_ac[airport == a, DOW== d, hour == T]

== model.park\_ac[hour==T-1]

+ sum assign\_fleet if timetable[timetable[“To”]==a][ETA] > assign\_fleet[hour] and timetable[timetable[“To”]==a][ETA] < assign\_fleet[hour+1]

- sum assign\_fleet if timetable[timetable[“From”]==a][ETD] > assign\_fleet[hour] and timetable[timetable[“From”]==a][ETD] < assign\_fleet[hour+1]

(có thể trừ đi 1 khoản trong timetable[timetable[“To”]==a][ETA] để tính là groundtime?)

Balance constraint giữa các ngày trong tuần:

If dow in range (2,8):

Node đầu ngày hôm sau = node cuối ngày hôm trước + arrival - departure

elif dow == 1:

first time node của thứ 2 = last time node của chủ nhật + arrival - departure

* Coverage constraint: Mỗi 1 số hiệu chuyến bay chỉ được khai thác duy nhất 1 loại tàu bay. (Vd: VN087 không thể khai thác đồng thời 321 và 787 tại cùng 1 thời điểm).

sum(model.assign\_fleet(sector, số hiệu, DOW, hour)) <= 1

Bổ sung thêm constraint về các chuyến bay bắt buộc phải bay

* Groundtime constraint (trong sách không có): tàu bay sau khi hạ cánh thì cần một khoảng thời gian groundtime nhất định tuỳ thuộc vào từng sân bay và loại tàu bay thì mới có thể bay tiếp được.

Nếu cộng thẳng groundtime tối thiểu vào giờ bay (điều chỉnh lại ETA) => problem: lúc tính chi phí sẽ bị sai.

* Airport constraint: một số đường bay chỉ dùng được tàu thân rộng, một số đường bay khác chỉ dùng được tàu AT7 do hạn chế về khoảng cách bay và cơ sở hạ tầng sân bay.

Model.assign\_fleet[sector==”HAN-CDG”, fleet ==”B787-9”].fix(1)

* Product constraint: Để đảm bảo tính đồng nhất của sản phẩm, các chuyến bay quốc tế trên 1 đường bay phải khai thác tại các khung giờ giống nhau và phải sử dụng loại tàu bay giống nhau giữa các ngày trong tuần (Vd: nếu HAN-ICN bay 1 chuyến ngày và 1 chuyến đêm thứ Hai thì tất cả các ngày trong tuần cũng tương tự, không thể có hôm bay trưa chiều, hôm bay sáng tối. Cần có constraint đảm bảo số tần suất quốc tế trải đều các ngày trong tuần thay vì bị tập trung vào 1-2 ngày (Chỉ áp dụng với schedule design optimization, ko áp dụng với fleet assignment optimization vì đã có sẵn thời điểm bay của từng chuyến bay).
* Số hiệu chuyến bay các ngày trong tuần phải dùng 1 loại tàu giống nhau.

Model.product\_constraint = Constraint(model.assign\_fleet[sector,flight no., fleet])

* Các chuyến bay tại các thị trường quan trọng (HAN-SGN, Nhật Bản) phải đạt được yêu cầu đề ra về loại tàu bay, thời điểm cất hạ cánh lý tưởng,…

Model.assign\_fleet[sector==”HAN-NRT”, fleet ==”B787-9”].fix(1)

* Aircraft constraint: số lượng tàu bay tại các khung giờ không được lớn hơn số tàu bay đã thuê mua.

Model.assign\_fleet + model.park\_ac <= maximum aircraft

* Slot constraint: tại các sân bay quốc tế chỉ được cất hạ cánh vào một số khung giờ nhất định.

Model.assign\_fleet[time<6].fix(0)

* Market share constraint: phải bay số tần suất tối thiểu (theo unit chuyến 321) để đạt được thị phần mục tiêu tại thị trường. (constraint: variable số chuyến 321 + variable số chuyến loại tàu khác x ghế loại tàu khác/ghế 321 >= số chuyến tối thiểu). (Chỉ áp dụng với schedule design optimization, ko áp dụng với fleet assignment optimization).

1. **Các vấn đề cần giải quyết**

* Làm thế nào để kết nối giữa các variable assign aircraft với tàu bay đến, tàu bay đi, tàu hiện có mà vẫn thể hiện được groundtime constraint.

1. **Note for variables**

* Sử dụng indexed components và function để xây các constraints (page 39)
* Sử dụng bounds keyword cùng với function để đặt constraint về số lượng tàu bay hiện có (page 41-42).
* Có thể đặt constraint sẵn cho variable dựa vào keyword bounds, within/domain. Ví dụ: đặt domain = pyo.Binary cho các variable về assign aircraft (page 41).
* Có thể assign variable bằng Python operator ‘=’. Set upper và lower bound = method setlb, setub (page 43).
* Variable có thể được cố định, ko bị optimizer thay đổi nếu dùng method fix, dùng method unfixed để đảo ngược lại (page 43).

1. **Note for objectives**

* pyo.value(model.objective) được dùng để tính ra giá trị của objective sau khi model thực hiện tối ưu (p45).
* Keyword argument “sense” để đặt pyo.maximize hoặc pyo.minimize

1. **Note for constraints**

* Tham khảo keyword rule và áp dụng function (p47) để làm constraint về groundtime (sau khi tàu gắn với số hiệu chuyến bay đáp xuống sân bay thì phải mất 1 khoảng tgian sau mới được phép cất cánh, khoảng tgian này phụ thuộc vào rule xem chuyến bay trước và sau khi cất cánh thuộc loại chuyến bay quốc tế dài, ngắn, nội địa và loại tàu bay).