Rami El Mawas <rami@berkeley.edu>
CE 291

DYNAMIC SECTORIZATION IN ACTION 2

Outline

- Introduction
- Modeling
- Building ASDI Raw Data
- Conclusion

Introduction

Introduction, Motivation, problem Statement, ATC Workload Factors, Fixed vs Dynamic boundaries, Big Picture

Introduction

- Air traffic controllers are persons who operate the air traffic control system to expedite and maintain a safe and orderly flow of air traffic. They typically accepts traffic from, and ultimately passes traffic to, the control of a Terminal Control Center or of another Center.
- Sector are geographic subdivisions in the airspace. Every sector belongs to a center.

Motivation

- Humans in charge: Air Traffic Controllers
- How to improve their job?
- Objective : minimize the workload of the Air Traffic Controllers
- Problem : the complexity of NAS
- Approach chosen in this project : dynamic boundaries

Problem Statement

- The main purpose is here to move the boundaries of the sectors in order to decrease the air traffic controller workload. By focusing only on en-route sectors, the choice of the Air Traffic Controller workload metric which was made is the peak count in a sector.
- How to resectorize the national air space (NAS) so that the maximum peak count of aircrafts per polygon is as small as possible.
- Analyzing flight plan is implicated directly into capacity distribution of the system

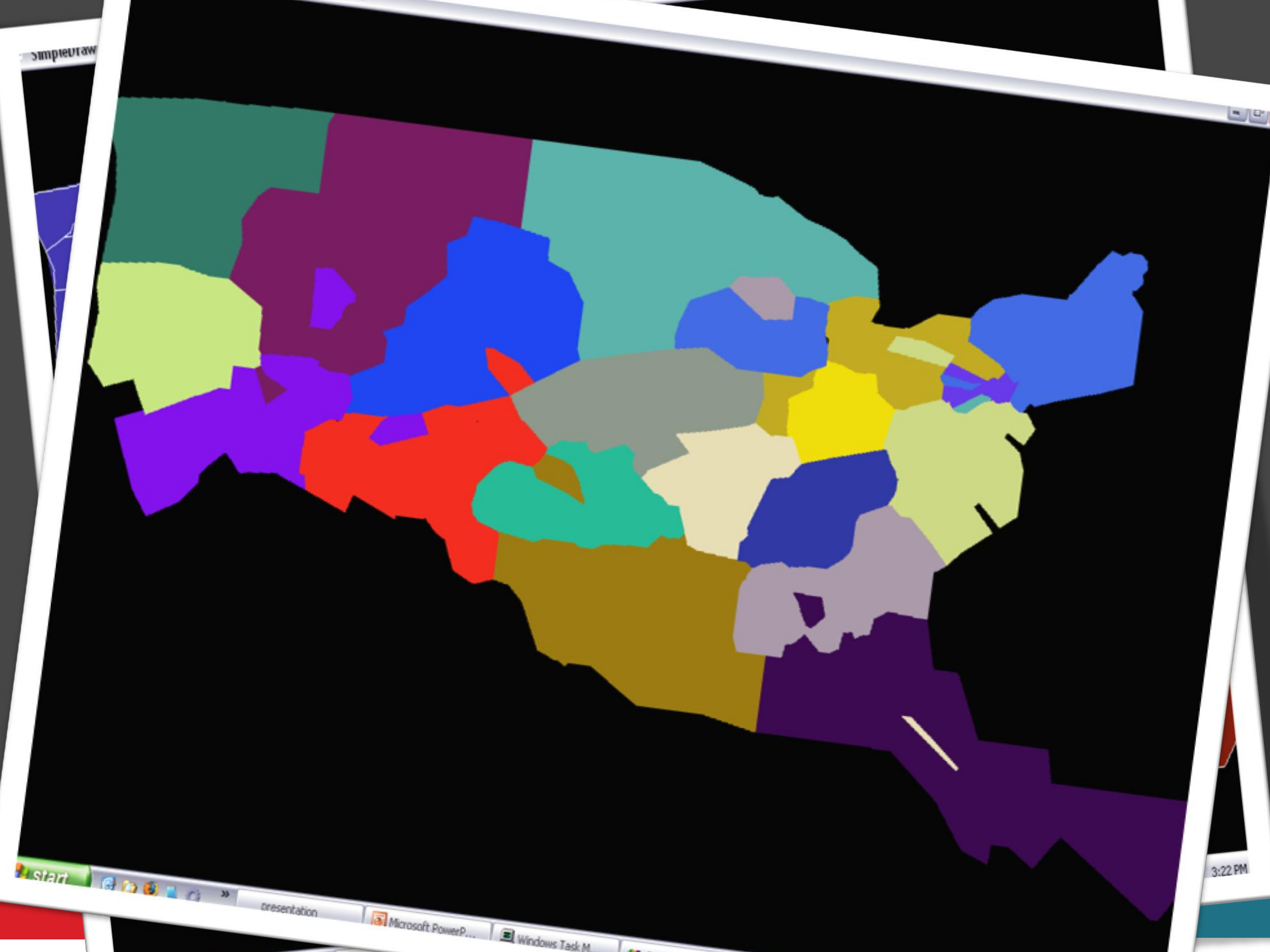
ATC Workload Factors

- The primary difficulty in modeling the ATC workload is to select the most relevant factors amongst the many potential variables affecting it: the number of aircraft (which contains peak counts), the presence of conflicts (number of intersections), the climbing and descending flights (traffic mix with arrivals, departures and overflights), the sector itself (geometry, size), the coordination and of course the traffic flow structure

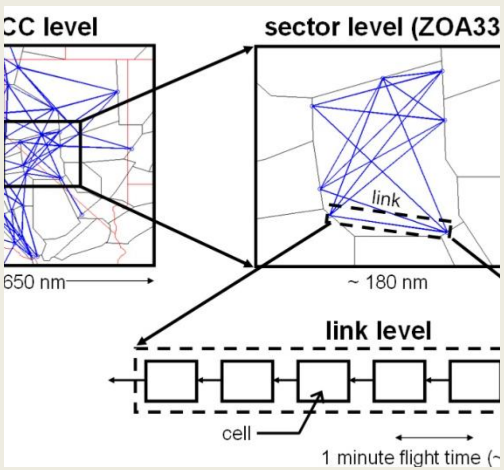
Fixed vs Dynamic Boundaries

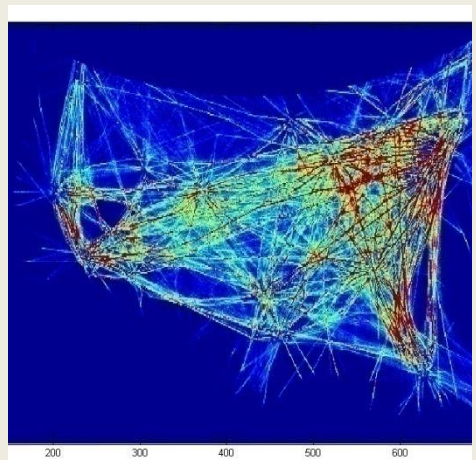
- Fixed boundaries: start with an initial sectorization setting and then change the center to which sectors belong.

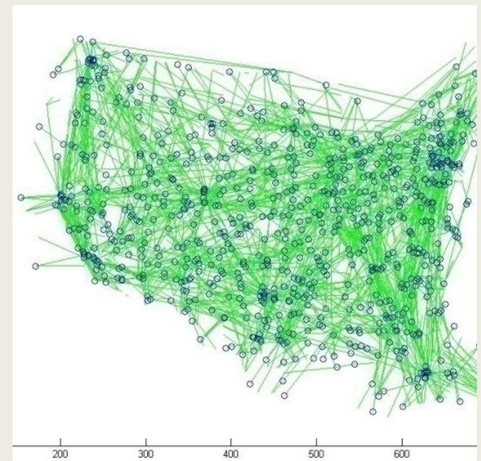
- Dynamic boundaries: change the geometry of the sectors



SimpleDraw
start
presentation
Microsoft PowerP...
Windows Task M...
3:22 PM







Modeling

Flow Model, Grid Model, Hybrid Model

Big Picture

ASDI

Flow Model | Occupancy Grid

↓

Weighted graph

↓

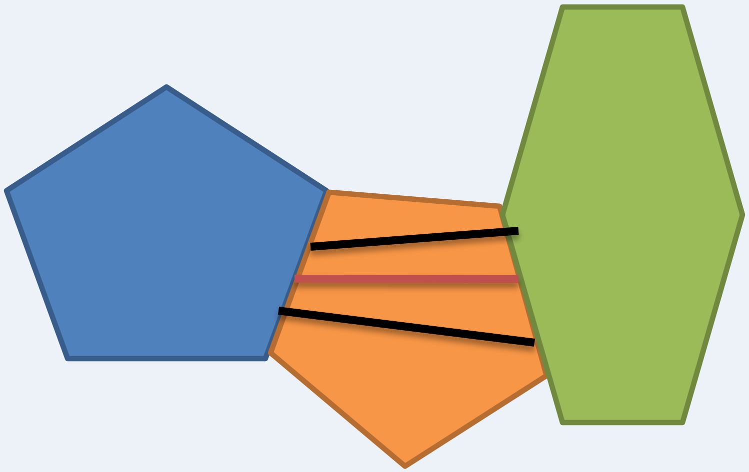
Spectral Bisection

↓

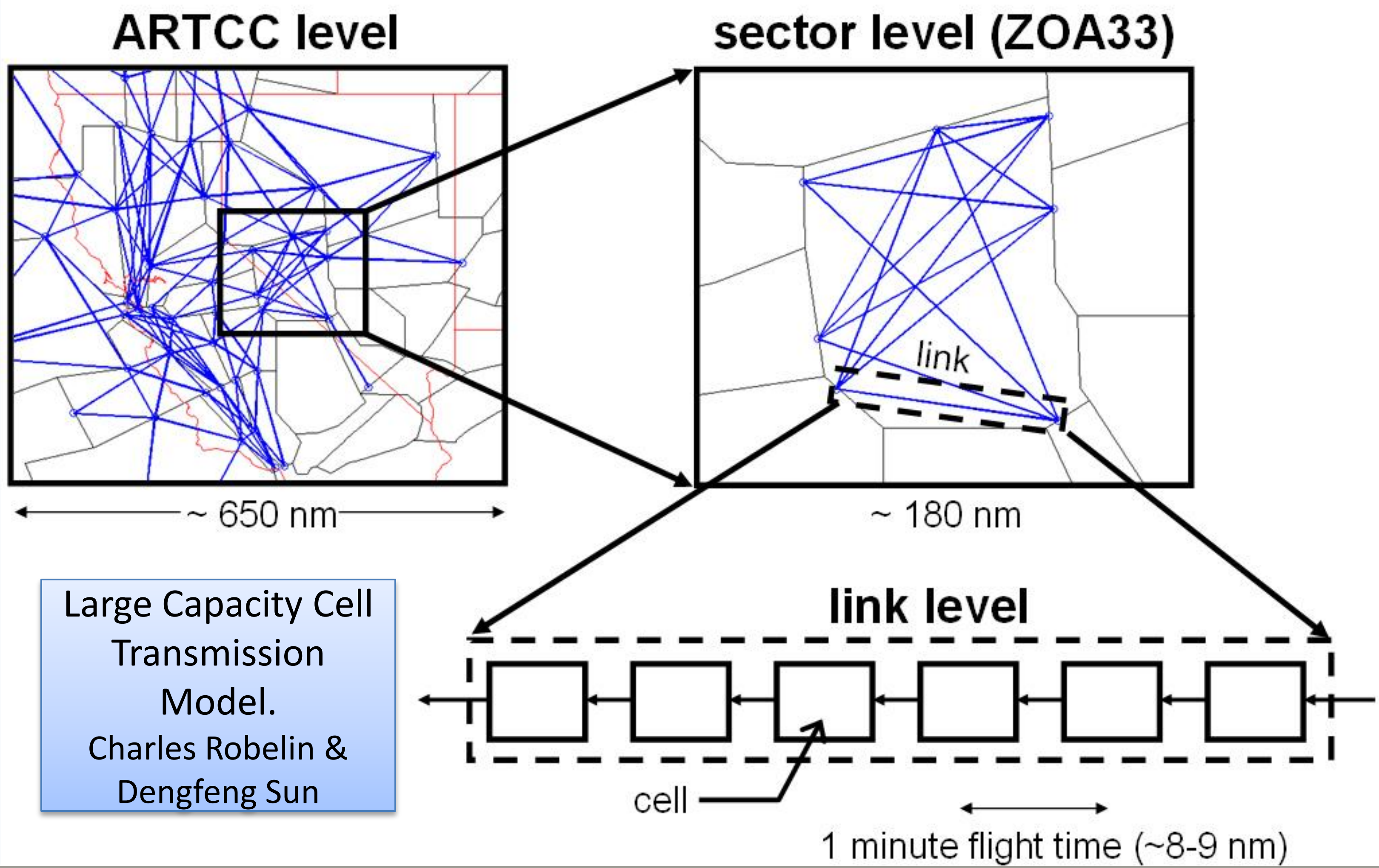
Geometric Sector Construction

Flow Model

- This model is able to give a graph representing the main flows for the NAS by aggregating the routes at a sector level.

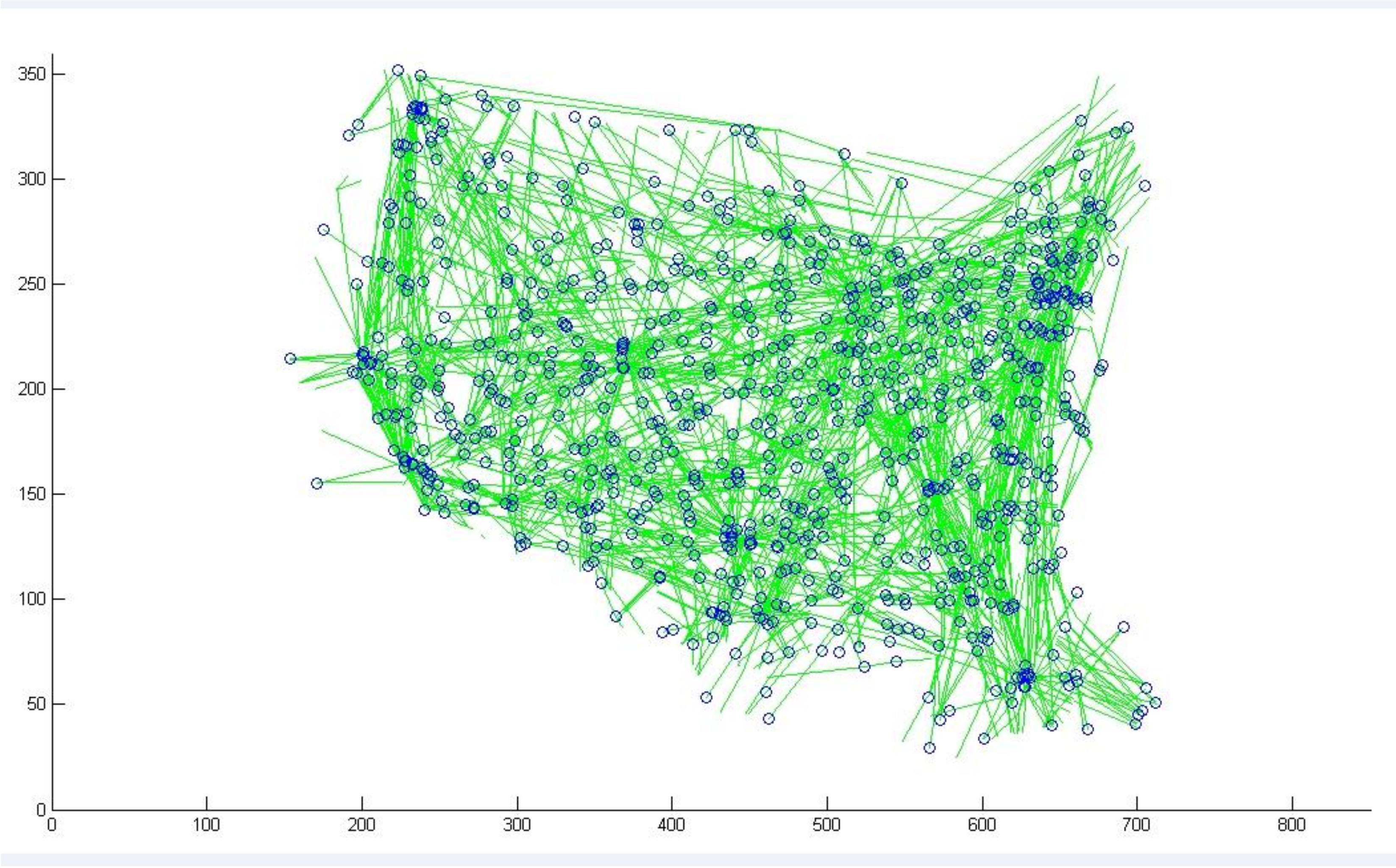


Flow Model

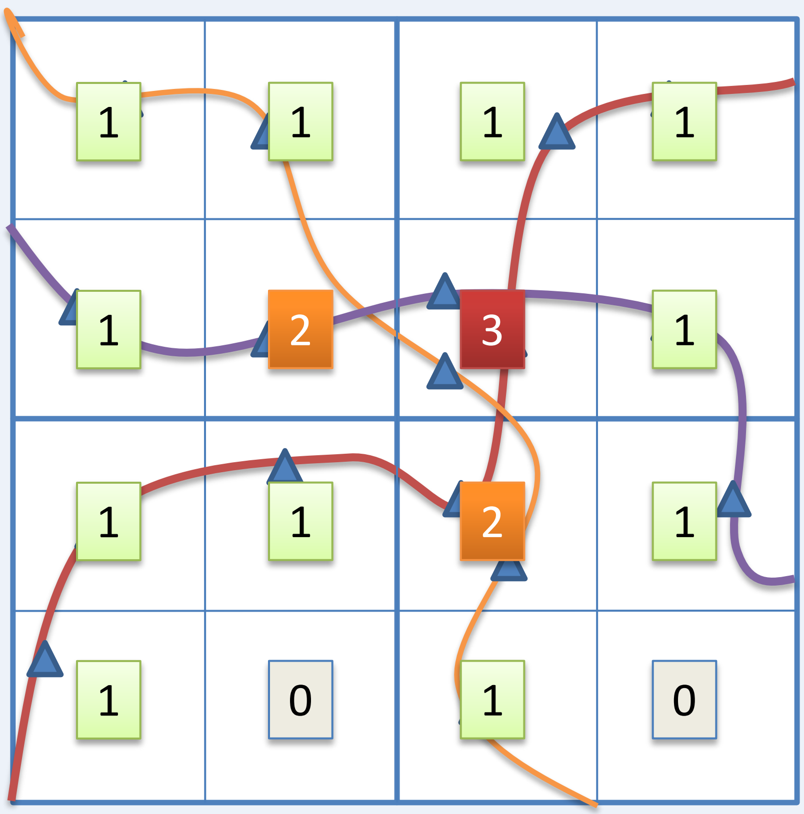


Large Capacity Cell Transmission Model.
Charles Robelin & Dengfeng Sun

Flow Model

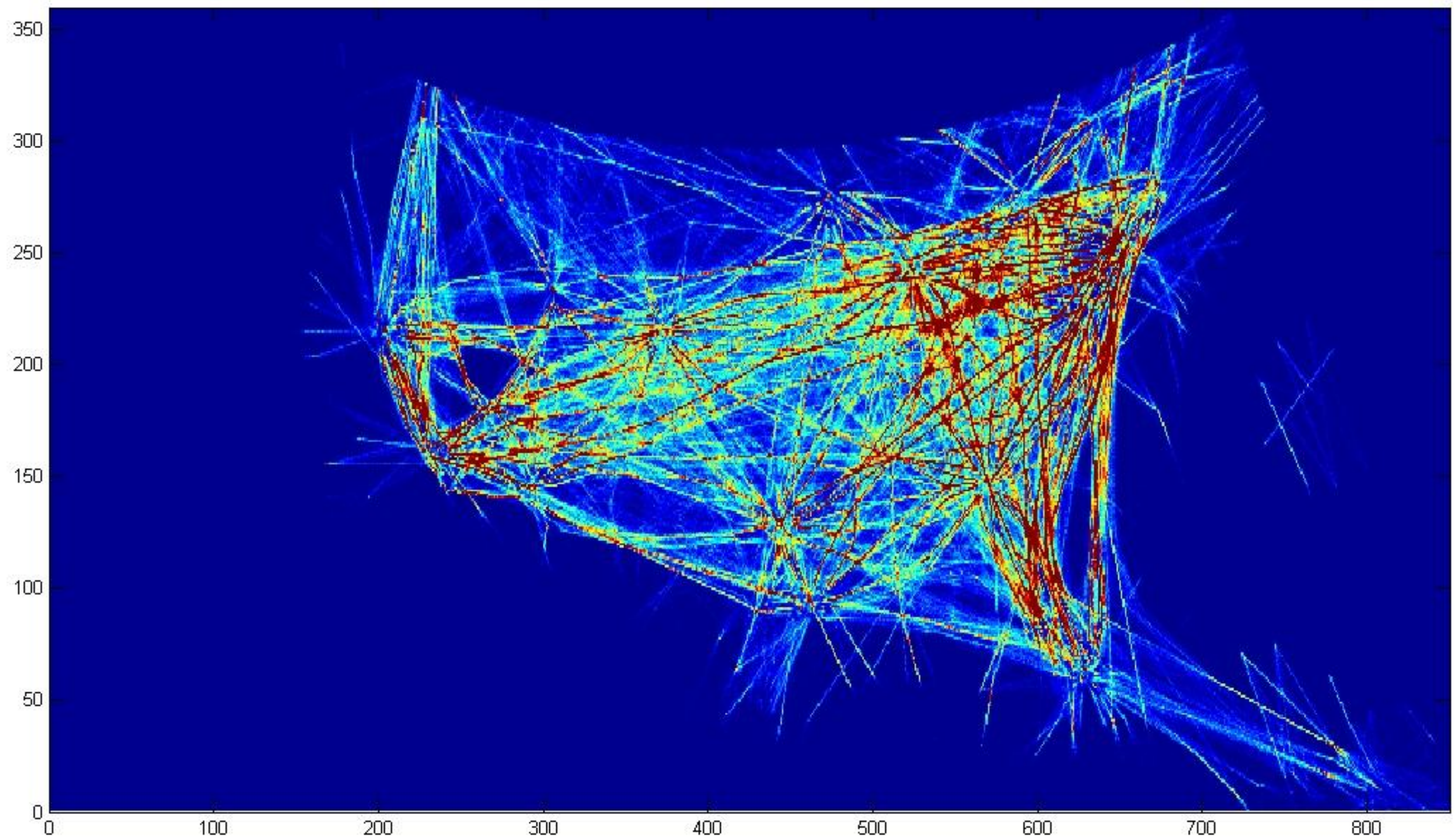


Occupancy Grid

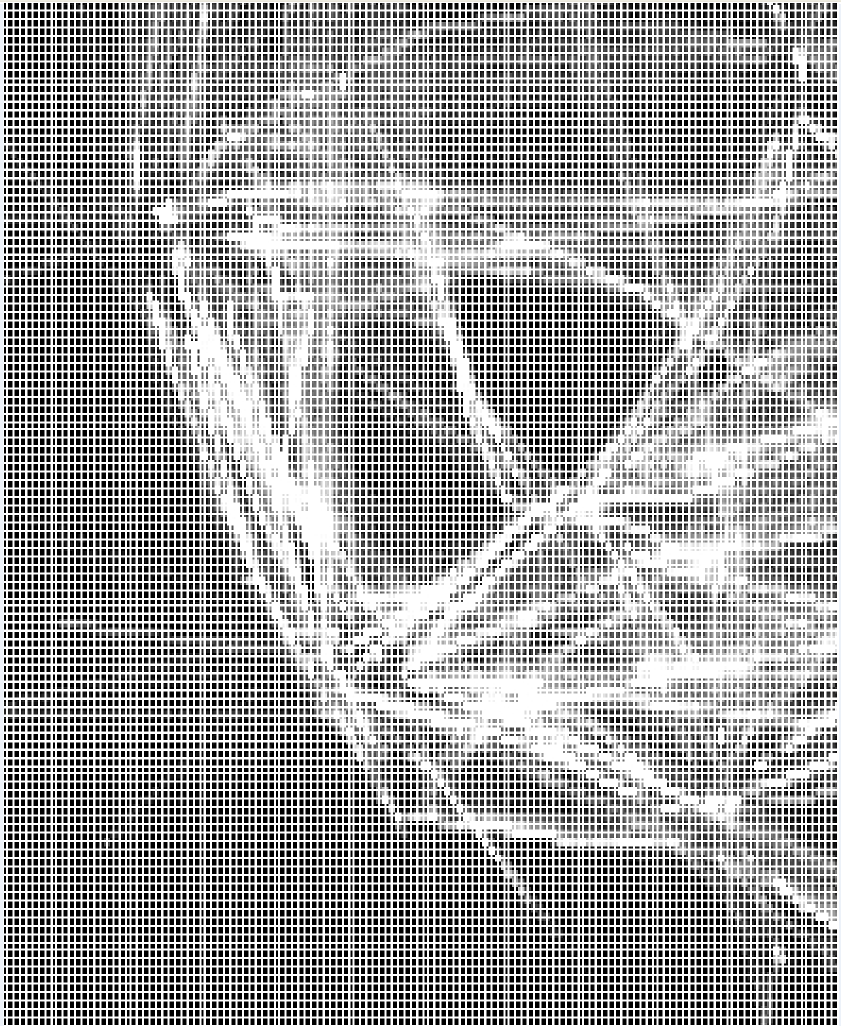


Compute cumulative or peak number of aircrafts for every cell. Model developed by S. Martinez

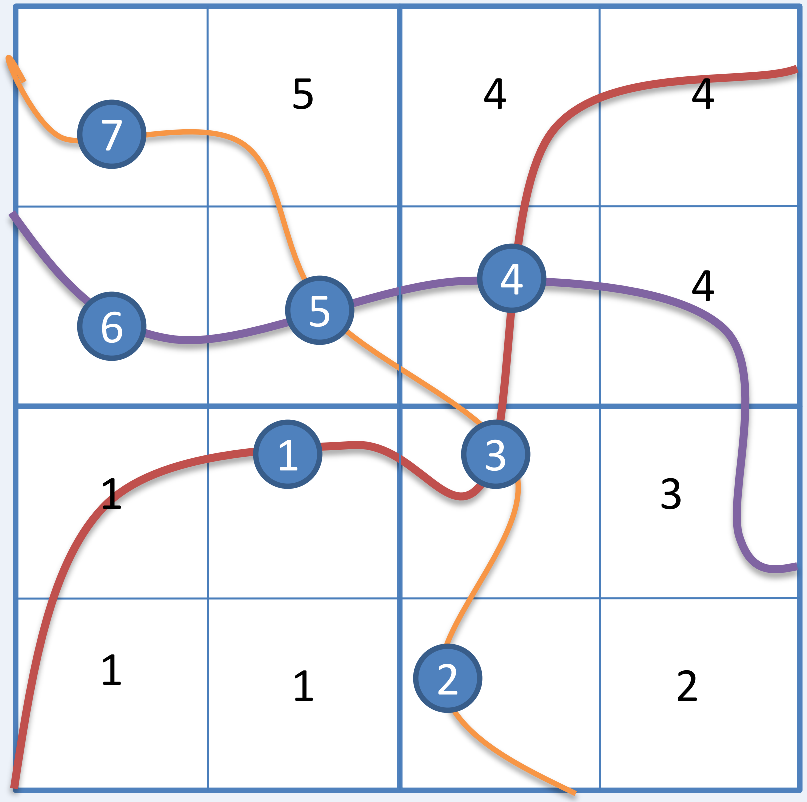
Occupancy Grid



Occupancy Grid

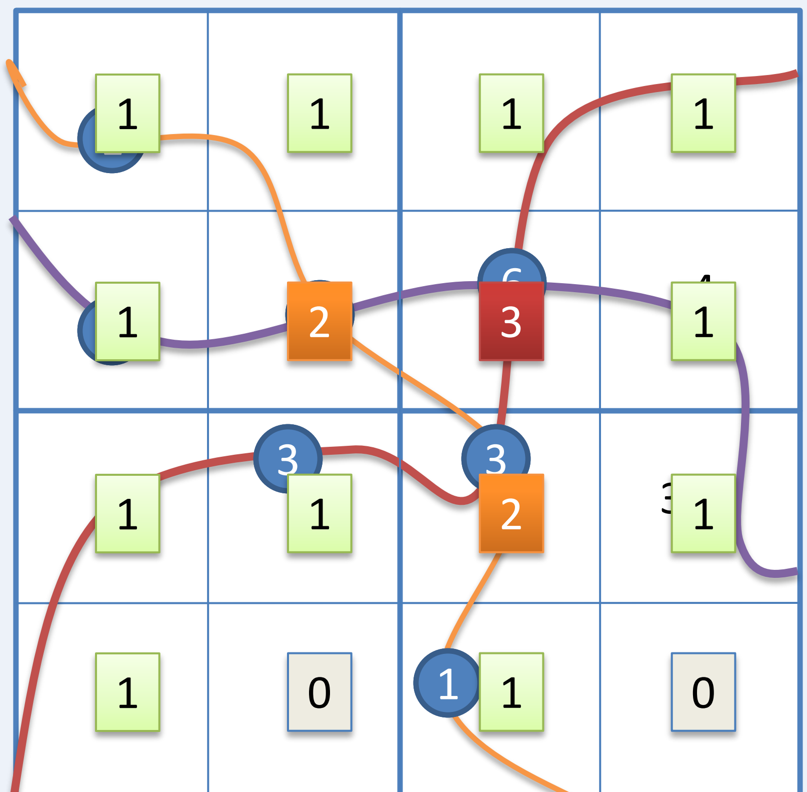


Hybrid Model



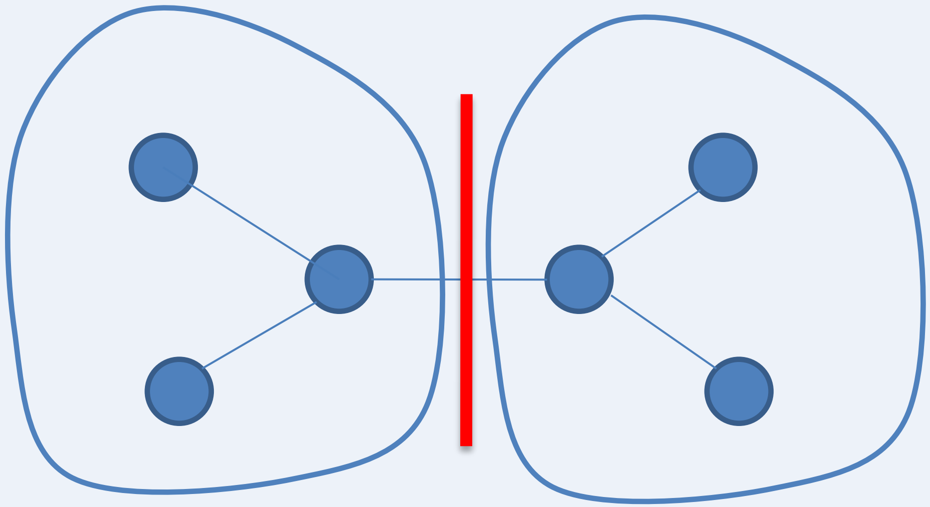
Use any Clustering Algorithm to influence all the cells by a node. Region growing or K-mean

Hybrid Model



Add the count in each cell affected by the corresponding nodes. Now we have weighted nodes

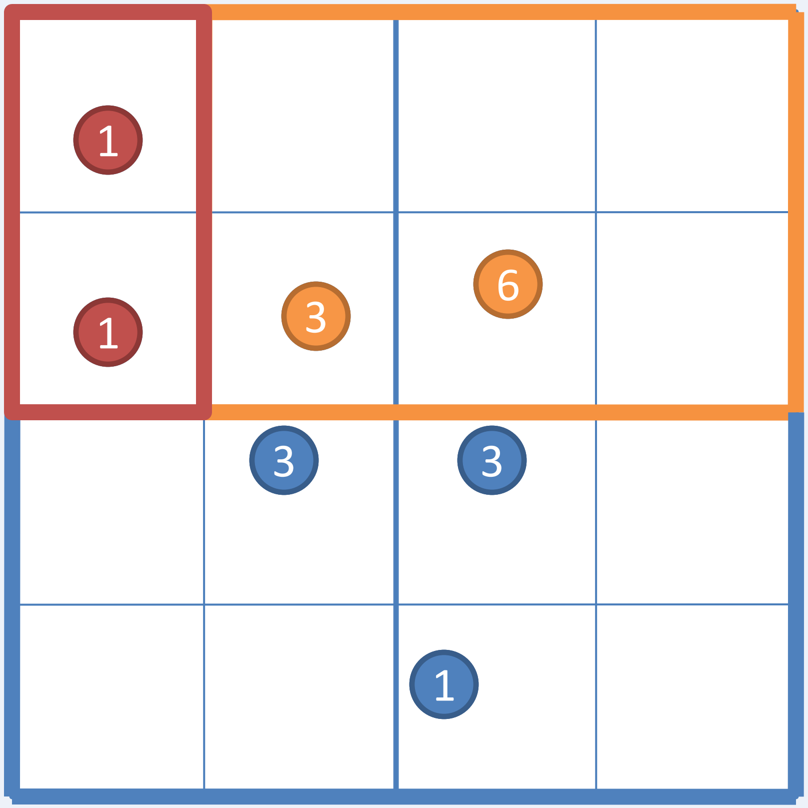
Spectral Bisection



It is based on the Laplacian Matrix
- Keep highestconnectivity
- Cut the least edges.

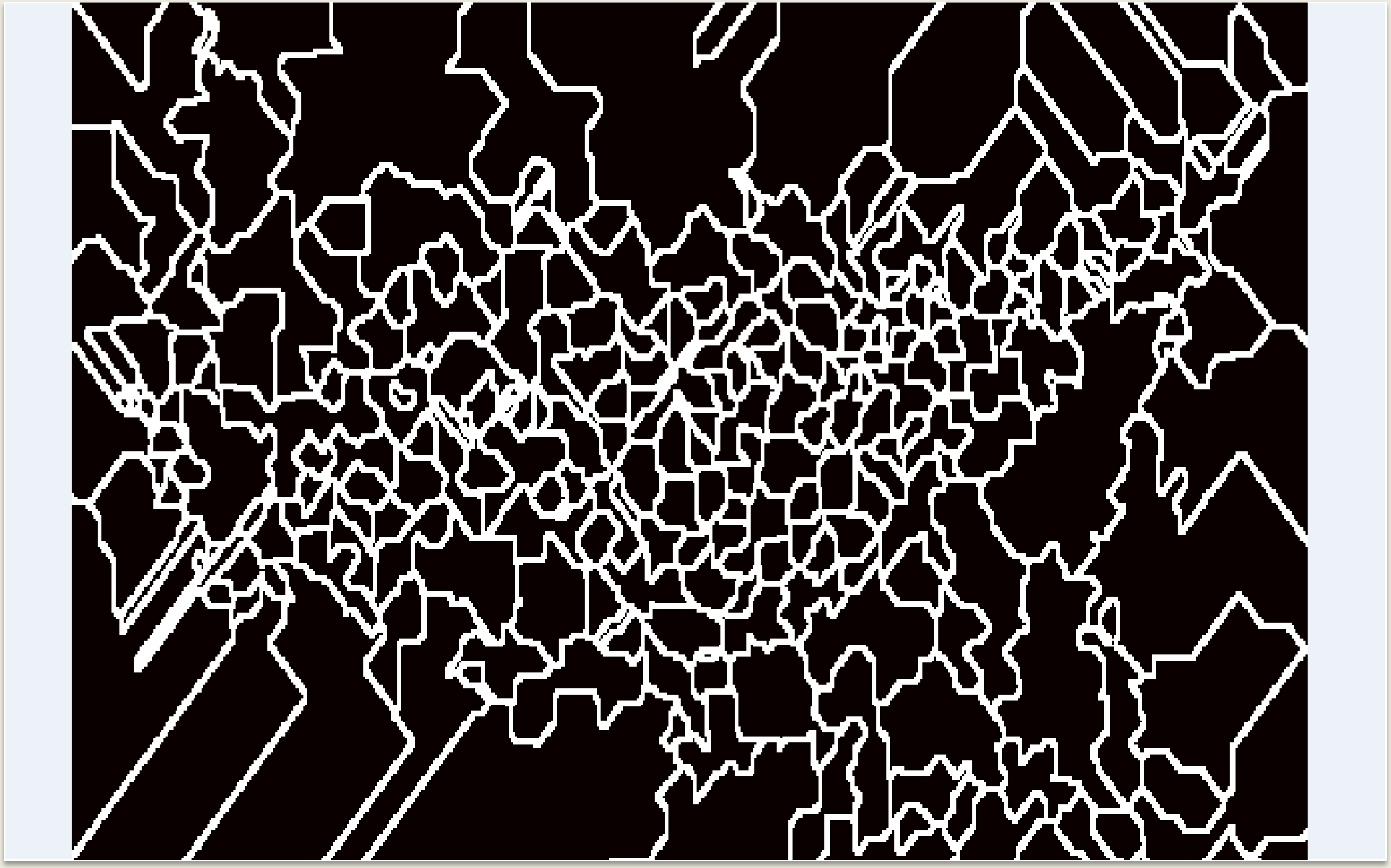
Repeat the same but with constraints are: number of sectors (actual number) or max peak count (less than limit)

Sector construction

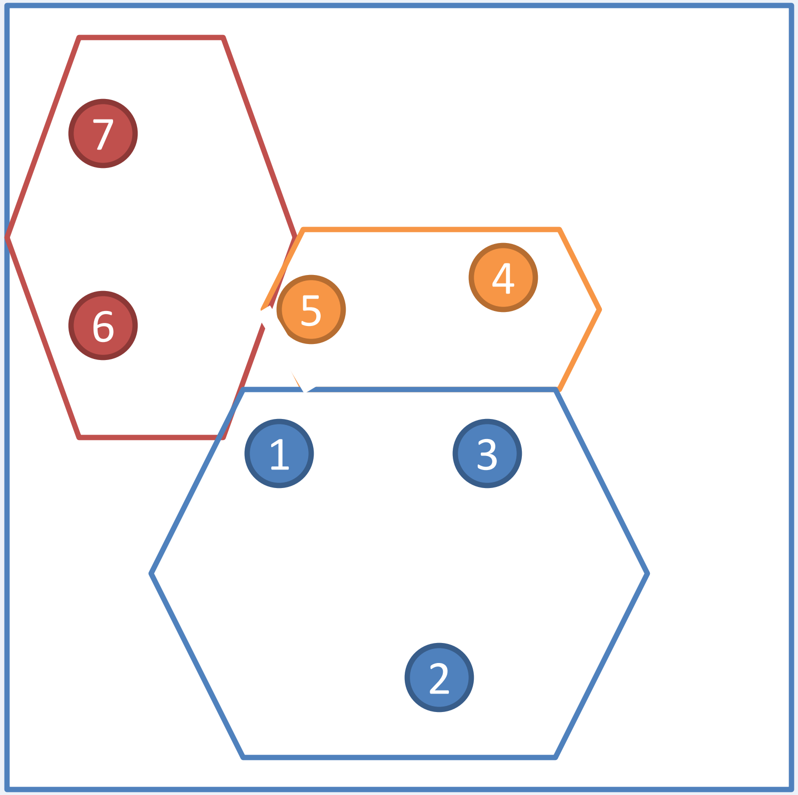


Use Influence area

Occupancy Grid

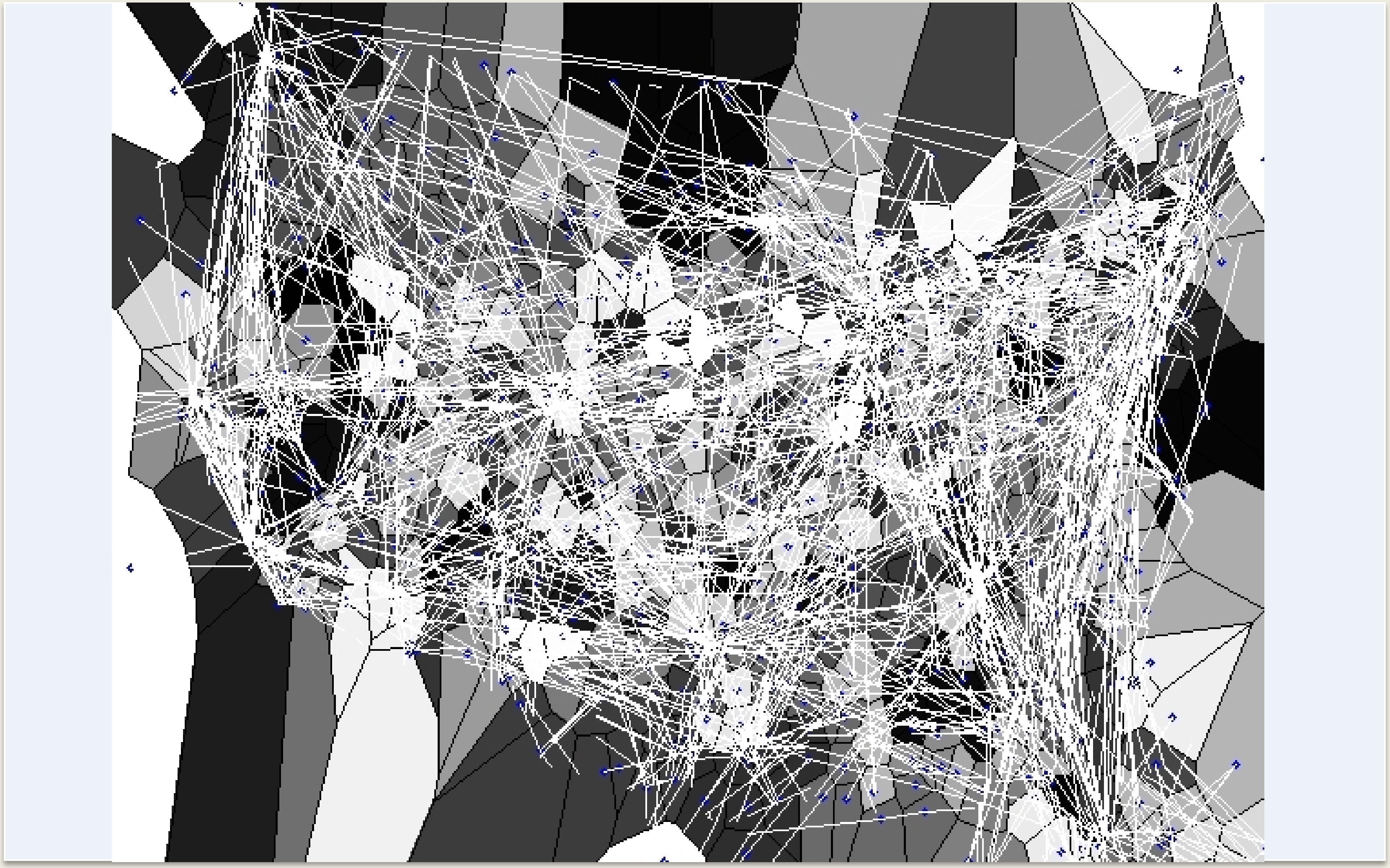


Sector construction



Use Voronoi diagrams

Occupancy Grid



Building ASDI Raw Data

ASDI format, script language, PScript, Route finder, ICAO/IATA

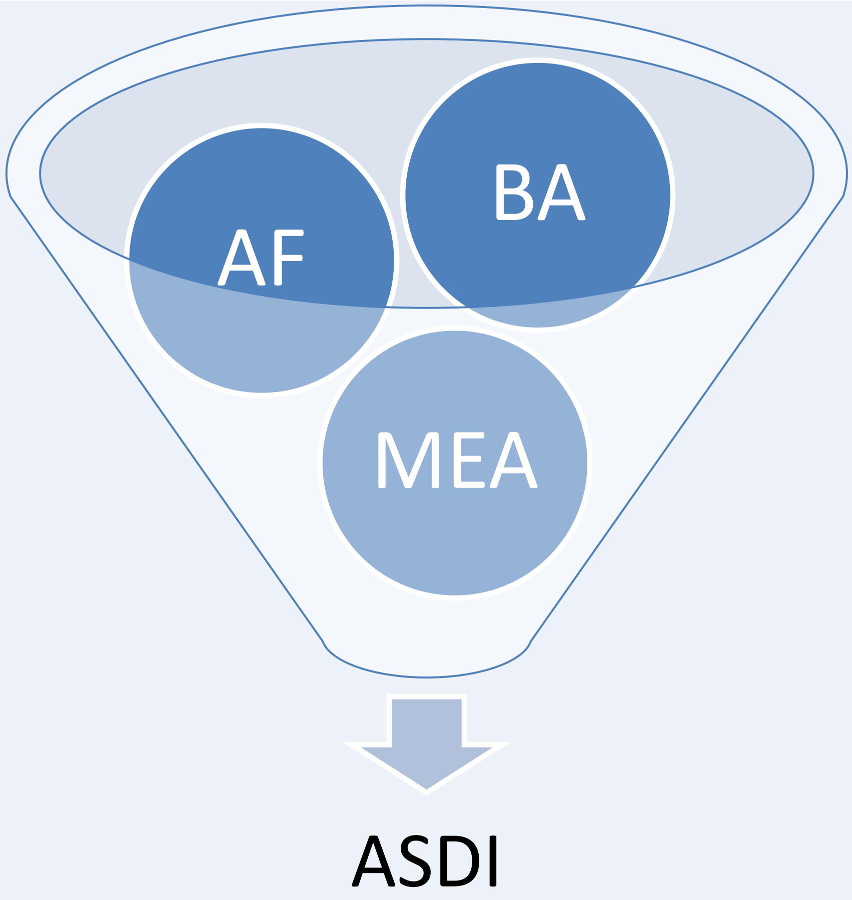
Big Picture

Airline Tables

BERLIN							**BER**
→ **Amsterdam** AMS							+1:00
1234567	06.00	07.35		01:35	KL1818	→	16/01-24/03
1234567	10.35	12.05		01:30	KL1822	→	16/01-24/03
1234567	12.10	13.45		01:35	KL1824	→	16/01-24/03
1234567	17.25	18.55		01:30	KL1830	→	16/01-24/03
12345-7	20.00	21.30		01:30	KL1834	→	19/02-23/03
→ **Paris** PAR							+1:00
1234567	07.10	08.55	2	01:45	AF1135	→	17/01-24/03
1234567	09.55	11.40	2	01:45	AF1435	→	17/01-24/03
1234567	12.55	14.40	2	01:45	AF1735	→	17/01-24/03
1234567	15.55	17.40	2	01:45	AF2035	→	17/01-24/03
1234567	18.55	20.40	2	01:45	AF2335	→	17/01-24/03
→ **Prague** PRG							+1:00
123456-	08.55	09.55	2	01:00	AF6301(OK)	→	17/01-24/03
12345-7	19.10	20.10	2	01:00	AF6303(OK)	→	17/01-23/03
BEYROUTH							**BEY**
→ **Paris** PAR							+1:00
1234567	02.05	05.45	2	04:40	AF563(ME)	→	17/01-24/03
1234567	08.05	11.45	2	04:40	AF569(ME)	→	17/01-24/03
1234567	15.55	19.35	2	04:40	AF565	→	17/01-24/03

ASDI

Big Picture



ASDI Data

- ASDI is an acronym for Aircraft Situation Display to Industry. The ASDI data stream is a service made available through the U.S. Department of Transportation's Volpe Transportation Center. The ASDI stream consists of data elements which show the position and flight plans of all aircraft in U.S. and optionally, UK airspace

ASDI Data (text file)

```
0A1F09235440KSDFTZ N604GW/673 242 088 3908N/08609W
0A2009235446KSDFTZ FFT608/427 343 110 3949N/08641W
0A2109235452KSDFTZ N1327N/168 096 045 3850N/08542W
0A2209235441KDENTZ FFT494/587 323 130 4001N/10409W
0A2309235441KDENTZ SKW6284/952 204 072 3957N/10442W
0A2409235447KDENTZ QXE4180/151 273 155 3916N/10408W
0A2509235447KDENTZ EJA394/108 183 070 3936N/10451W
0A2609235453KDENTZ NWA546/029 322 120 4004N/10428W


0A1F09235440KSDFTZ N604GW/673 242 088 3908N/08609W
09 : day
23 : hour
54 : min
40 : sec
3908N : latitude 39 degre 08 min
08609W : longitude 086 degre 09 min
088 : FL
```

ASDI Data (xml file)


```
<asdiMessage sourceFacility="KZKC" sourceTimeStamp="2007-01-31T18:59:50.0Z">
    <trackInformation>
        <nxcm:aircraftId>N154NS</nxcm:aircraftId>
            <nxcm:computerId>
                <nxce:idNumber>254</nxce:idNumber>
            </nxcm:computerId>
            <nxcm:speed>493</nxcm:speed>
            <nxcm:reportedAltitude>
                <nxce:assignedAltitude>
                    <nxce:simpleAltitude>244C</nxce:simpleAltitude>
                </nxce:assignedAltitude>
            </nxcm:reportedAltitude>
            <nxcm:position>
                <nxce:latitude>
                    <nxce:latitudeDMS degrees="38" minutes="19" directio ORT
                </nxce:latitude>
                <nxce:longitude>
                    <nxce:longitudeDMS degrees="088" minutes="52" direc "WE
                </nxce:longitude>
            </nxcm:position>
    </trackInformation>
</asdiMessage>
```

Script Language

- Scripting languages (commonly called scripting programming languages or script languages) are computer programming languages that are typically interpreted and can be typed directly from a keyboard. Thus, scripts are often distinguished from programs, because programs are converted permanently into binary executable files (i.e., zeros and ones) before they are run. Scripts remain in their original form and are interpreted command-by-command each time they are run.Scripts were created to shorten the traditional edit-compile-link-run process.

Script Language: PScript

<u>Data retrieval</u>

s(n1:n2), s(n), s(n1:), s(‘val’)

i(n1:n2), i(n), i(n:), i(‘val’)

ls(in1,in2,in3,…) where in:=s,i,or

*, *(n)

<u>Logical</u>

or(in1,in2,…) where in:=s,i,ls

if(in) where in:=s,i,ls,or,not

not(in) where in:=s,i,ls,or

Script Language: PScript

Loops

hloop(n), hloop(n1:)

vloop(n), vloop(n1:)

Other

end

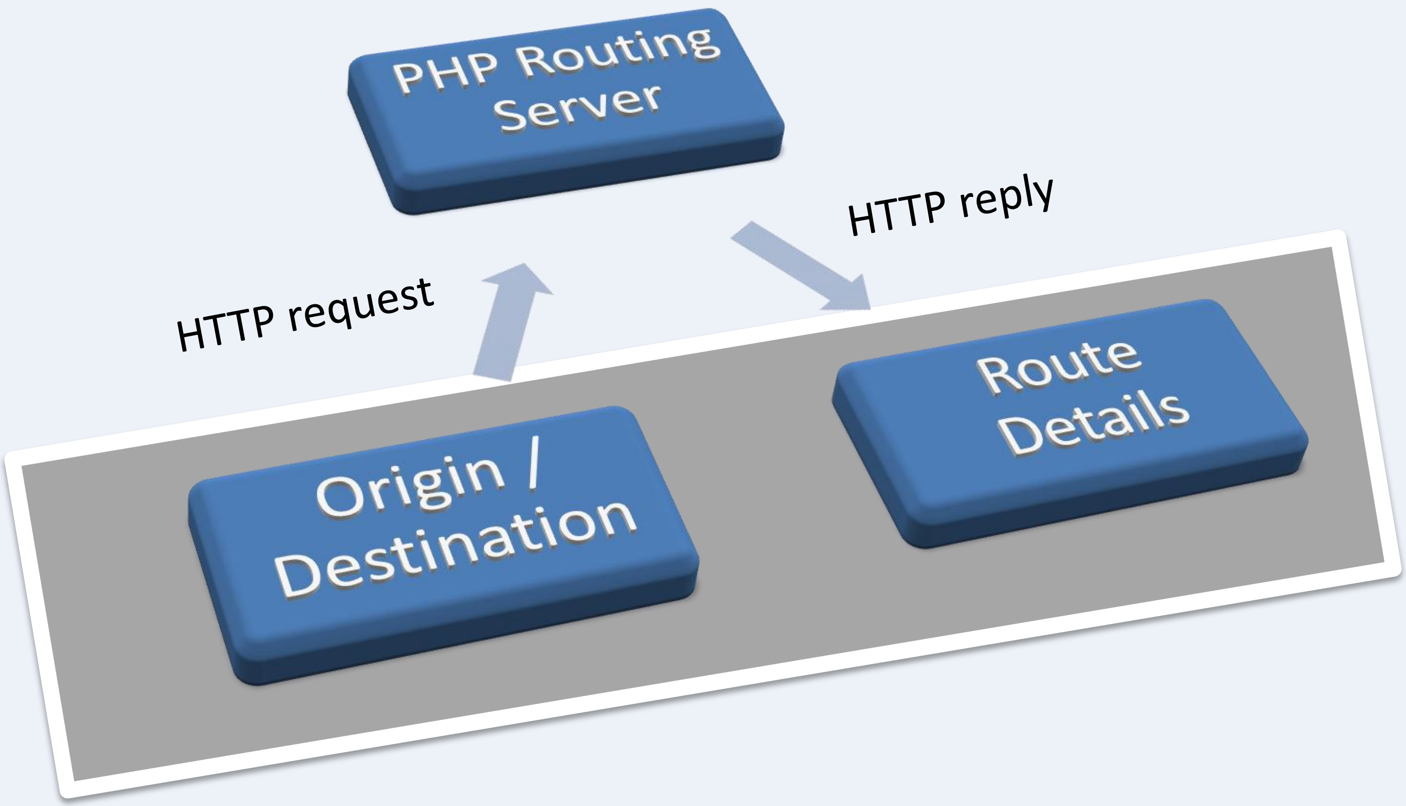
break

jump(n)

Route Finder

- Using the Aeronautical Information Management tools developed by the company ASA srl (Italy founded in 1991), RouteFinder offers a powerful set of tools aimed at effective flight planning (on PC flight simulation envinronment only!)
- They have set up a custom PHP interface script that we can query via HTTP

Route Finder



Route Finder

ID	DIST	Coords		Name/Remarks
LFPG	0	N49°00'46.00"	E002°33'00.00"	CHARLES DE GAULLE
NURMO	49	N49°49'34.00"	E002°45'19.00"	NURMO
DIPER	41	N50°20'45.00"	E002°03'41.00"	DIPER
VESAN	2	N50°22'19.00"	E002°01'35.00"	VESAN
RATUK	23	N50°39'25.00"	E001°38'11.00"	RATUK
SOVAT	10	N50°46'46.00"	E001°28'00.00"	SOVAT
SANDY	23	N51°03'51.00"	E001°04'03.00"	SANDY
EGLL	63	N51°28'39.00"	W000°27'41.00"	HEATHROW

Charles De Gaulle to Heathrow Airport

ICAO vs IATA Codes

- The ICAO airport code or location indicator is a four-letter alphanumeric code designating each airport around the world. These codes are defined by the International Civil Aviation Organization.The ICAO codes are used by air traffic control and airline operations such as flight planning.

- An IATA airport code, also known an IATA location identifier, IATA station code or simply a location identifier, is a three-letter code designating many airports around the world, defined by the International Air Transport Association (IATA). The characters prominently displayed on baggage tags attached at airport check-in desks are an example of a way these codes are used.

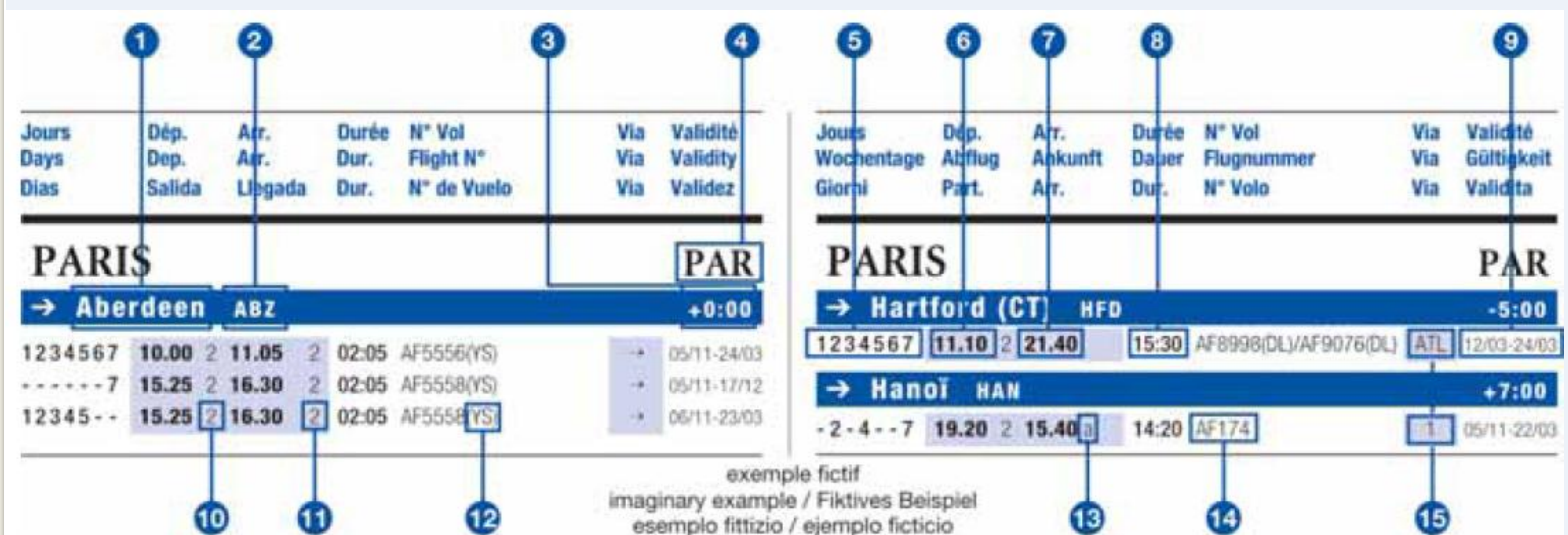
Big Picture

Airline Tables → IATA O/D → ICAO O/D → Routes → ASDI

Airline Companies Timetable

- All airline company issue every 7 months a detailed timetable of all its air flights: Origin, Desitination, Local time, operating days, Depature time, arrival time, duration, validity period

Air France



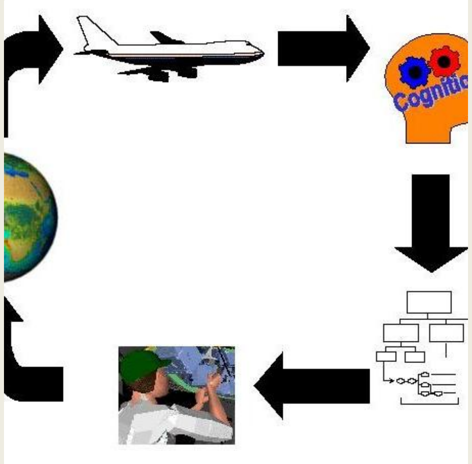
Jours Giorni	Dép. Part.		Arr. Arr.		Durée Dur.	N° Vol N° Volo	Via Via	Validité Validita
PARIS								**PAR**
→ San Francisco (CA) SFO								-7:00
1234567	10.15	2	12.30	I	11:15	AF084	→	09/04-27/10
1234567	10.15	2	12.55	I	11:40	AF084	→	25/03-08/04

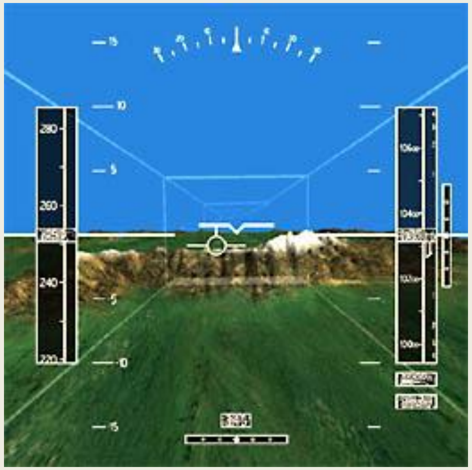
AIR FRANCE
faire du ciel le plus bel endroit de la terre

PScript (example)

```
S(3)s(' ')or(s('+'),s('-'))i(1:)s(':')i(2)
vloop(1:)
        hloop(7)
                or(i(1),s('-'))s(' ')
        end
        i(2)s('.')i(2)s(' ')
        cif(ls(or(i(1),s(1)),s(' ')))
                or(i(1),s(1))s(' ')
        end
                        i(2)s('.')i(2)s(' ')
        hloop(2)
                cif(ls(or(i(1),s(1)),s(' ')))
                        or(i(1),s(1))s(' ')
                end
        end
        i(2)s(':')i(2)s(' ')
        hloop(1:)
                s(2)i(1:)
                cif(s('('))
                 s('(')or(s(3),s(2),ls(s(1),i(1)))s(')')
                end
                cif(s('/'))
                        s('/')
                end
        end
        s(' ')or(*(1),s(3),i(1))s(' ')
        i(2)s('/')i(2)s('-')i(2)s('/')i(2)
end
```

Chunk of the code that parse the AF file





Conclusion

Conclusion & Future Work

- Validate the Dynamic sectorization algorithm over Europe and compare the results with the actual sectors.