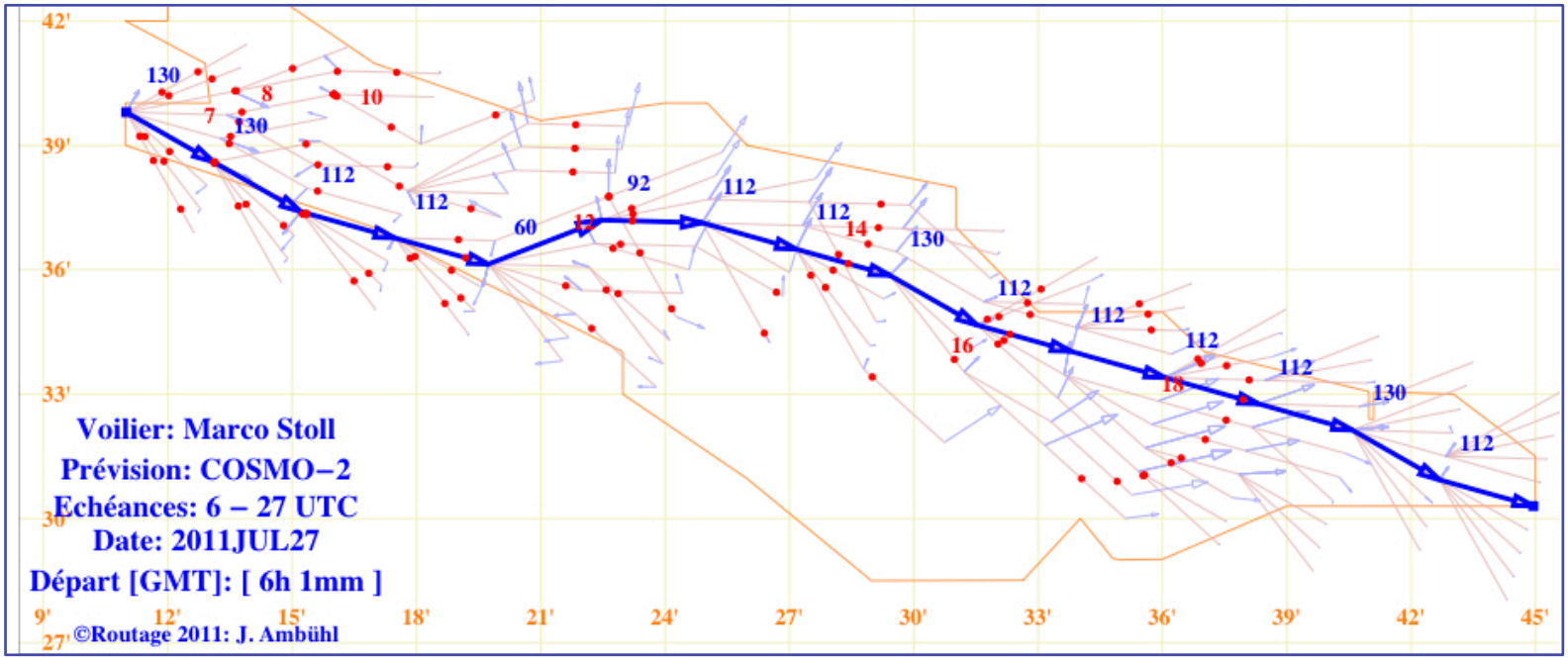
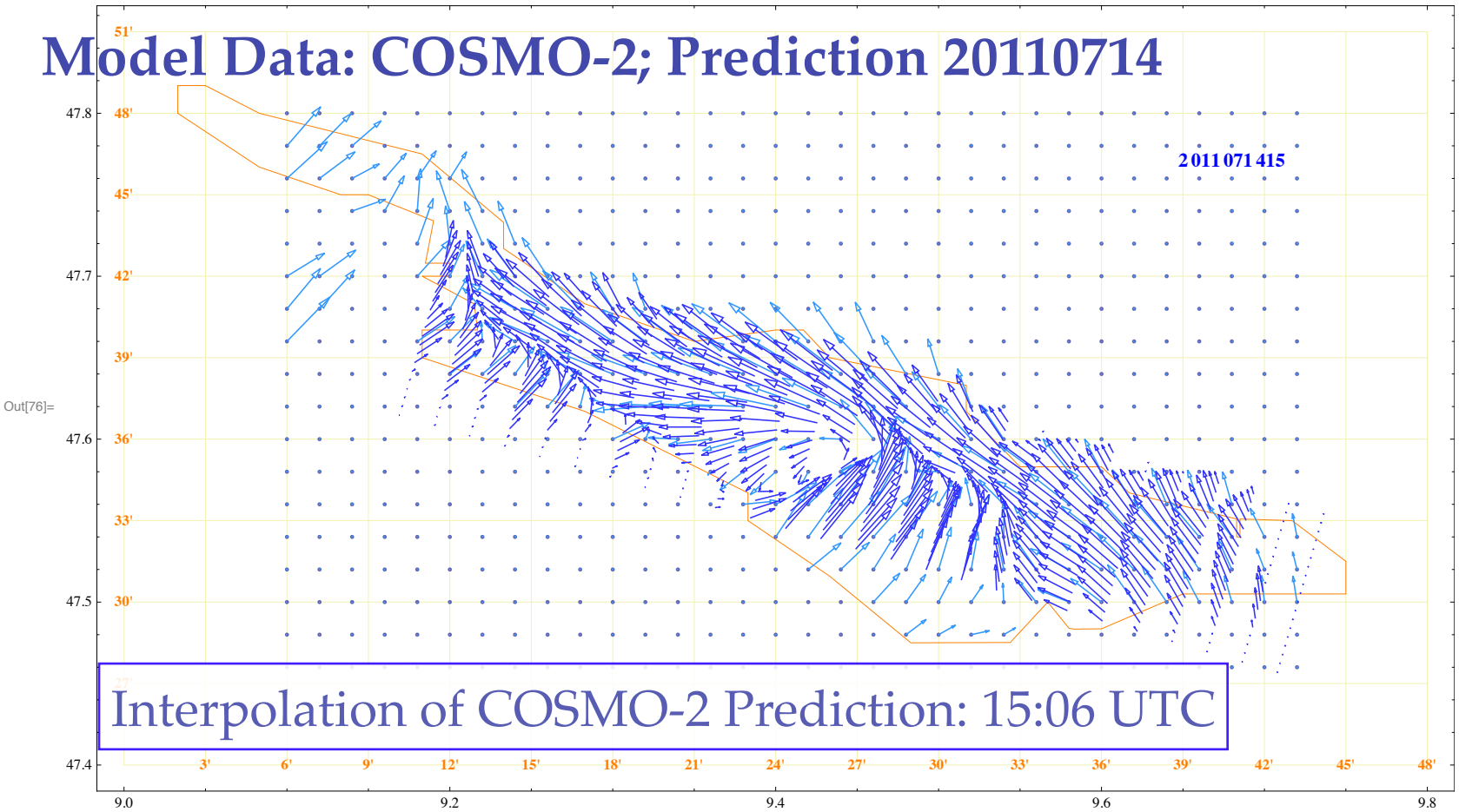
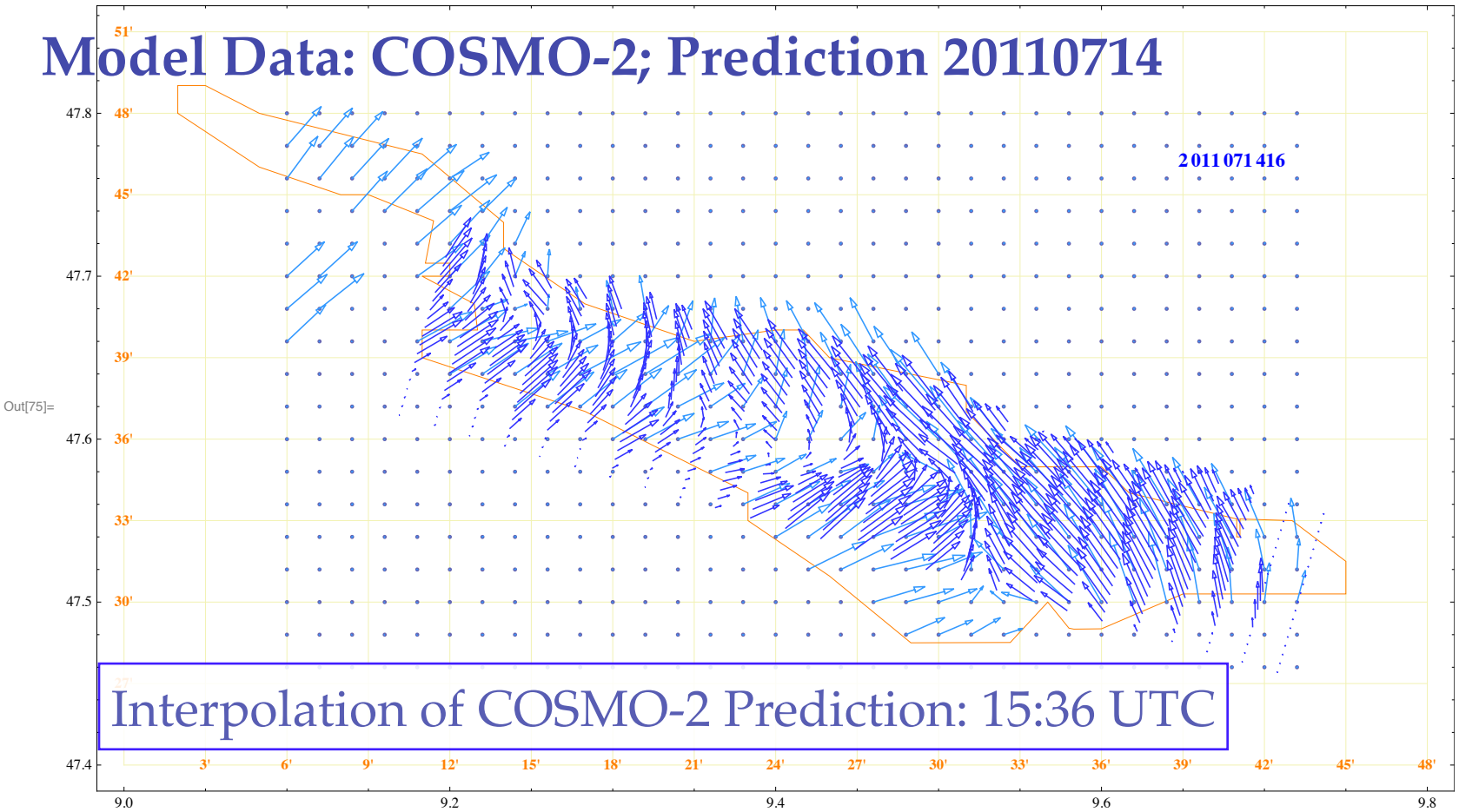
Lake routing based on dynamical programming



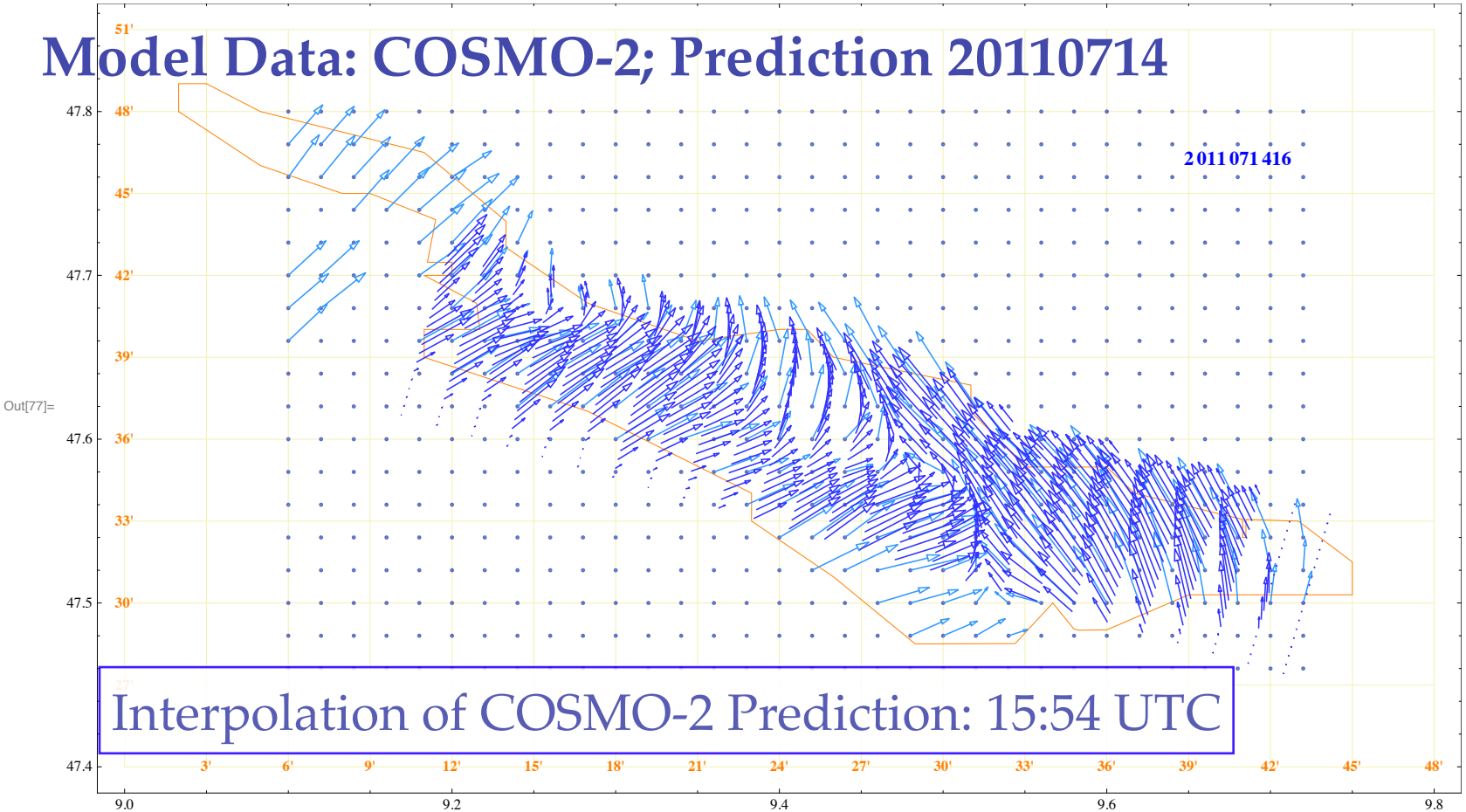
Model Data: COSMO-2; Prediction 20110714



Model Data: COSMO-2; Prediction 20110714



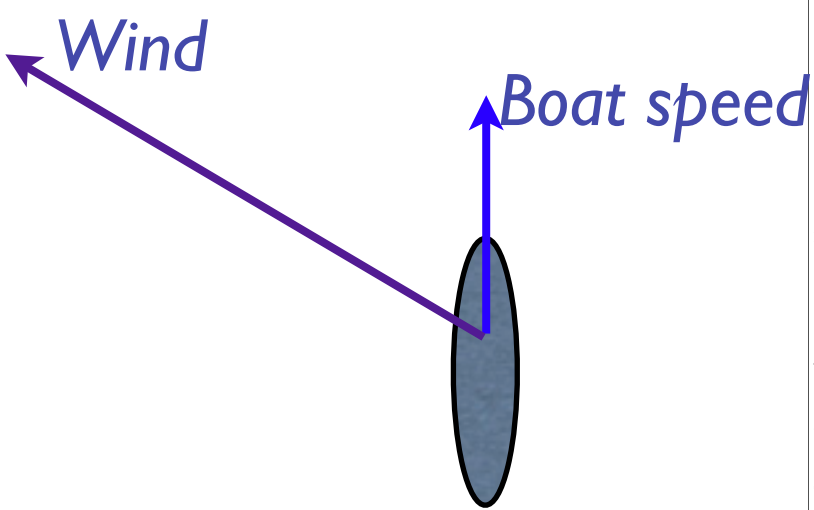
Model Data: COSMO-2; Prediction 20110714

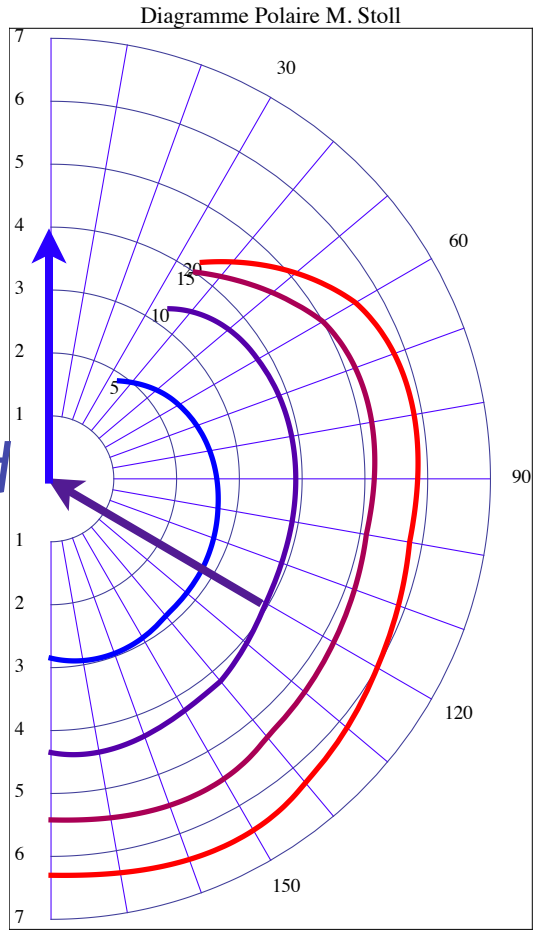


Interpolation of COSMO-2 Prediction: 15:54 UTC

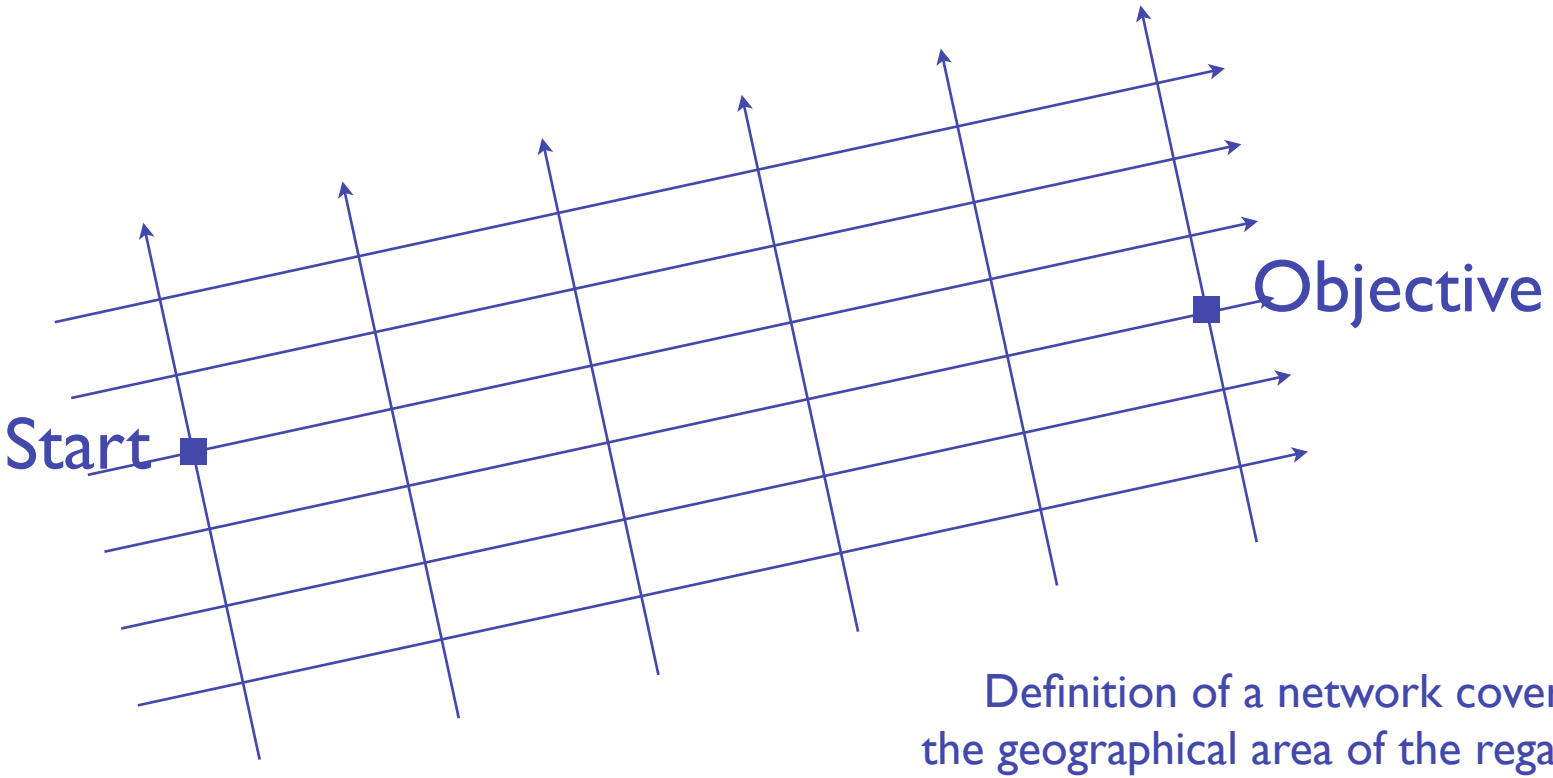
Customer profile

Polar diagramme of a sailing boat



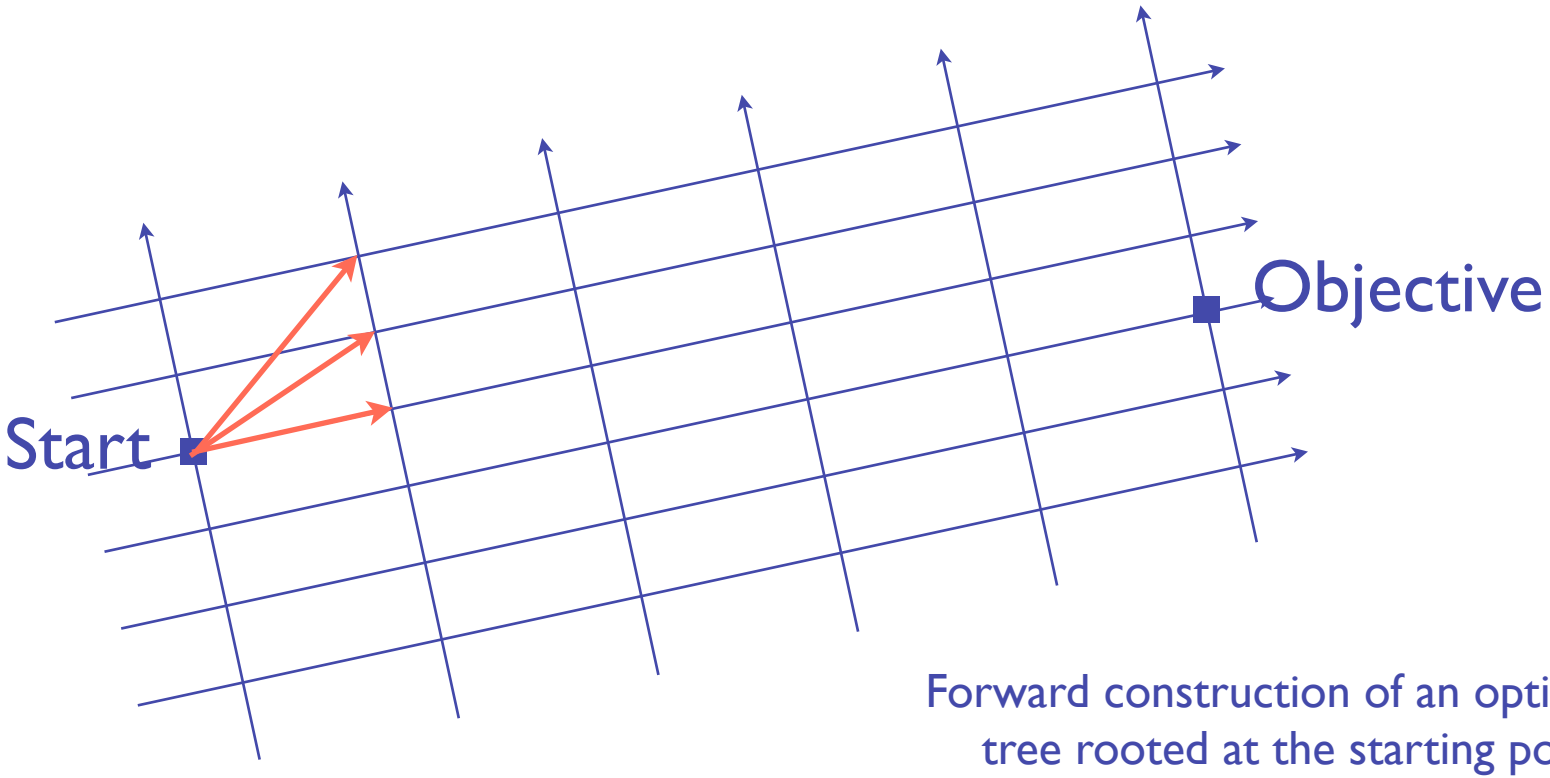


Decision Scheme: Dynamical Programming



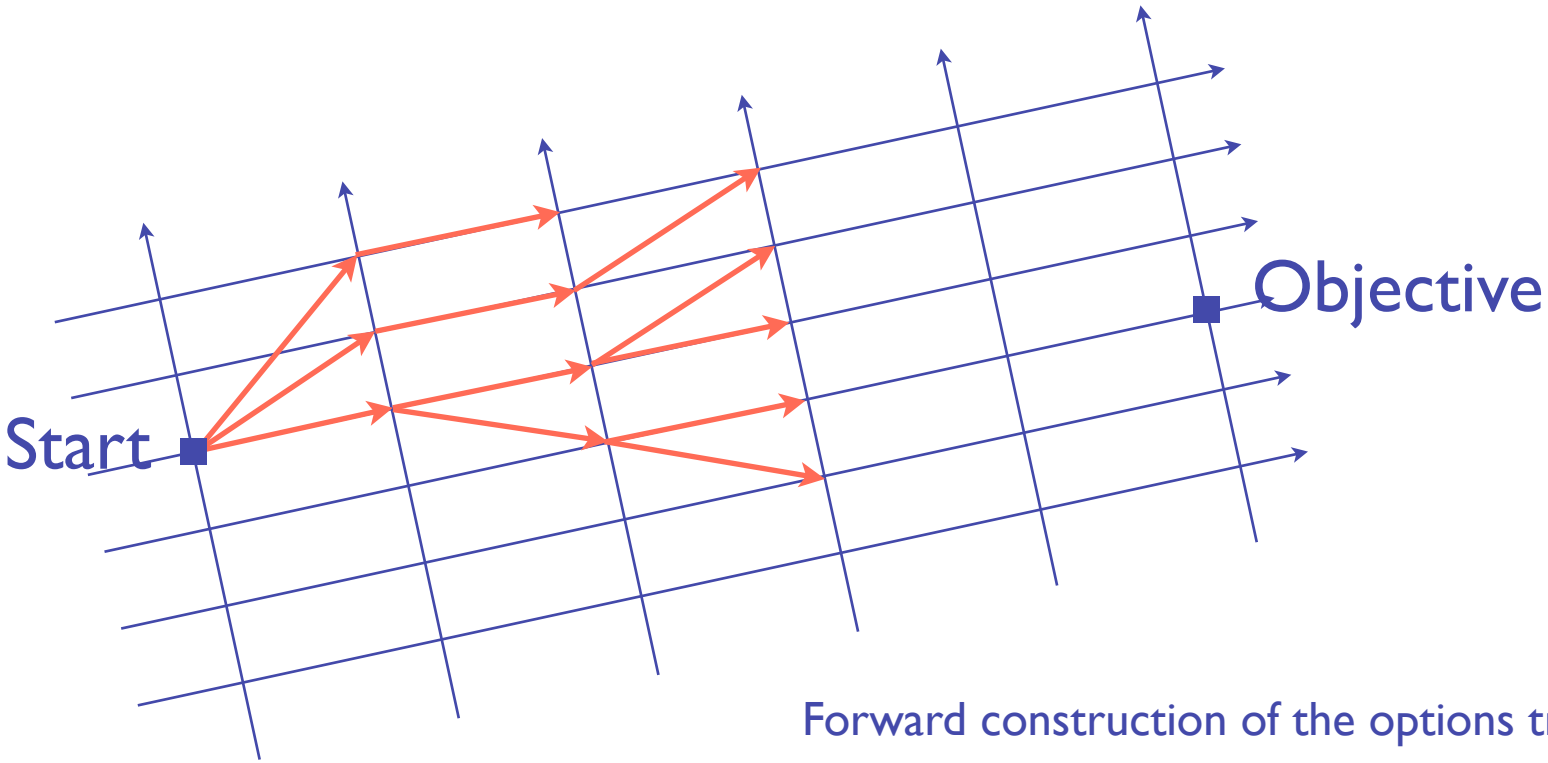
Definition of a network covering
the geographical area of the regatta.

Decision Scheme: Dynamical Programming



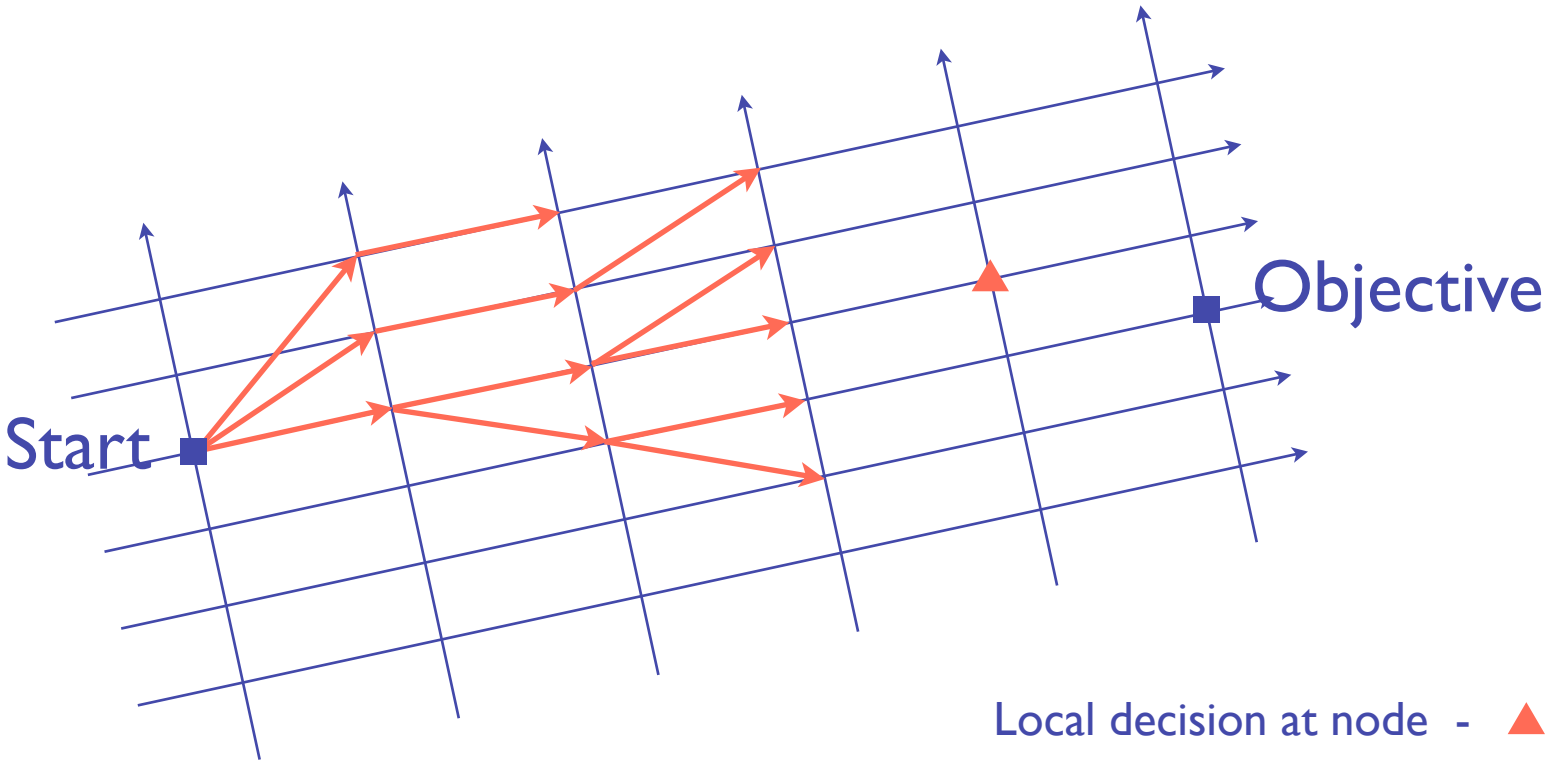
Forward construction of an options tree rooted at the starting point.

Decision Scheme: Dynamical Programming

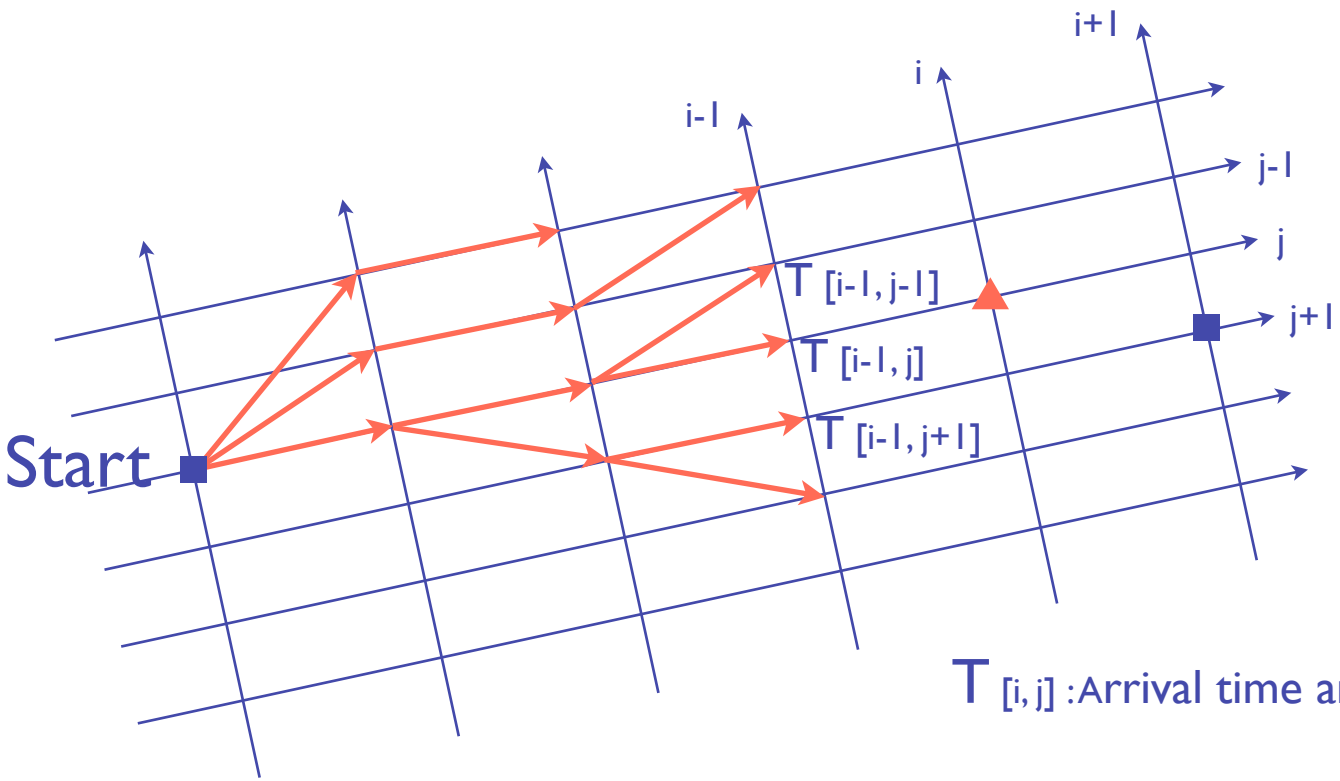


Forward construction of the options tree.

Decision Scheme: Dynamical Programming



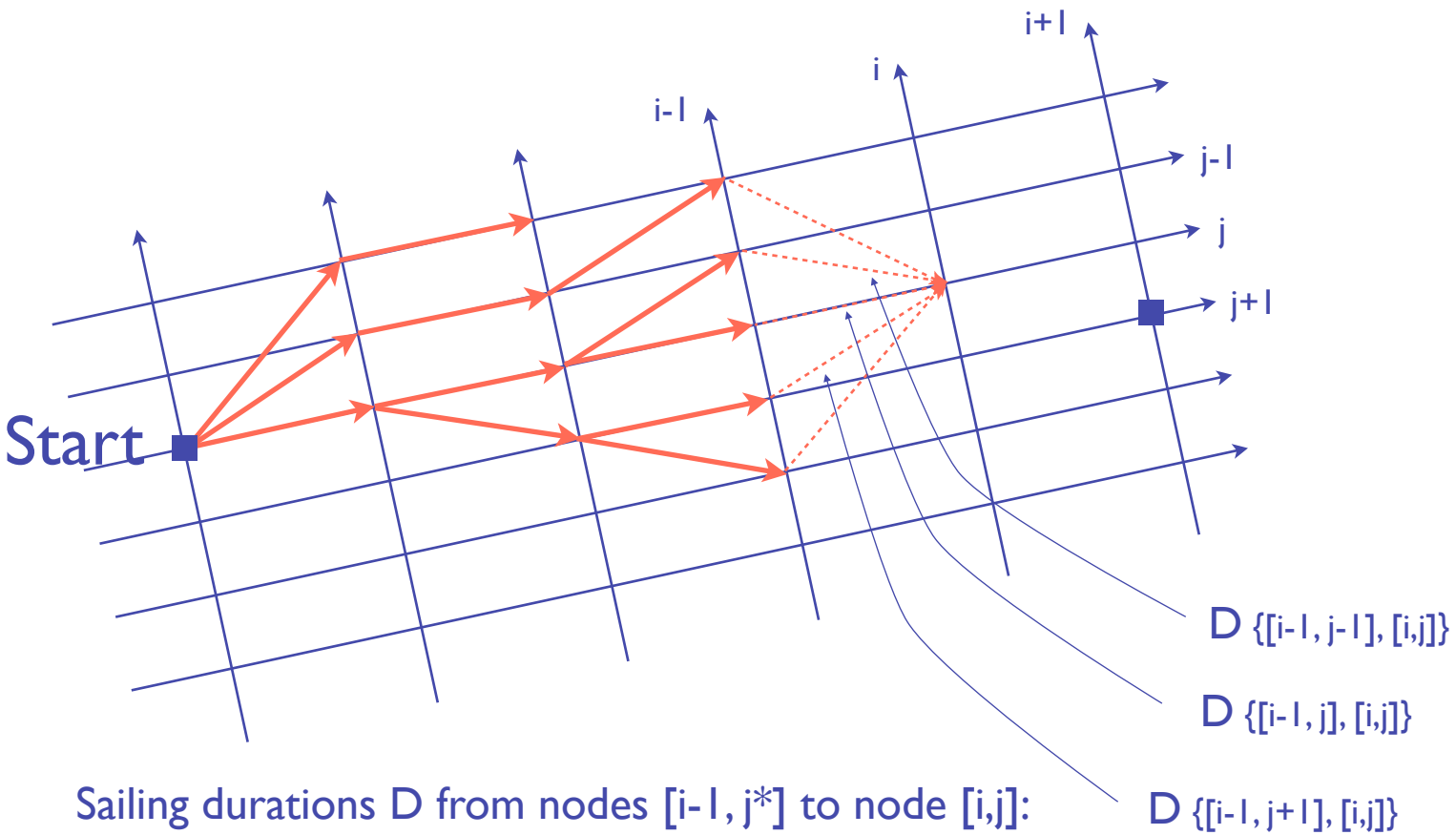
Decision Scheme: Dynamical Programming



$T_{[i,j]}$:Arrival time an node [i, j]

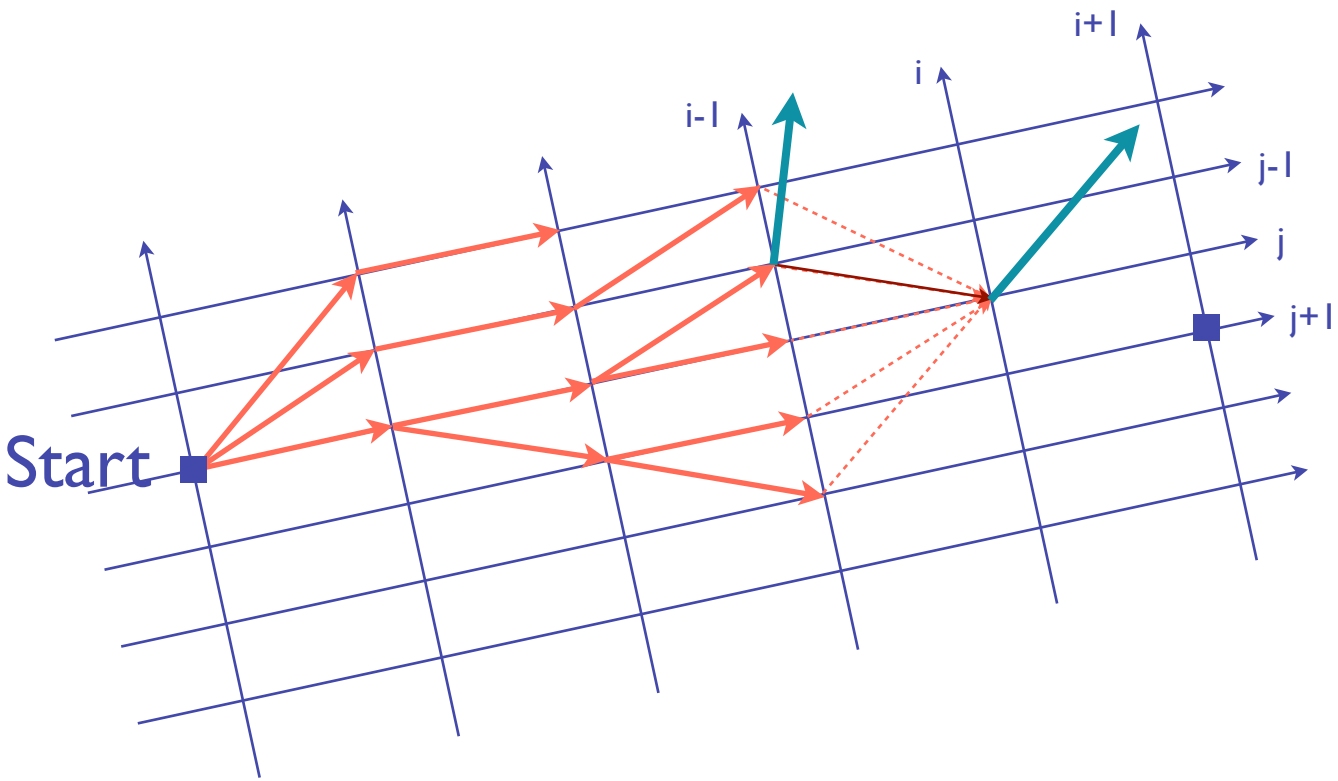
Thanks to the arborescent structure, the $T_{[i-1,j^*]}$ are (recursively) known ($T_{[Start]}$ provided)

Decision Scheme: Dynamical Programming



Sailing durations D from nodes [i-1, j*] to node [i,j]:

Decision Scheme: local computations

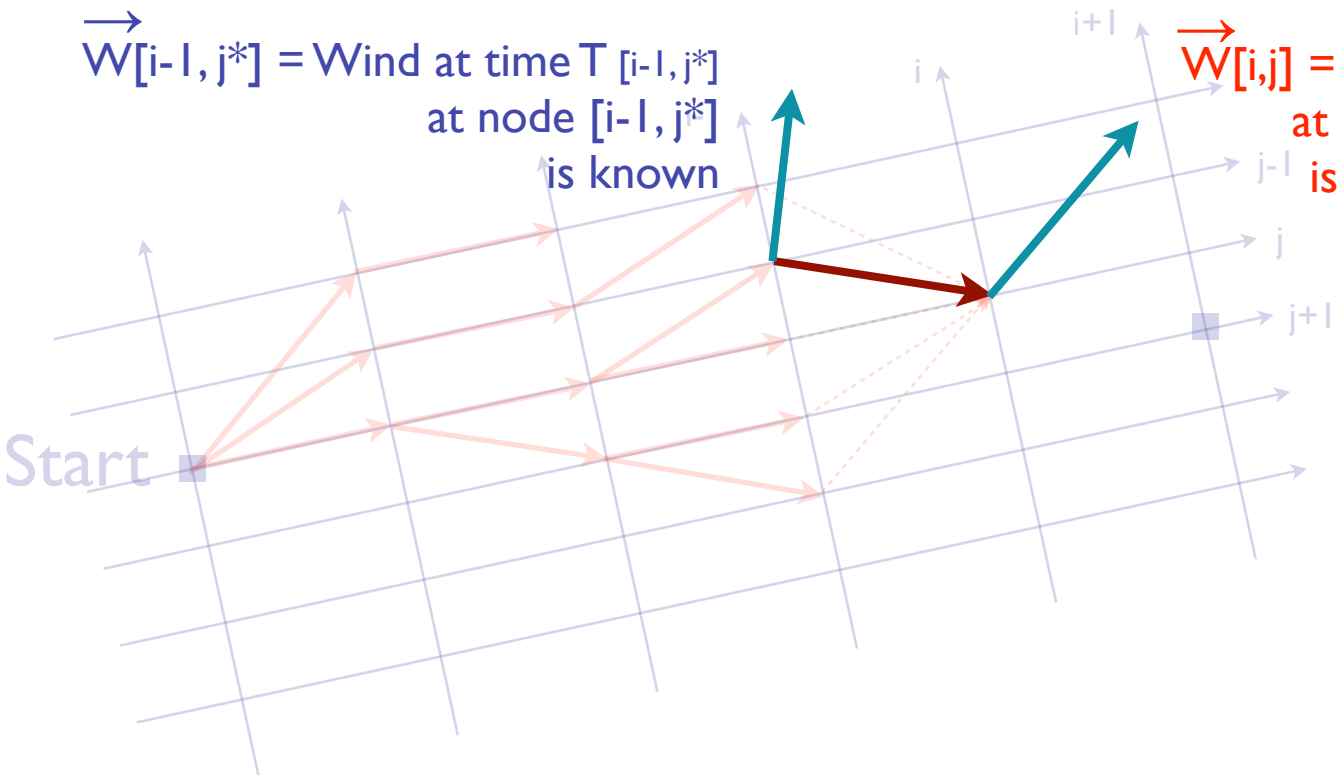


Sailing durations D from nodes [i-1, j*] to node [i,j] depend on **wind conditions** on each segment

Decision Scheme: local computations

$\overrightarrow{W}$[i-1, j*] = Wind at time T [i-1, j*] at node [i-1, j*] is known

$\overrightarrow{W}$[i,j] = Wind at node [i, j] at time T [i, j]= ?? is unknown !!

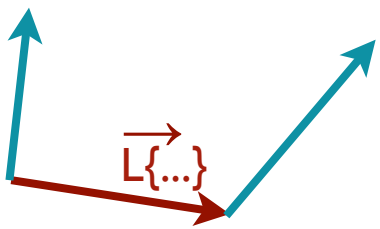


Sailing durations D from nodes [i-1, j*] to node [i,j] depend on **wind conditions** on each segment

Decision Scheme: local computations

$\overrightarrow{W}[i\text{-}1, j^*]$ = Wind at time T $[i\text{-}1, j^*]$ at node $[i\text{-}1, j^*]$ is known

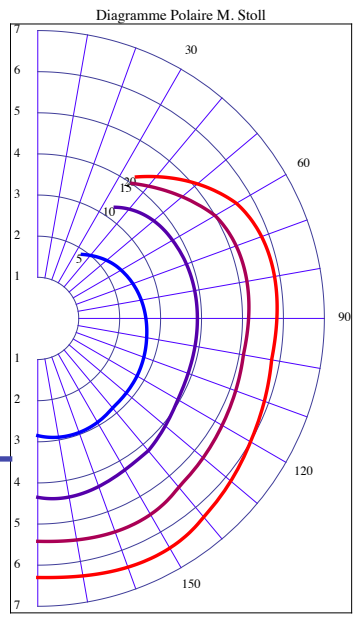
$\overrightarrow{W}[i,j]$ = Wind at node $[i, j]$ at time T $[i, j]$= ?? is unknown !!



Duration D from node $[i\text{-}1, j^*]$ to node $[i,j]$:

$$D\{[i\text{-}1, j^*], [i, j]\} = \frac{2 \left| \overrightarrow{L}\{[i\text{-}1, j^*], [i, j]\} \right|}{P\left(\overrightarrow{W}[i\text{-}1, j^*], \overrightarrow{L}\{...\}\right) + P\left(\overrightarrow{W}[i, j], \overrightarrow{L}\{...\}\right)}$$

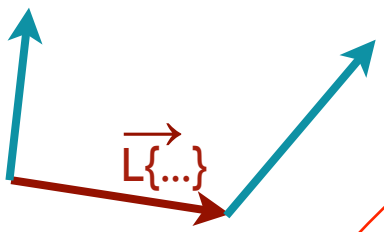
Polar Diagram - Customer Profile



Decision Scheme: local computations

$\overrightarrow{W}[i\text{-}1, j^*]$ = Wind at time T $[i\text{-}1, j^*]$ at node $[i\text{-}1, j^*]$ is known

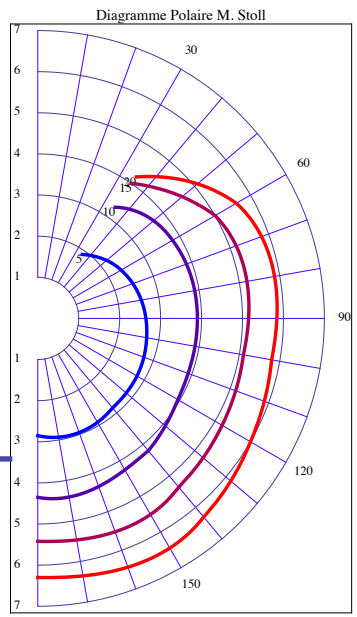
$\overrightarrow{W}[i,j]$ = Wind at node $[i, j]$ at time T $[i, j]$= ?? is unknown !!



Duration D from node $[i\text{-}1, j^*]$ to node $[i,j]$:

$$D\{[i\text{-}1, j^*], [i, j]\} = \frac{2 \left| \overrightarrow{L}\{[i\text{-}1, j^*], [i, j]\} \right|}{P\left(\overrightarrow{W}[i\text{-}1, j^*], \overrightarrow{L}\{...\}\right) + P\left(\overrightarrow{W}[i, j], \overrightarrow{L}\{...\}\right)}$$

Polar Diagram - Customer Profile



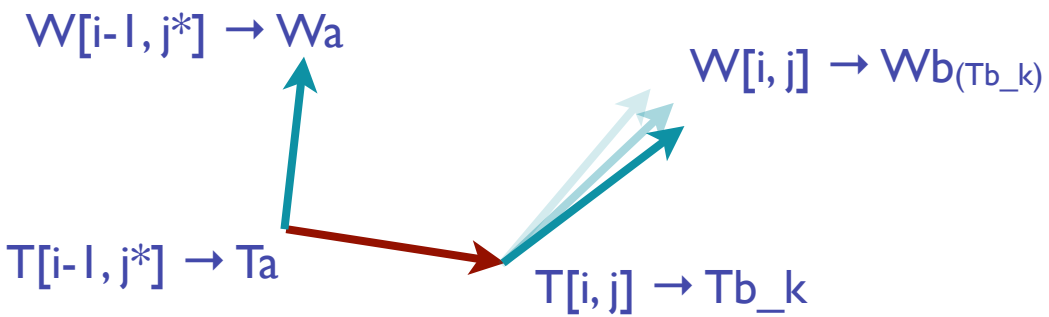
Circularity: Fixed Point Problem

- Duration D from node [i-1, j*] to arrival node [i,j] :
  depends on the wind W[i, j] blowing at that node at arrival time
  T[i-1, j*] + D {[i-1, j*], [i, j]}
- Valuation of wind W[i, j] at node [i, j] :
  depends on duration D {[i-1, j*], [i, j]}
- Formally:

$$W[i,j] = \varphi\left(T[i-1, j^*] + D\{[i-1, j^*], [i, j]\}_{(W[i,j])} \right)$$

- Fixed Point Problem $x = \varphi(x)$ (Brower, Lefschetz and all ...)

Circularity: iterative solution



$Tb\_0 = Ta + D[Wa, Wb_{(Ta)}] \rightarrow$

$Tb\_1 = Ta + D[Wa, Wb_{(Tb\_0)}] \rightarrow$

$Tb\_2 = Ta + D[Wa, Wb_{(Tb\_1)}] \rightarrow$

...

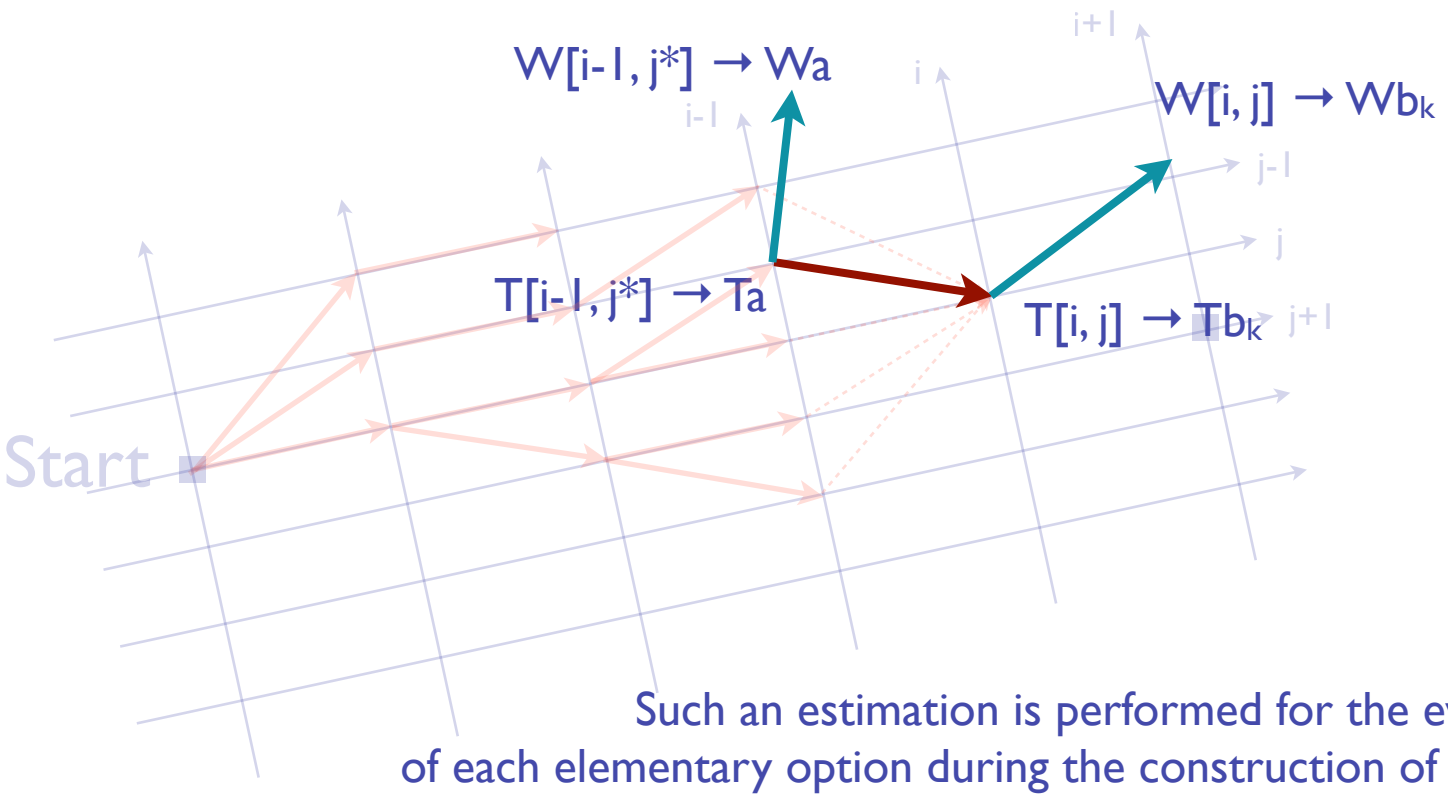
$Tb\_i = Ta + D[Wa, Wb_{(Tb\_i-1)}] \rightarrow$

... ... →

Implemented:
i = 0, ..., 4

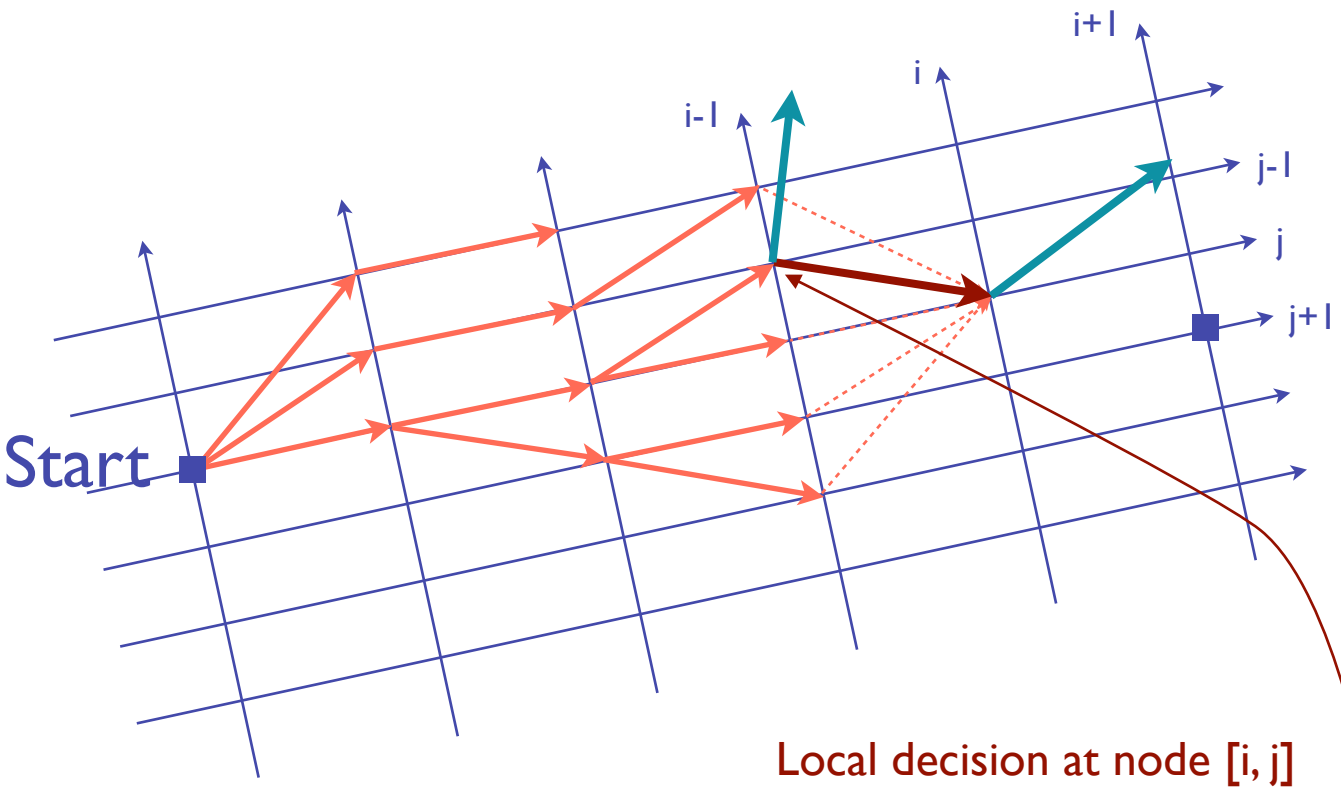
Convergence:
fixed points theory

Circularity: implementation after temporal recursion



Such an estimation is performed for the evaluation of each elementary option during the construction of the tree.

Decision Scheme: Dynamical Programming

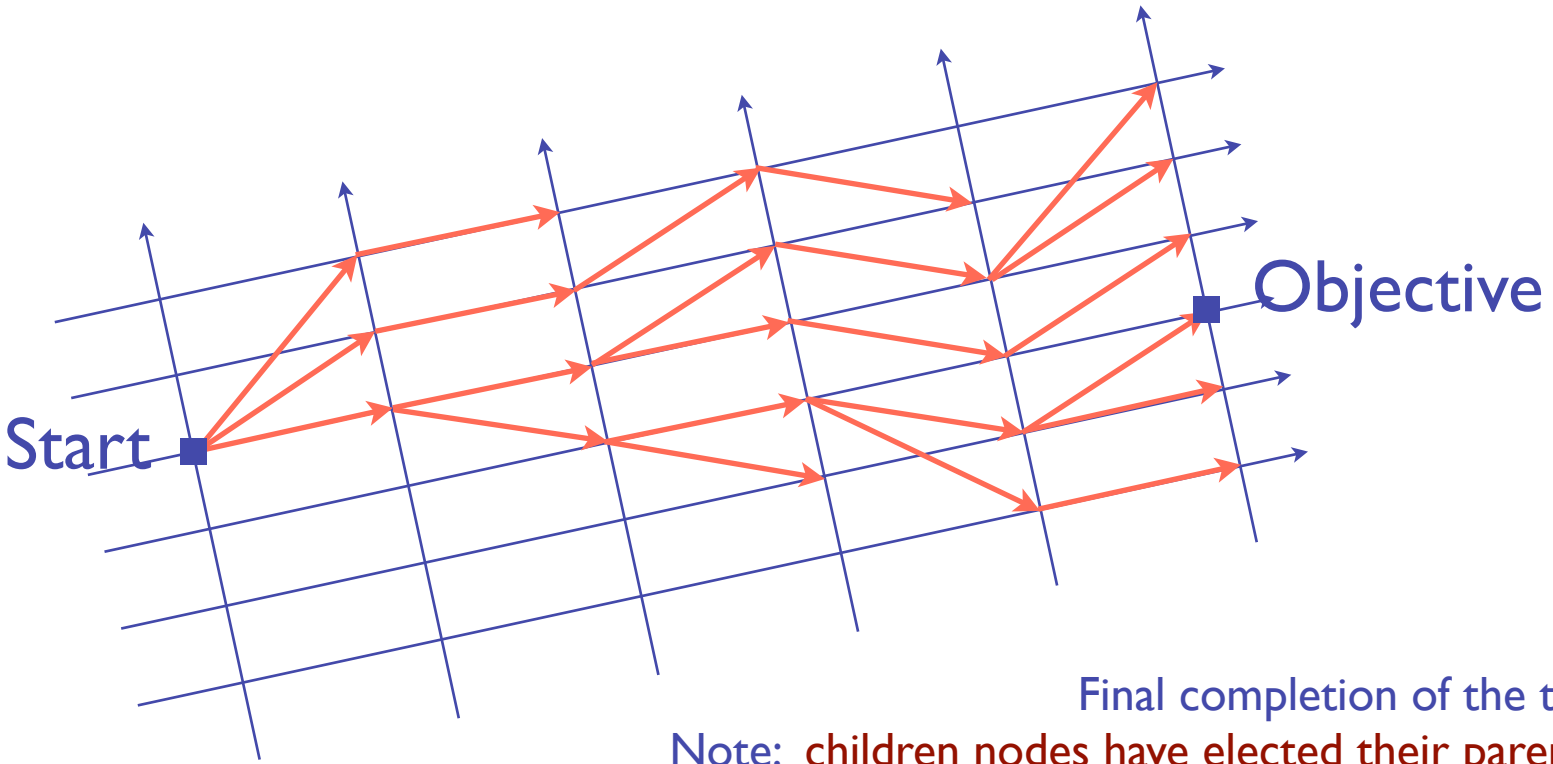


Local decision at node [i, j]

Minimum $\{T_{[i-1, j^*]} + D_{\{[i-1, j^*],[i, j]\}}\}$ → parent node [i-1, j^]

j*

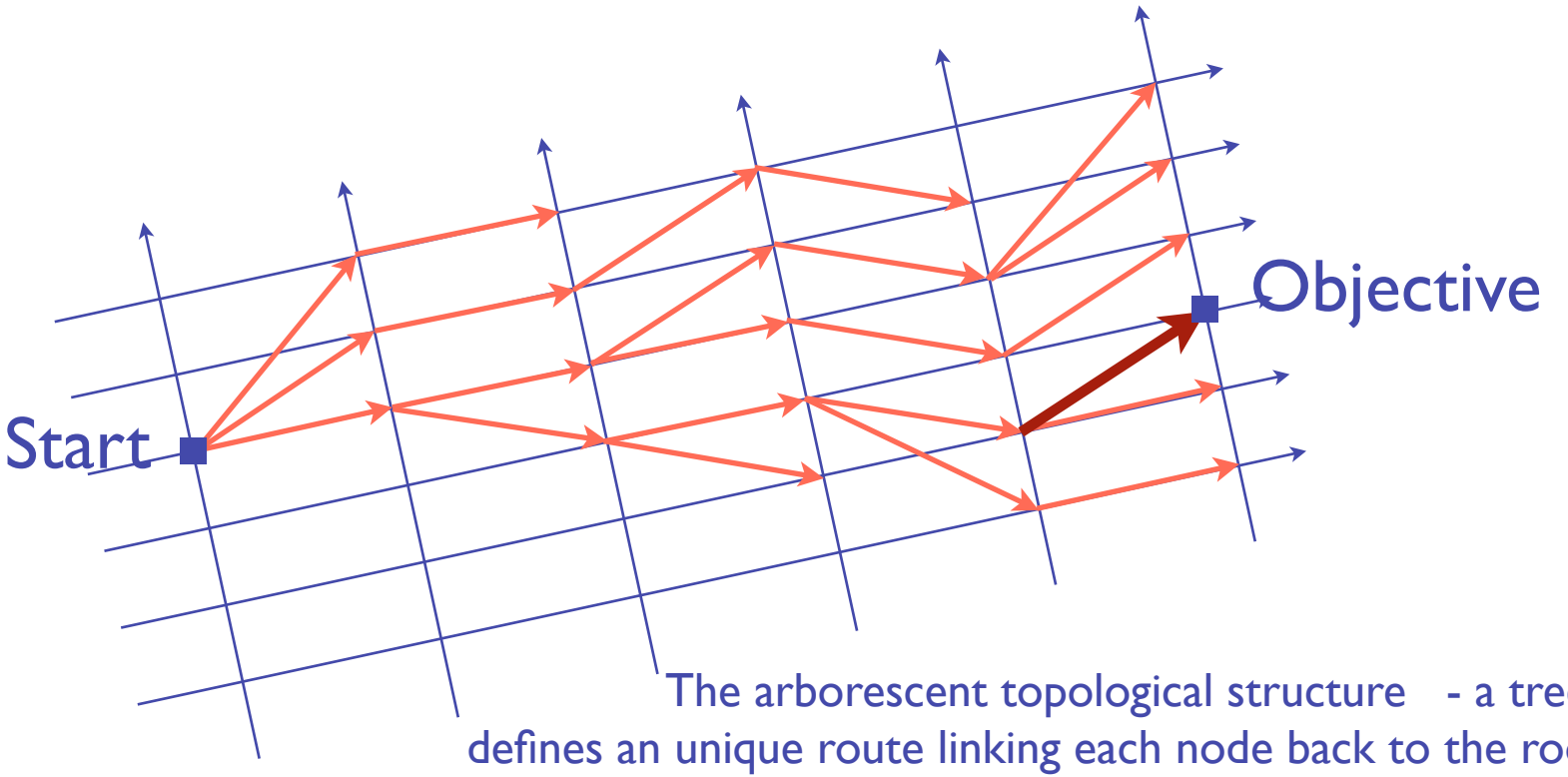
Decision Scheme: Dynamical Programming



Final completion of the tree

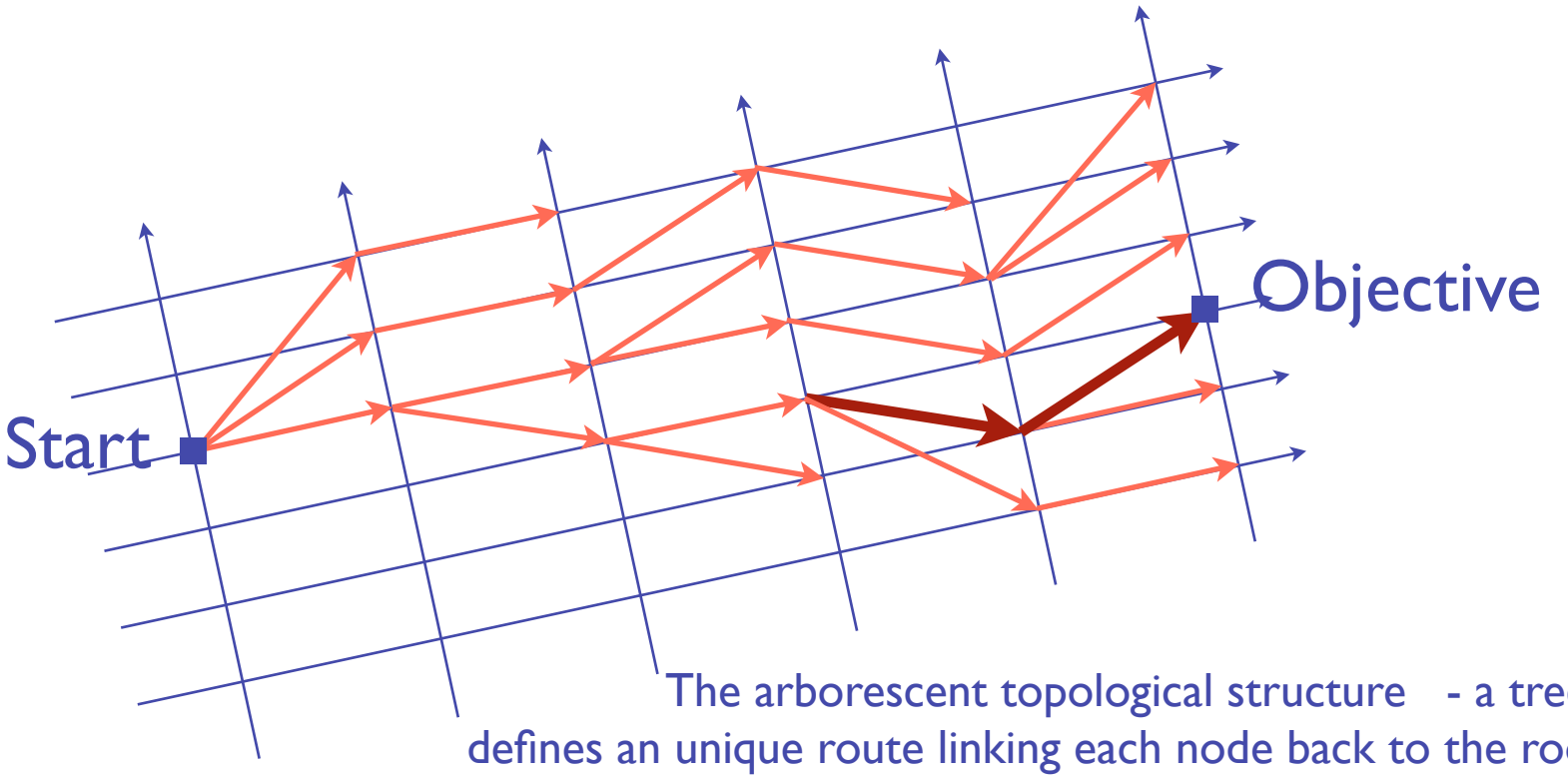
Note: children nodes have elected their parents!

Decision Scheme: Dynamical Programming



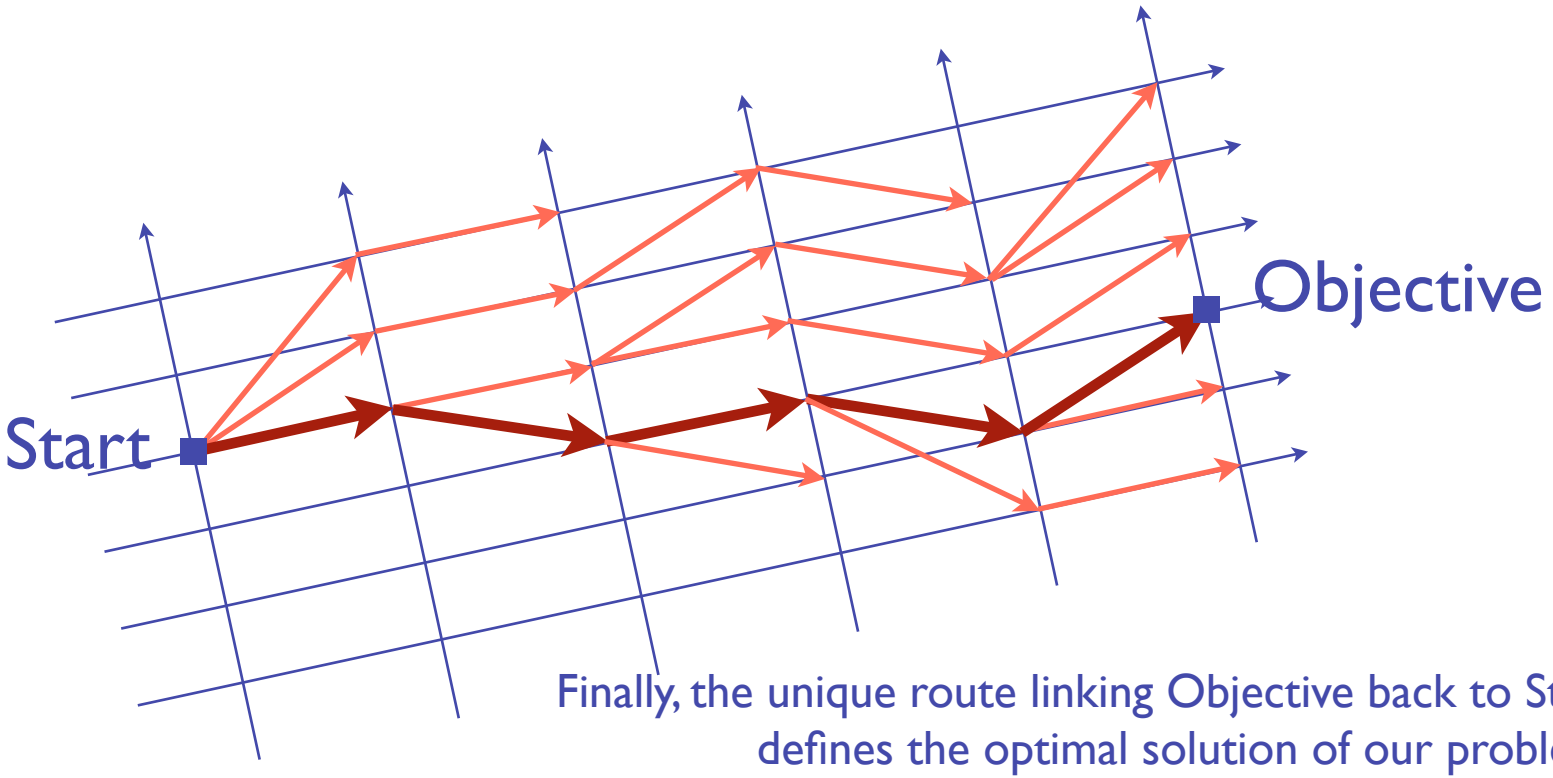
The arborescent topological structure - a tree - defines an unique route linking each node back to the root.

Decision Scheme: Dynamical Programming



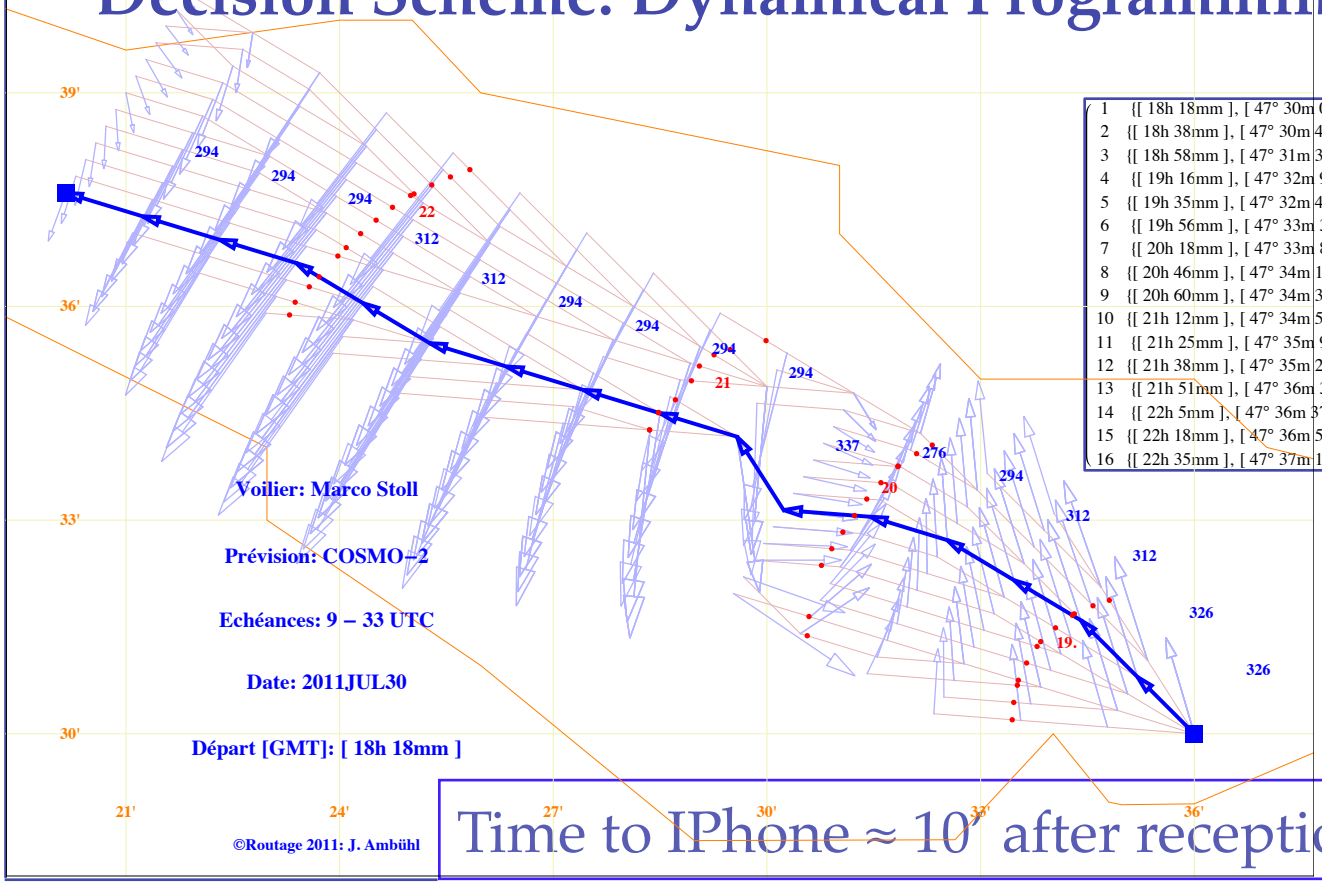
The arborescent topological structure - a tree - defines an unique route linking each node back to the root.

Decision Scheme: Dynamical Programming



Finally, the unique route linking Objective back to Start defines the optimal solution of our problem.

Decision Scheme: Dynamical Programming



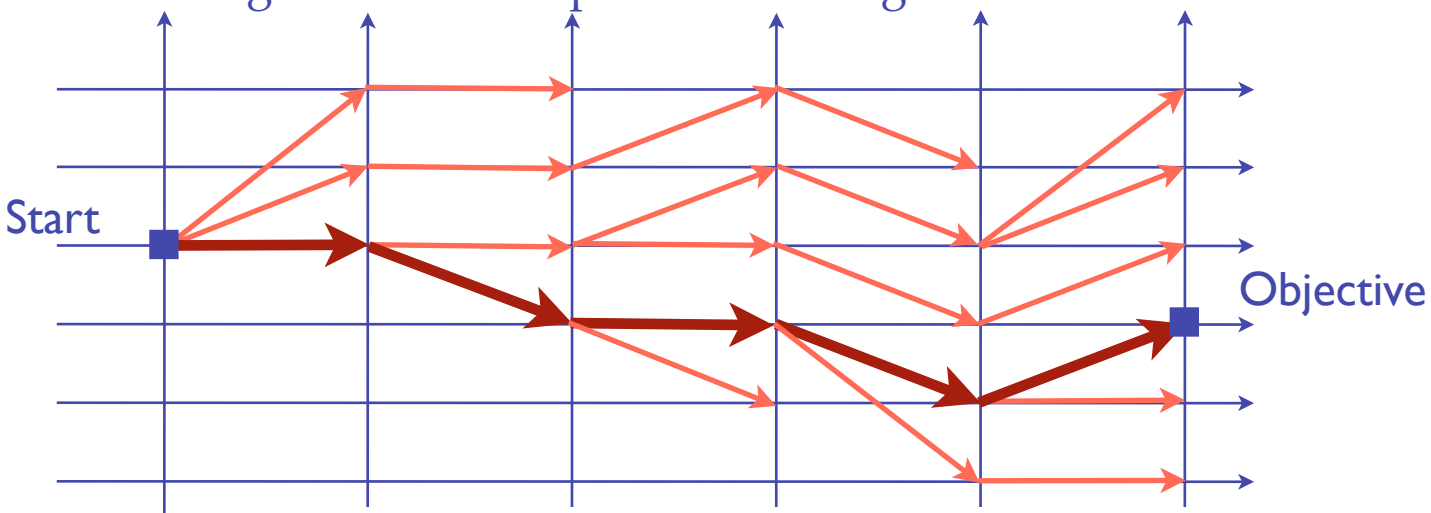
1 {[18h 18mm], [47° 30m 0s], [9° 35m 60s], [Kompass: 326°], [Speed: 2.87698 Kt]}
2 {[18h 38mm], [47° 30m 48s], [9° 35m 12s], [Kompass: 326°], [Speed: 2.93191 Kt]}
3 {[18h 58mm], [47° 31m 36s], [9° 34m 24s], [Kompass: 312°], [Speed: 2.81867 Kt]}
4 {[19h 16mm], [47° 32m 9s], [9° 33m 28s], [Kompass: 312°], [Speed: 2.57827 Kt]}
5 {[19h 35mm], [47° 32m 43s], [9° 32m 32s], [Kompass: 294°], [Speed: 2.32491 Kt]}
6 {[19h 56mm], [47° 33m 3s], [9° 31m 27s], [Kompass: 276°], [Speed: 2.30476 Kt]}
7 {[20h 18mm], [47° 33m 8s], [9° 30m 14s], [Kompass: 337°], [Speed: 2.38657 Kt]}
8 {[20h 46mm], [47° 34m 10s], [9° 29m 35s], [Kompass: 294°], [Speed: 3.44981 Kt]}
9 {[20h 60mm], [47° 34m 30s], [9° 28m 30s], [Kompass: 294°], [Speed: 3.78637 Kt]}
10 {[21h 12mm], [47° 34m 50s], [9° 27m 25s], [Kompass: 294°], [Speed: 3.71636 Kt]}
11 {[21h 25mm], [47° 35m 9s], [9° 26m 21s], [Kompass: 294°], [Speed: 3.77567 Kt]}
12 {[21h 38mm], [47° 35m 29s], [9° 25m 16s], [Kompass: 312°], [Speed: 3.84366 Kt]}
13 {[21h 51mm], [47° 36m 3s], [9° 24m 20s], [Kompass: 312°], [Speed: 3.75565 Kt]}
14 {[22h 5mm], [47° 36m 37s], [9° 23m 23s], [Kompass: 294°], [Speed: 3.46677 Kt]}
15 {[22h 18mm], [47° 36m 56s], [9° 22m 19s], [Kompass: 294°], [Speed: 2.96467 Kt]}
16 {[22h 35mm], [47° 37m 16s], [9° 21m 14s], [Kompass: 294°], [Speed: 2.20838 Kt]}

Log-book

Time to IPhone ≈ 10' after reception of GPS-position

Synthesis

1. Seeking forward for options: building the tree

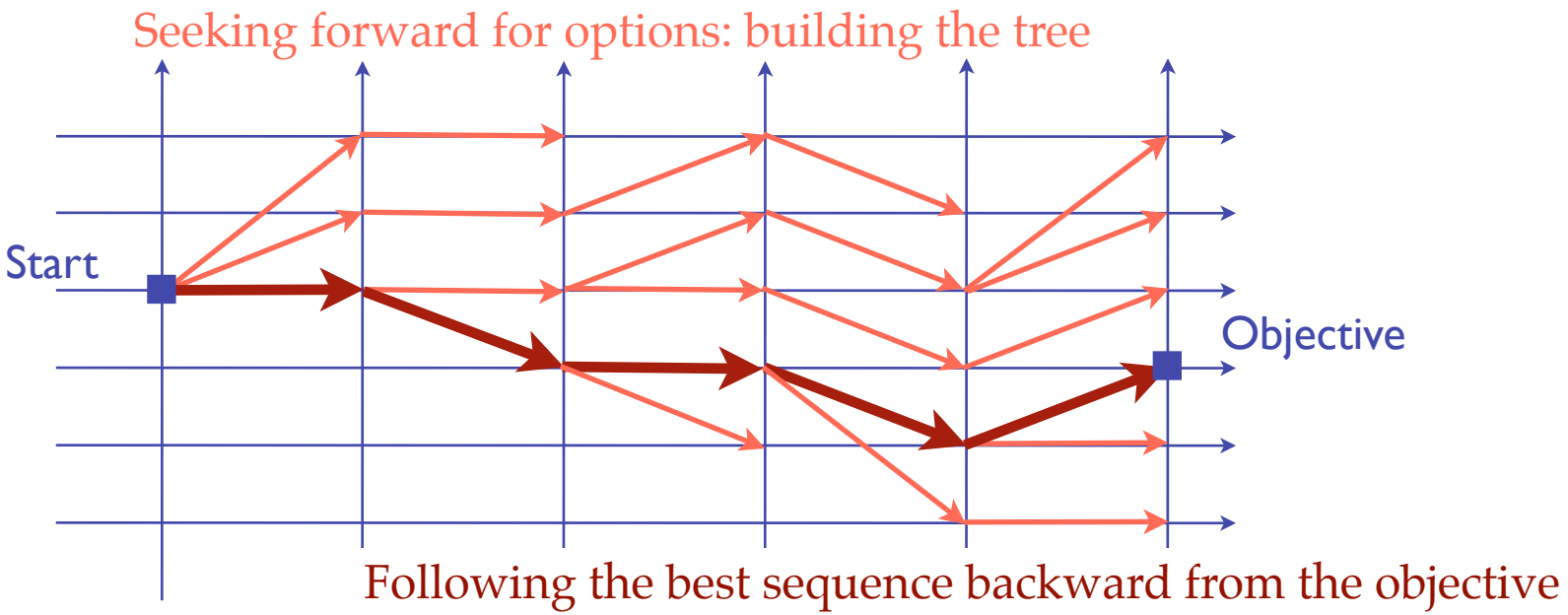


2. Following the best sequence backward from the objective, as provided by the tree

Classical technique in (financial) options trading

Alinghi, America Cup (EPFL, Prof. Dalang)

Synthesis



Optimization of weather dependent sequential processes

Renewable Energies!

Erstellung des Polardiagramms

Polardiagramm

Entscheidungskern

Entscheidungs-baum

Erstellung des Entscheidungsnetzes

Entscheidungsnetz

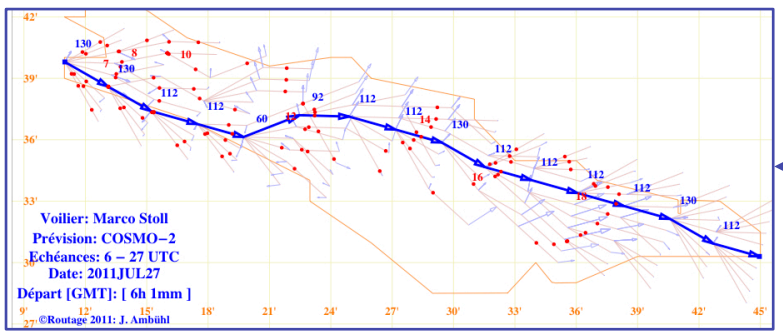
Windfeld aus COSMO

Interpolation aufs E-Netz

Windfeld auf E-Netz

Optimale Route

Allgemeine Darstellung



Log Buch

Erstellung des Polardiagramms

Polardiagramm

Entscheidungskern

Entscheidungs-baum

Erstellung des Entscheidungsnetzes

Entscheidungsnetz

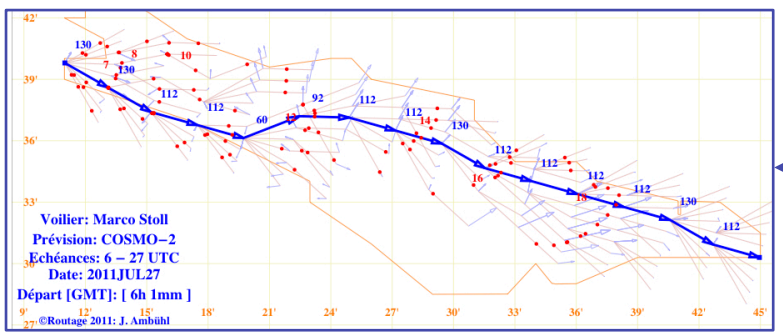
Aufgaben 1 + 2

Windfeld aus COSMO

Interpolation aufs E-Netz

Windfeld auf E-Netz

Optimale Route



Allgemeine Darstellung

Log Buch

Aufgabe 3

Erstellung des Polardiagramms

Polardiagramm

Erstellung des Entscheidungsnetzes

Entscheidungsnetz

Entscheidungskern

Entscheidungs-baum

Windfeld aus COSMO

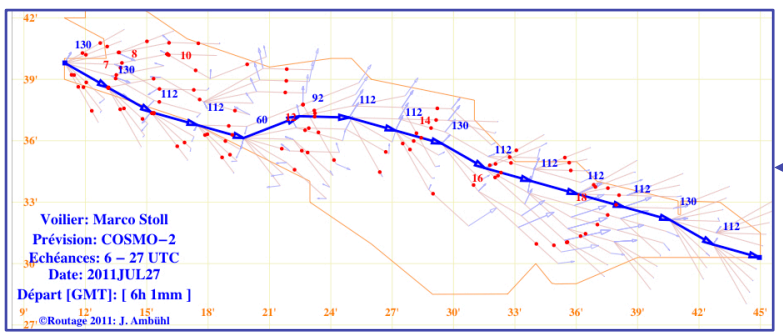
Interpolation aufs E-Netz

Windfeld auf E-Netz

Optimale Route

Allgemeine Darstellung

Log Buch



Erstellung des Polardiagramms

Polardiagramm

Entscheidungskern

Entscheidungs-baum

Erstellung des Entscheidungsnetzes

Entscheidungsnetz

Windfeld aus COSMO

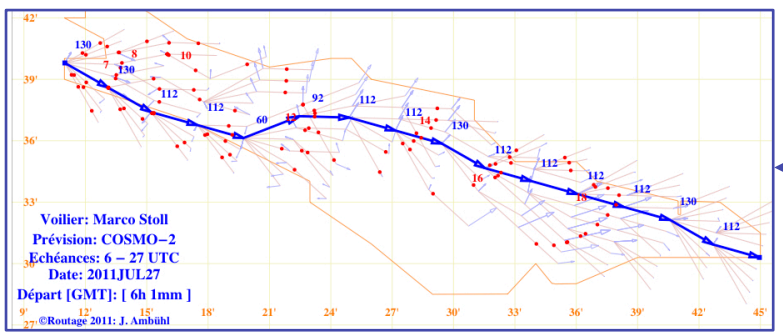
Interpolation aufs E-Netz

Aufgabe 4

Windfeld auf E-Netz

Optimale Route

Allgemeine Darstellung



Log Buch

Erstellung des Polardiagramms

Polardiagramm

Erstellung des Entscheidungsnetzes

Entscheidungsnetz

Entscheidungskern

Entscheidungs-baum

Windfeld aus COSMO

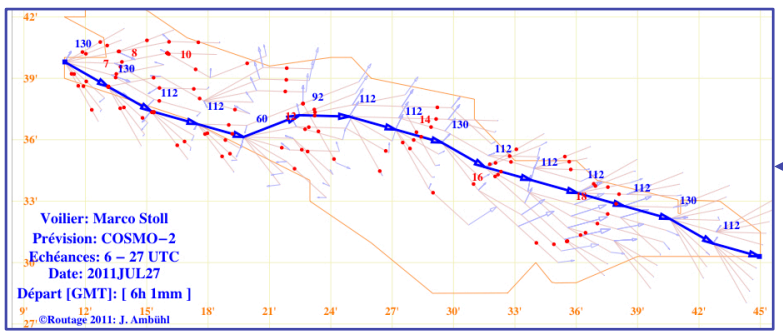
Interpolation aufs E-Netz

Aufgabe 5

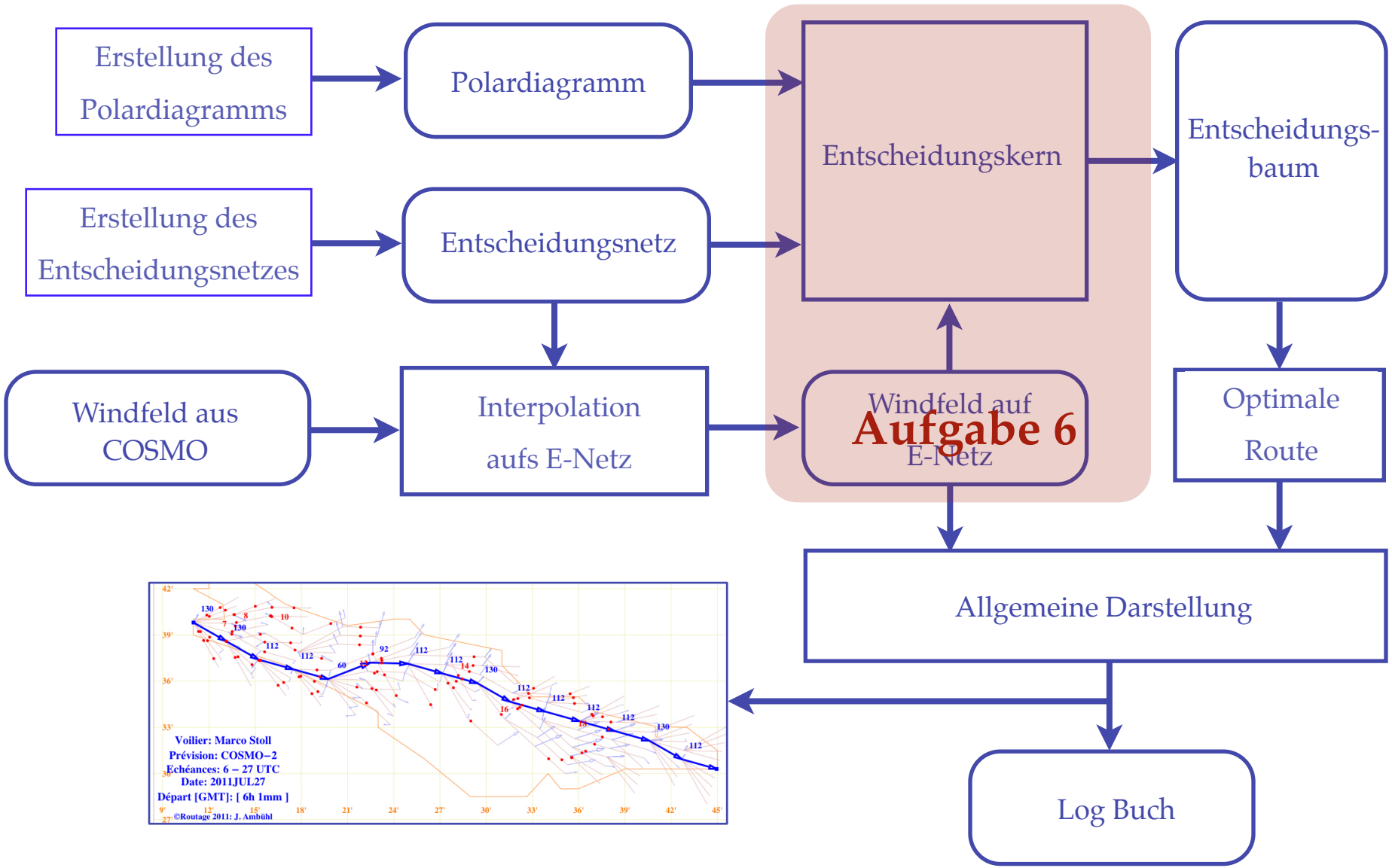
Windfeld auf E-Netz

Optimale Route

Allgemeine Darstellung



Log Buch



Erstellung des Polardiagramms

Polardiagramm

Entscheidungskern

Entscheidungs-baum

Erstellung des Entscheidungsnetzes

Entscheidungsnetz

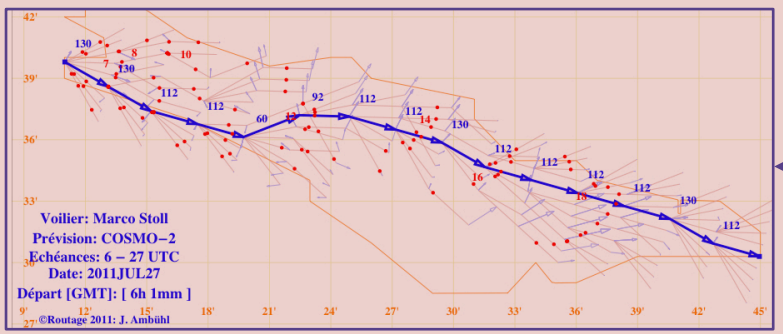
Windfeld aus COSMO

Interpolation aufs E-Netz

Windfeld auf E-Netz

Optimale Route

Allgemeine Darstellung



Aufgabe 7

Log Buch