Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Home Affairs FDHA
Federal Office of Meteorology and Climatology MeteoSwiss

Scientific Report MeteoSwiss No. 98

Sailing the virtual Bol d'Or

Jacques Ambühl

ISSN: 1422-1381

Scientific Report MeteoSwiss No. 98

Sailing the virtual Bol d'Or

Jacques Ambühl

Recommended citation:

Ambühl, J: 2014, Sailing the virtual Bol d'Or, *Scientific Report MeteoSwiss,* **98,** 44 pp.

Editor:

Federal Office of Meteorology and Climatology, MeteoSwiss, © 2014

Abstract

This project is aimed at determining the course of a sailing boat on a lake, enabling the fastest journey from a departure to an arrival point.
The frame of the lake, the wind field on water surfaces and its temporal evolution, as well as the sailing boat, are subprocesses integrated in a decision scheme based on dynamic programming. The wind field is provided by the COSMO-2 weather forecasting model operated at MeteoSwiss.
The sailing duration is considered an abstract cost having to be minimized. Departure and arrival points are the initial and final conditions for the decision scheme.

Acknowledgement

My thanks are addressed to Marco Stoll, weather forecaster by MeteoSwiss Zürich and sailor on the Bodensee, Olivier Codeluppi, forecaster, Daniel Leuenberger, numerical meteorologist, both by MeteoSwiss, Oliver Fuhrer, atmospheric scientist at ETHZ and Philippe Jeanneret, Television Weatherman in Geneva and sailor on the lake Léman.

Furthermore, I express my gratitude to MeteoSwiss for having put COSMO-2 data at my disposal for the project.

.

Contents

1 Introduction

The aim of the project is easily formulated: determine the course of a sailing boat on a lake, enabling the speediest journey from a departure to an arrival point.

The frame of the lake, the wind field on water surfaces and its temporal evolution, as well as the sailing boat, are subprocesses integrated in a decision scheme based on Dynamic Programming. The sailing duration is considered an abstract cost having to be minimized. Departure and arrival points are the initial and final conditions for the decision scheme.

Figure 1 exhibits the expected outcome: the optimal route (bold blue line) of a sailing boat on Lake Léman[1]. The start occurs at 9 h. local time (7 UTC) in front of Geneva. The objective is a turning buoy located at the opposite end of the lake, in front of Le Bouveret.

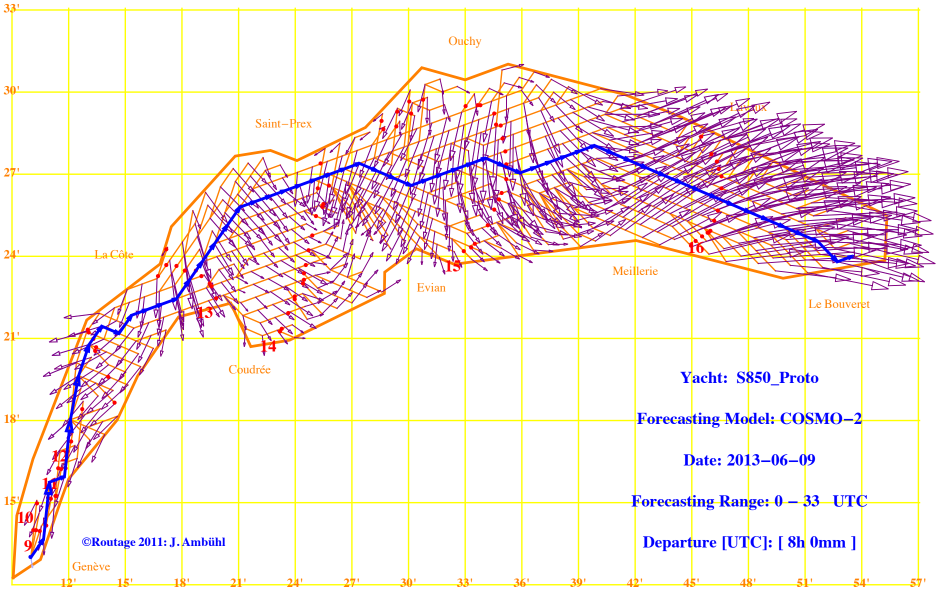

Figure 1: *Optimal route of a boat sailing from Geneva to the Bouveret on 9 June 2013. Numerical Weather Forecasting Model: COSMO 2*

[1] Geneva Lake

The algorithm delivers:

- a decision tree, sketched in the background in orange, growing from the departure towards the arrival point,
- the wind vector at each vertex of the tree at the time at which the sailing boat would cross that vertex, and
- the isochrones, sketched as curves of red dots laid almost perpendicularly to the main route. They exhibit the progression throughout the diagram.

The performance of the sailing boat is provided by its polar diagram, described in chapter 4. The wind field is delivered by the COSMO-2 Weather Forecasting Model operated at MeteoSwiss [1]. As on-line routing is forbidden during the regatta, option charts similar to Figure 1 are provided to the skipper before the starting gun is fired. One chart covers the way forth from Geneva to the Bouveret, the second chart covers the way back. The departure time of the latter is provided by the arrival time of the former.

1.1 A short tale about routing

Everything began in 1984. Pierre Fehlmann, the epochal swiss sailor, was preparing the Whitbread Round the World Race 1985-86. Bruce Farr, a famous naval architect, had been mandated to design the maxi boat "UBS-Switzerland" then under construction in Morges at the shipyard Decision, held and managed by Bertrand Cardis. EPFL, the Ecole Polytechnique Fédérale de Lausanne, had offered computing time on its cutting-edge CDC-Cyber machine.

Fehlmann, already a seasoned high sea sailor, visited once a young meteorologist and applied mathematician, the author, and asked him wether he had "a trick to cope with the odds of the weather in the tropical and equatorial zone". An active gliding instructor, the young scientist knew another tale, where Paul MacCready had won the 1954 world gliding championships after having designed and used a self made analog computer[2] that took into account the potential of the weather and the performance of his glider, provided as a polar diagram. MacCready's example would be seminal in our meteorologist's mind. He had spend valuable time at the European Center for Medium Range Weather Forecast, ECMWF [4], in Reading (UK) and

[2] A circular slide rule mounted on the variometer, on the instrument panel of the glider.

knew the potential of the brand new "T-106 spectral, global weather forecasting model", whose forecasts were becoming available for applications. Provided that the computing power would be available, the telecommunication would work (Internet was just dawning), then, making use of an optimization technique called Dynamical Programming, it would be possible to compute optimal sailing options all around the world.

Together with Pierre Eckert, Bertrand Cardis, and the valuable help of EPFL and MeteoSwiss[3], the small team managed to transform the dream into a comprehensive computing, telecommunication and decision system that reached operational status in summer 1985. The outcome happened to be unmissable: UBS-Switzerland, Pierre Fehlmann and his crew won the Round the World Race 1985-1986 in real time [7,8].

As active routing was forbidden during the Race, climatic analysis and decision schemes were provided to the skipper before the start. Such a Map is presented in Figure 2. Later, real time algorithms similar to the system presented in this report were developed for regattas where routing was allowed. Both Merit 1 and 2 won several such regattas in the beginning of the nineties [9].

By the mid nineties, Bertrand Piccard took contact with Pierre Eckert and the author in order to envisage a ballon flight around the planet. Eckert took the lead. Making use of the know-how acquired so far, he developed similar algorithms, this time tailored for a ballon. Eckert's simulations enabled Piccard firstly to convince potential sponsors of the feasibility of such a flight, then to design the ballon[4], a duty taken over by EPFL, and finally to fly. After two failed attempts, taking off from Chateau-d'Oex on 1 Mars 1999, Piccard succeeded in his round the world flight, together with his copilot Brian Jones.

At that time, a culture of scientific adventure had settled in western Switzerland. Academics, sailors, aviators, were bound together and eager to open new horizons. Of course, the America Cup had laid all the time on the horizon line of those audcious guys, however, at unspeakable distance. In this respect, Alinghi happened to be a kind of coronation. Prof. Dalang, F.

[3] At that time still named Swiss Federal Institute for Meteorology.

[4] Mostly the thermo-dynamical properties of the ballon.

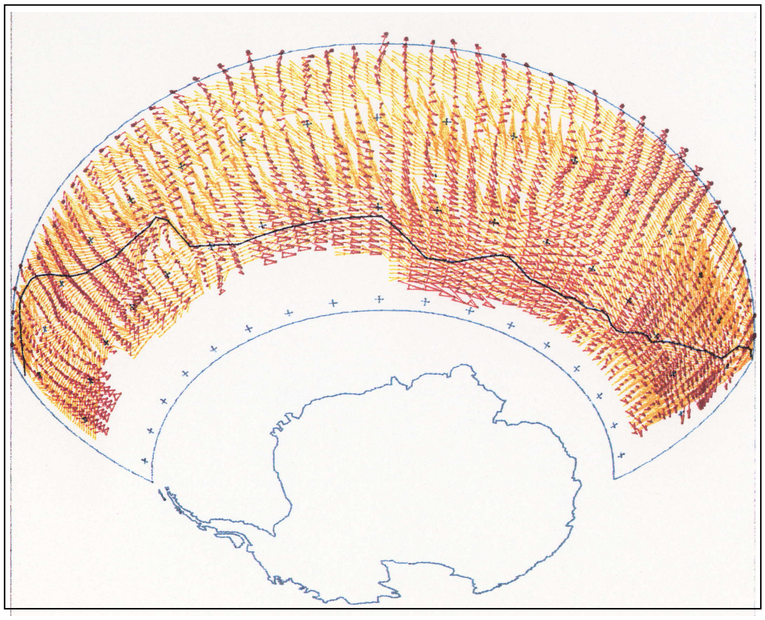

Figure 2: *3-D projection of a simulation of a regatta on the track Punta del Este (Uruguay) - Perth (Australia), 1992. The sailing boat is Merit 2 and the continent is Antarctica. [9].*

Dumas and L. Winckenbosch developed a decision scheme based on Stochastic Dynamical Programming that would optimize decision schemes under the straight laced rules of the America Cup. The consequence is known: Ernesto Bertarelli and his crew brought the cup back to Europe in 2003 and defended it successfully in Valencia in 2007 [11].

1.2 Lake routing

On a more humble footing, the idea that a routing algorithm could once be made available for a lake, and no longer on an ocean, had been alive for all the years. The inconspicuous project presented here was started in 2011 on private basis together with Marco Stoll, a meteorologist at MeteoSwiss. The COSMO-2 model of MeteoSwiss, now routinely operated with an horizontal grid-spacing of 2.2 km, provided the kick. The physics of current local models and their parametrization lay light years away from those of the models currently used in the eighties. A COSMO-1 version, tuned to run at 1 km, is in development and should be routinely operated by 2015.

Not only the computing power had increased during the years, modern software development tools enabled us to design algorithms much more accurate than those we used in the eighties. They are briefly sketched in the present report. Some of them have been translated in the Java programming language within the framework of a bachelorthesis at the Zürcher Hochschule für angewandte Wissenschaften [12].

This report is structured as follow: chapter 2 explains Dynamical Programming with the help of a simple example. Wind forecasts are discussed in chapter 3. The Polar Diagram of a sailing boat is introduced in chapter 4. Spherical navigation is introduced in chapter 5. Chapter 6 is devoted to the estimation of the time elapsed on one arc of the network. The integration of the sub-systems is briefly sketched in chapter 7. The application of the system to a Bol d'Or regatta is then presented and discussed in chapter 8. Conclusions and outlook are provided in chapter 9.

2 Dynamical Programming

2.1 A first glance

Belonging to the realm of applied mathematics, dynamical programming (DP) consists in defining a sequence of decisions taking into account progress done so far, as well as future opportunities. Invented shortly after the second World War, Dynamical Programming has blossomed into a vast array of fields, from logistics to finance. Dynamical Programming deals with a System, that can for example be a portfolio of shares, an intelligent gird fed by sources of renewable energy or, in our case, a sailing boat on a lake.
Figures 3 and 4 give a glance of the method. The state of the system is transformed according to a sequence of decisions, where each decision i triggers a transition from a previous State $i-1$ to a next State i of the system. States and allowed transitions are specific to the system under consideration and, of course, decisions shall only trigger allowed transitions. Finally, each decision induces a cost labelled $Cost\ i$.

The initial state of the system, labelled $i = 0$ is known. The final state, labelled $i = N$, represents the goal having to be reached. The last transition of the system, the last decision to be taken, is settled: from any state $N-1$, reach state N.

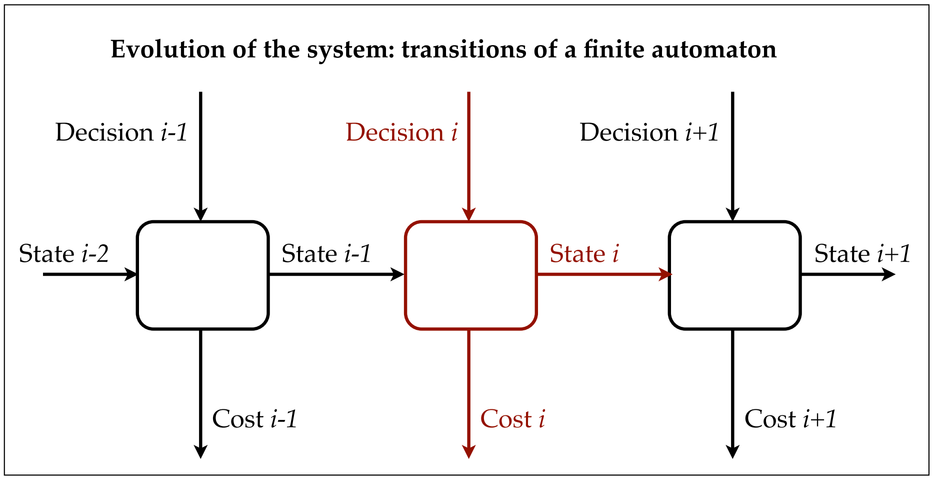

Figure 3: *Dynamical programming described as a sequence of decisions acting on a system arbitrarily sketched as a box.*

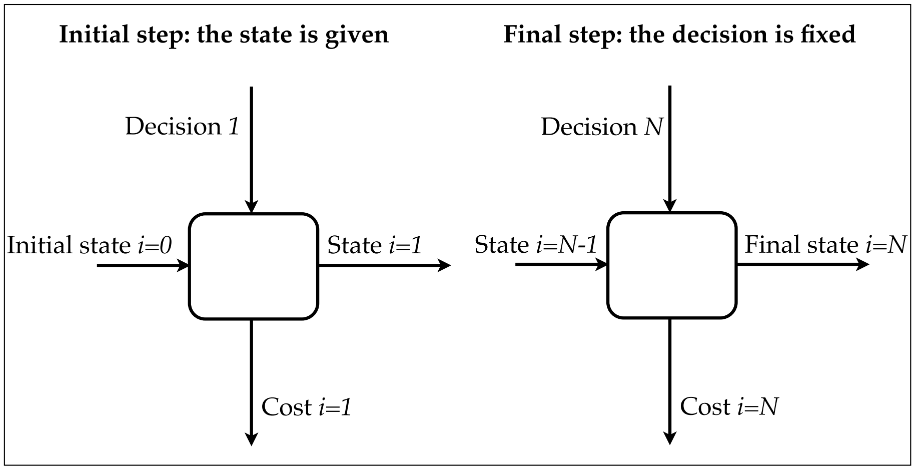

Figure 4: *Dynamical programming described as a sequence of decisions. Initial and final conditions.*

Thus, dynamical programming consist in defining a sequence of decisions $\mathcal{D} = \{D_1 \dots D_N\}$, triggering allowed transitions of our system from an initial state 0 to a final state N, while minimizing the overall induced cost:

$$\min_{\mathcal{D}} \sum_{i=1}^{N} Cost\ i$$

This problems can be formulated recursively, indeed a simple but outstanding property. No solution would be reachable without recursion. Let we discover the stratagem.

2.2 A first exercise

We begin with a toy example, a vehicle moving on earth's surface at the constant speed of 1 *kn* (one knot), required to travel from a Departure to the Arrival Point on the sphere. Of course, the expected solution is the arc of great circle linking Departure and Arrival. Let us, however, try to use DP to compute that route.

2.2.1 Decision network

For that purpose, one constructs a network of possible routes covering the area on which an optimal route is to be sought Figure 5. The position of the network is fixed by its Anchors D, respectively A. Both are drawn in green. Anchor D lies in Switzerland, Anchor A in New Zealand. The network is composed of N slices of M vertexes, with $\{N, M\} = \{11, 11\}$ in Figure 5. The slices, pictured as blue dots disposed orthogonally to the main direction, represent the decision steps presented in Figures 3 and 4. The network is the composition of routes linking vertices from the first slices to vertices belonging to the last slice. It is worth noting that each vertex of a slice is only connected to a subset of vertices of the next slice. In this sense, the network is not complete. This property, linked to a parameter called 'spread', is explained later. The departure vertex, corresponding to the starting location, has to be chosen in the initial slice, on the left. Correspondingly, the arrival vertex is chosen in the final slice, on the right. Drawn in purple, departure and arrival vertices need not to be located at the anchors.

2.2.2 Decision tree

Starting with a simple formulation that will be progressively refined, one defines: Δ_i, $i = 1 \dots N$ is the time required to travel from slice $i - 1$ to slice

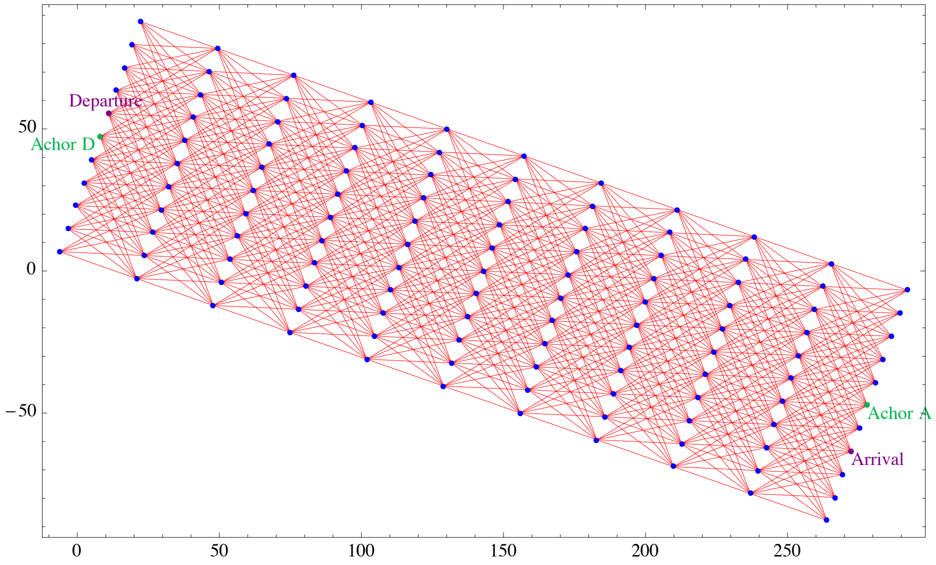

Figure 5: *Network onto which a route will be constructed. Abscissa: longitudes on the sphere, ordinate: latitudes. The sequencing is provided by the slicing of the array, from the northwest to the southeast.*

i. $i = 0$ corresponds to the initial slice, to which Anchor Departure belongs, and slice $i = N$, to which Arrival belongs.
D_i, $i = 0 \ldots N$ is the time at which the slice i is reached. $D_{i=0}$ is the departure time, $D_{i=N}$ is the arrival time.
The requirement for minimization of the induced cost is now a requirement on the time elapsed from Departure to Arrival. It is then given by:

$$D_N \; = \; \min_{i=1}^{N} \sum_{i=1}^{N} \Delta_i$$

Let us consider a intermediate step I with $1 \leq I \leq N$. It can be rewritten as:

$$\begin{aligned} D_I &= \min_{i=1}^{I} \sum_{i=1}^{I} \Delta_i \\ &= \min_{i=1}^{I} [\sum_{i=1}^{I-1} \Delta_i + \Delta_I] \\ &= \min_{I} [\min_{i=1}^{I-1} [\sum_{i=1}^{I-1} \Delta_i] + \Delta_I] \\ &= \min_{I} [\ D_{I-1} + \Delta_I \ ] \end{aligned}$$

Equating the first and the last expressions yields a recursive formulation of the problem. This derivation results from a profound optimization principle: a strategy is then optimal if all its sub-strategies are optimal. Let we now refine our formulation.

Deciding consists here in choosing for each vertex $\{i-1, k\}$ in the slice $I-1$ a next vertex $\{i, j\}$ belonging to the slide I towards which our vehicle should sail, or, equivalently, for each vertex $\{i, j\}$ in the slice I a previous vertex $\{i-1, k\}$ in the slice $I-1$ from which our vehicle should have sailed. Introducing the latter expression in the previous recursive expression for D_I, the problem is then formulated according to the *Belman-Ford equation* [5]: For each vertex $\{i, j\}$ belonging to the slice I, determine the vertex $\{i-1, k^*\}$ belonging to the slice $I-1$ satisfying:

$$D_{i,j} \ = \ \min_{k \, \in \, I-1} \ [\ D_{i-1,k} \ + \ \Delta_{i-1,k}^{i,j} \ ]$$

The initial condition is provided by $D_{I=0,k=departure} \ = \ departuretime$. The vertex $\{i-1, k^*\}$ represents the elementary decision taken in the slice I for the vertex $\{i, j\}$. D_I, whose name can be traded from "duration" to "decision", happens to be a vector of M elementary decisions, represented as

$$D_I \ = \ [\ \{i-1, k^*\} \ \rightarrow \ \{i, j\} \ ]$$

The elementary sailing duration $\Delta_{i-1,k}^{i,j}$ between the vertices $\{i-1, k\}$ and $\{i, j\}$ is given by the quotient of the orthodromic distance between their locations, $\{\theta_{i-1,k}, \phi_{i-1,k}\}$ and $\{\theta_{i,j}, \phi_{i,j}\}$, and the speed of the mobile. The

orthodromic distance, expressed in nautical miles[5] is provided by a function $\mathcal{O}_{\{\{\theta_{i-1,k}\phi_{i-1,k}\},\{\theta_{i,j},\phi_{i,j}\}\}}$ developed in Section 7.1 of the Appendix. In our simple case, where speed = 1 knot, $\Delta^{i,j}_{i-1,k} = \mathcal{O}_{\{\{\theta_{i-1,k}\phi_{i-1,k}\},\{\theta_{i,j},\phi_{i,j}\}\}}$. As speed remains constant, routes of shortest time are shortest routes as well. Of course, in real computations involving a sailing boat submitted to changing winds, the actual boat speed will be instrumental in the decision process.
A foible is to be circumvented. For a vertex located on one side of the array, our algorithm could choose, as it is defined, a next vertex located far away on the other side of the array, thus provoking a wild crossing through the array, indeed an unfortunate maneuver. A control mechanism has to be introduced. It consists in defining for each vertex $\{i,j\}$ in the slice I a subset of vertices $\{i-1,k^{\#}\}$ in the slice $I-1$ that could be addressed from the vertex $\{i,j\}$. Simply controlled by a integer called *spread*, already evoked, it defines allowed transitions of our system by restricting the decision process as follow:

$$D_I \; = \; [\; \{i-1,k^{\#}\} \; \rightarrow \; \{i,j\}; \; |k^{\#}-j| \leq \textit{spread}\;]$$

Additional constraints will be introduced later in order to cope with the shape of a lake.

2.3 Action

Following Figure 6 exhibits DP in action. The algorithm starts from the initial departure vertex and proceeds forwards in the network in constructing an arborescence, or a tree: each vertex has a unique parent, any vertex can have multiple offspring. By that way, an unique route links each vertex backwards to the root, to the departure vertex. All routes are optimal in the sense that they minimize the duration of the journey - in our case the distance - from the root to any vertex in the network. And only one route links arrival and departure vertices. This is the optimal solution of our problem.

Few remarks are in order.

1. On the final slice, the arrival and anchor vertices, although closely located, are reached by very different routes, both starting at the departure

[5] Refer to Section 7.1.2. for an explanation of the units

vertex. Different tactics may yield similar outcomes.

2. A few words about algorithmic complexity: would we try to evaluate all possible routes allowed in the decision network, Figure 5, we would face an unfathomable problem. The number of options would be of the magnitude order of $(SM)^N$ with spread $S = 7$, M, the width of a slice $= 11$ and the number of slices $N = 10$. In this setting, the number of options is $(7 \cdot 11)^{10}$. In real applications presented later, one has $S \approx 7$, M, ≈ 20 and the number of slices $N \approx 40$. A contraption evaluating a billion of options per second would require a computing time much longer than the age of our universe (10^{17} seconds). Fortunately, the recursive formulation allows the complexity of our problem to collapse from exponential to polynomial. The complexity is of the order of $S \cdot M \cdot N$ and a desktop computer achieves a computation within minutes.

3. DP generates a kind of herd dynamics: options are sought ahead in the solution space, then, the winning route, reaching the arrival vertex, defines the optimal solution. This analogy with herds is used to establish isochrones (loci of equal progression) in the final diagrams.

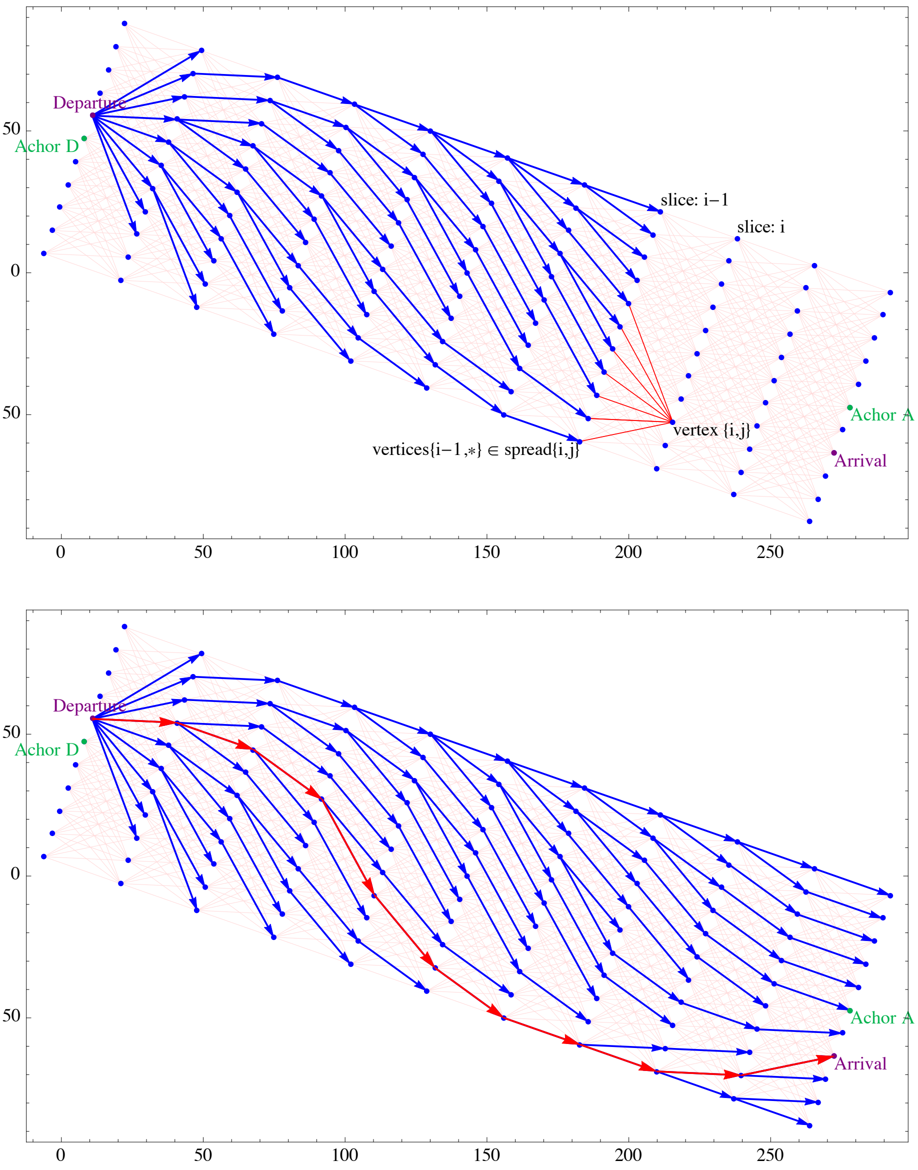

Figure 6: *Upper panel: construction of a tree (an arborescence) in the decision network. Lower panel: tree and the optimal route linking Departure to Arrival. Abscissa: longitudes on the sphere, ordinate: latitudes.*

3 Wind forecast

3.1 The COSMO-2 forecasting model

The wind forecast is delivered by the numerical weather forecasting model COSMO-2 in operation at MeteoSwiss [1]. The "Consortium for Small Scale Modeling", COSMO, is a consortium of eight European Weather Services[6] aimed at developing and operating numerical forecasting models [2].
COSMO-2 covers the alpine area on a rectangular domain with a diagonal reaching from Montpellier in France to Brno in the Czech Republic. Operated at a grid spacing of 2.2 km, COSMO-2 is non hydrostatic, equipped with comprehensive physics explicitly tuned to cope with alpine topography. It is run at CSCS, the Swiss Center for Scientific Computing, on two Cray XE6 machines [3]. COSMO-2 is nested and inherits initial and boundary conditions from a broader COSMO-7 model, itself nested in the global forecasting model IFS from ECMWF [4]. Last, but not the least, a nudging system provides COSMO-2 with real-time surface and remote observations collected on the forecasting domain during the assimilation cycles.

Put in a nutshell, COSMO-2 is a non-linear dynamical system of 78 million variables, the dimension of its phase space. It is integrated at CSCS 8 times a day and thus delivers every three hours 0-32-hours forecasts for almost all possible meteorological parameters in the domain. Relevant data, tailored to customers' needs, is then extracted with a software called Field-Extra.

3.2 Wind field over the lake

For us the wind field at 10 meters height over terrain or water on an area covering the basin of the lake under consideration is relevant. Figure 8 shows such fields as green arrows, for a +12 hours forecast, upper panel, and +21 hours forecast, lower panel. One notices that the wind field provided by COSMO-2 does not follow the latitude-longitude frame, it is tilted. The "north pole" of the model is shifted in order to allow an optimal implementation of numerical algorithms in the alpine area. Blue arrows are wind vectors interpolated from the model frame to the vertexes of the decision network. This network is presented in Figure 7.

[6] At time of writing

3.2.1 Cropping algorithm

In Figure 7 below, the decision network appears almost exactly tailored to the shape of the lake. This is realized in two steps. Firstly, after having fixed the departure and arrival anchors, one generates a rectangular network similar to Figure 6, covering an area slightly broader than that of the lake. A theorem by Camille Jordan inspires the second step. It claims that *any closed line in the plane divides it in two distinct regions.* Of course, for us, the shore of the lake defines the curve, provided as a closed polygon. The method, derived from Jordan's proof, consists in sending from each vertex of the network half straight lines in any direction and in counting the number of intersections of a half line with the polygon. Is this number odd, then the vertex lies within the polygon, on the lake. Is the number even, then the vertex is located outside[7] [5]. This operation is repeated for each vertex and delivers the sub-network illustrated in Figure 7. Vertices lying on the lake are referred as 'elected vertices' in the sequel.

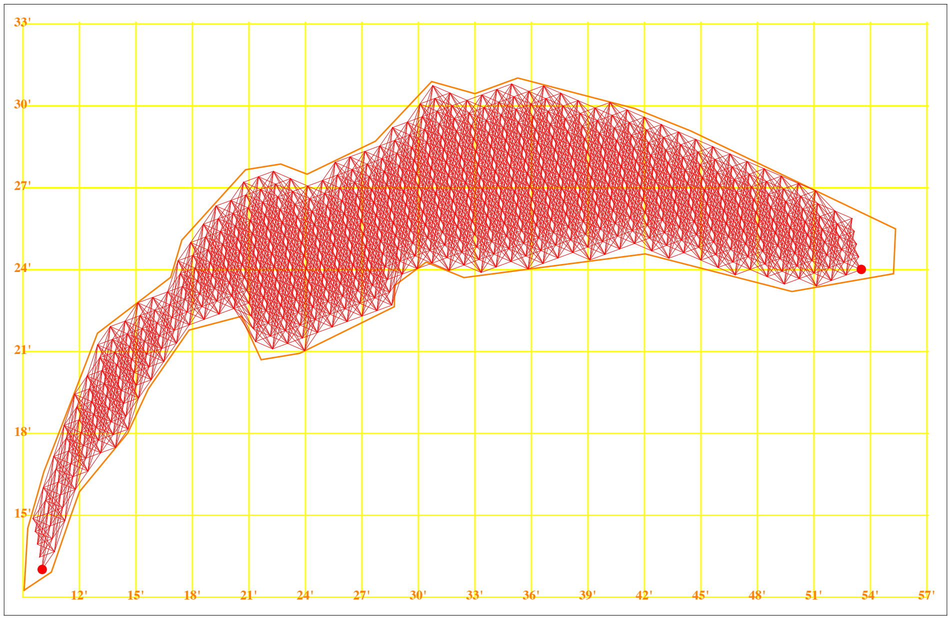

Figure 7: *Network of elected vertices.*

[7] Tangency points of a half line with the shore of are disregarded while counting.

The frame of the network has to be carefully set. Coarse course changes of the sailing boat might be forced if the ratio of the distances between slices in the network and lateral neighborhood vertices is low. In this respect, due of its arched shape and numerous coves, Lake Léman represents a challenge to the designer. In the setting shown in Figure 7, the lake is accurately covered by the network, at the price of having rather coarse courses changes, mostly in the middle of the lake.

3.2.2 Interpolation algorithm

Interpolation of the model wind field onto the set of elected vertices is realized with a 'pondered closest neighbourhood' algorithm. Making use of the orthodromic function, one determines for each elected vertex v of the network its four closest neighbours $m_1 \dots m_4$ among the grid points of the model. One collects the four distances $d_1 \dots d_4$ and their sum $D = \sum_{i=1}^{4} d_i$. Then, the interpolated wind at elected vertex v is given by:

$$\begin{bmatrix} U \\ V \end{bmatrix}_v = \frac{1}{3D} \sum_{i=1}^{4} \sum_{j \neq i}^{4} d_j \begin{bmatrix} U \\ V \end{bmatrix}_{m_i}$$

U and V are the zonal, respectively the meridional components[8] of the wind field at vertex v, respectively at grid point m_i.
This operation is repeated for each temporal forecast of the model, from 0 to +32 hours, and produces a kind of 'wind silo' that will later be tapped by the DP algorithm.

3.2.3 Stochastic versus deterministic dynamical programming

The intelligence of our system is mostly encapsulated in the forecasting model, COSMO-2. This intelligence reaches far beyond strict meteorology and encompasses a large fraction of the environmental processes influencing wind fields on the lake, including the orography, the vegetation types in the surrounding landscape, the radiative processes, to cite few of them. Indeed COSMO-2 simulates the interactions between those elements on the whole forecasting area, that is incomparably broader than the basin of the lake. The forecast of the wind field on the lake, a tiny subset of the material data by the model, is then provided as a local set of deterministic information.

[8] U, the zonal component, flows along a latitude circle. V, the meridional component, flows along a meridian circle.

Compared to the huge simulation process, the decision scheme based on dynamical programming happens to be rather trivial.

A different approach takes into account observations made on the lake and consists in inferring statistical properties of the wind field, as provided through this measurement process. The local climatology of the lake is then established. In such a setting, the decision scheme manipulates stochastic data and is much more refined than our simple dynamical system: a Stochastic Dynamical Programming scheme has to be implemented. A transfer of "intelligence" occurs from the meteorological model to the decision scheme. This approach was chosen by the Alinghi team for the America Cup. Prof. Dalang of the ETH Lausanne and his team developed such stochastic dynamical systems, that are mentioned in reference [11].

Considered from the point of view of game theory, the first approach is akin to a game with complete information, provided by the weather forecasting model. In the second approach, the information, being provided on a probabilistic basis, is only partially available to the decision scheme.

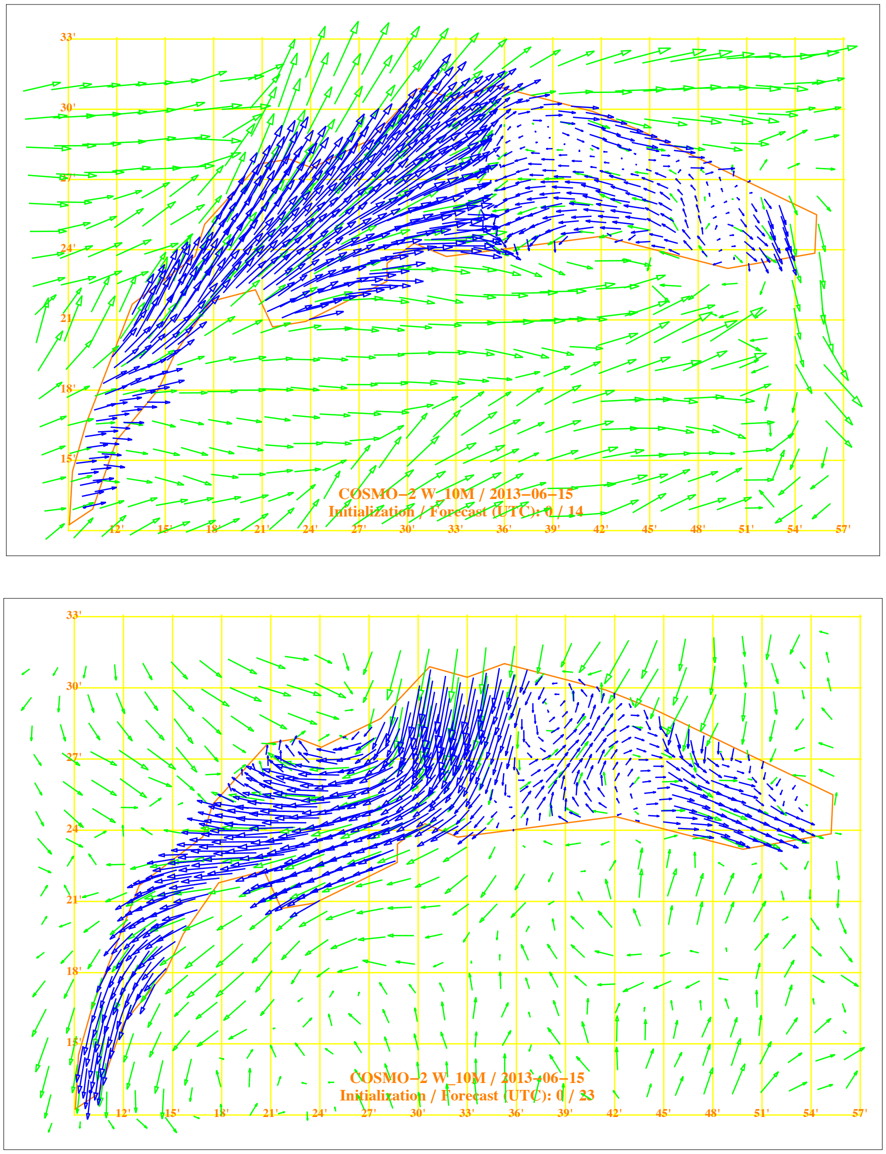

Figure 8: *Wind Fields delivered by COSMO-2 model, green vectors, and their interpolation on the decision network, blue vectors. Upper panel: +14 hours forecast, lower panel +23 hours forecast.*

4 The sailing boat and its polar diagram

The polar diagram provides the best performance that can be expected from a sailing boat, for each incidence angle and speed of the wind falling on its track. Provided on following Figure 9, it is parametrized in incidence angles, from 0 to 180 degrees, and sailing speed of the boat, given as the radius on the diagram. Sailing speed is vertically labelled in knots on the left side of the diagram. Represented here for starboard, the diagram is symmetric and equally valid for port.
The array of curves on the polar diagram is labelled in wind speed between 1 and 25 knots (kn). The polar radius of a wind curve of speed $|\vec{W}|$ at incidence angle α is the corresponding sailing speed of the boat at this wind speed and incidence angle. Thus, the boat speed is expressed as a function $\mathcal{Y}_{(\alpha,|\vec{W}|)}$, (with $\mathcal{Y}$ for 'yacht'), as provided on the diagram.

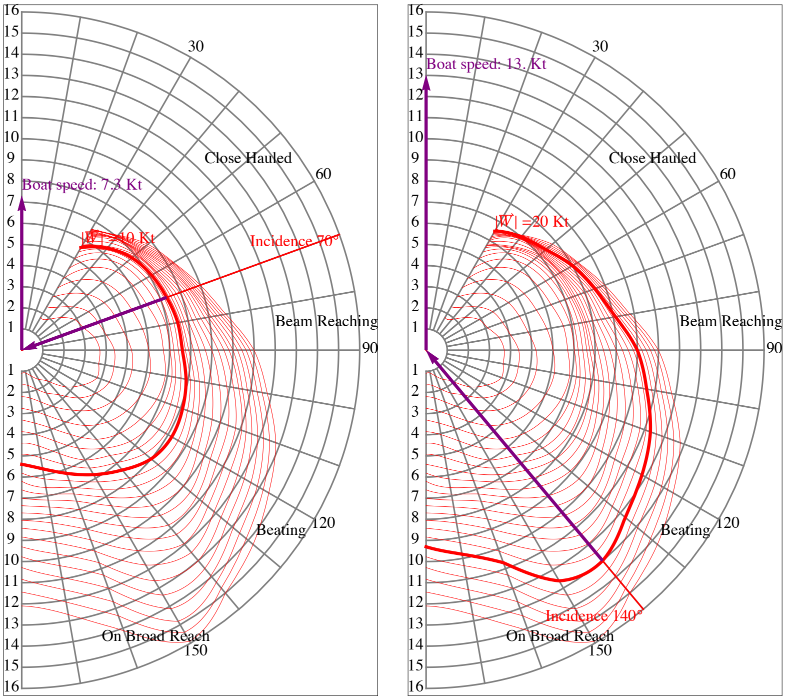

Figure 9: *Polar Diagram: the boat is sailing towards the top of the diagram. Left panel: beam reaching, wind speed 10 kn, boat speed 7.3 kn. Right panel: on broad reach, wind speed 20 kn, boat speed 13 kn.*

Sailors' language is picturesque. A boat is *close hauled* if the wind is falling on its track with an angle between 30^o and 60^o. Sailing under cross wind, almost perpendicular to the track, is called *beam reaching.* The boat is *beating* when the incidence angle is about 120^o, it is *on broad reach* when the wind hits it laterally astern, with an incidence angle about 150^o. Further, it is *on run*, in back wind. These sailing setups are sketched in Figure 9. Examples: left panel: our boat is sailing under 10 knots of wind falling on its track with an angle of 70^o: it is beam reaching at 7 knots. Right panel: sailing on broad reach - 140^o - under 20 knots wind delivers a boat speed of 13 knots.
Only the real wind, related to the geographical coordinate system, is considered as input in the polar diagram. The apparent wind results in the vector composition of the real wind and the speed of the sailing boat. Mostly effective when reaching under light airs, it is taken into account in the diagram itself. In our example, beam reaching under 2 knot real wind delivers a boat spread of 2.7 knots: thanks to the apparent wind, the boat speed is higher than the wind speed.
The polar diagram is usually delivered by the naval architect who designed the boat. Practical measurements on waters, yielding insights in the boat's behavior and its performance, may improve the representativeness of a polar diagram.
The polar diagram is provided under the assumption that the sailing boat is optimally trimmed, a responsibility taken over by the crew.

4.1 Yacht function

Taking into account the setting of the polar diagram and making use of the navigation instruments developed in section 3, the incidence angle $\alpha_{\{\vec{W},\vec{T}\}}$ of a boat sailing on track $\vec{T}$ under wind conditions $\vec{W}$ is given by:

$$\alpha_{\{\vec{W},\vec{T}\}} \;=\; \pi - \arccos[\;\frac{\vec{W}\;\cdot\;\vec{T}}{|\vec{W}|\;|\vec{T}|}\;]$$

Finally, the wind speed $|\vec{W}| = \sqrt{U^2+V^2}$ (with zonal and meridional components U and V) and the incidence angle $\alpha_{\{\vec{W},\vec{T}\}}$ are introduced in an interpolation scheme that delivers the corresponding boat speed expressed in knots as:

$$\mathcal{Y}_{(\alpha_{\{\vec{W},\vec{T}\}},|\vec{W}|)}$$

Few precautions are in order: for numerical reasons, the boat speed should never equal zero. Therefore, it is set to a low value, typically 0.1 knot, when calm reigns, or if an attempt is made to compute a speed headwind. If the wind speed is higher than the maximal wind speed considered in the diagram, then the calculation is done with that maximal wind speed, for the corresponding incidence angle. Finally, no penalty is ascribed to a tack, an improvement that would enhance the reliability of the simulation.

5 Navigation

5.1 Spherical coordinates

In this project, all navigation issues are solved on a Earth's globe. The Earth is represented as the S^2 unit sphere immersed in R^3. Locations are given on earth with longitude θ and latitude ϕ, where $\theta \in [0, 2\pi)$ spans longitudes eastwardly and $\phi \in (-\frac{\pi}{2}, \frac{\pi}{2})$ spans latitudes from south to north poles. The immersion obeys the standard application:

$$
\begin{array}{rcl}
[0,2\pi) \times (-\dfrac{\pi}{2}, \dfrac{\pi}{2}) & \longrightarrow & S^2 \subset \mathrm{R}^3 \\
\{\theta, \phi\} & \longrightarrow & \vec{X}_{\{\theta,\phi\}} = \begin{bmatrix} cos(\theta)\ cos(\phi) \\ sin(\theta)\ cos(\phi) \\ sin(\phi) \end{bmatrix}
\end{array}
$$

A location on the sphere, $\{\theta_{i,j}, \phi_{i,j}\}$, is associated to each vertex $\{j, j\}$ belonging to the decision network. Each vertex is then projected into R^3 through the application $\vec{X}_{\{\theta_{i,j},\phi_{i,j}\}}$. Accordingly, the navigation problem is entirely treated in 3 dimensional space. This procedure provides us with elegant solutions, even on a lake, and Swiss coordinate and metric systems are disregarded. Why?

5.2 Why nautical miles?

A Nautical Mile its the distance separating two locations on earth with longitude θ and latitude ϕ, $\{\theta_1, \phi_1\}$ and $\{\theta_2, \phi_2\}$, making with the centre $\vec{X}_C$ of earth an angle $\angle(\vec{X}_{\{\theta_1,\phi_1\}}, \vec{X}_C, \vec{X}_{\{\theta_2,\phi_2\}})$ of one minute of degree.
Navigation is performed without any reference to the actual radius of Earth, or of any planet on which a regatta would be organized. Furthermore, evaluating speed in knots - nautical miles per hour - makes us only dependent on the angular rotation of the planet, one rotation of 2π in 24 hours. These properties justify the way navigation systems are conceived: they are suited

to any planet or rotating spherical body. I almost forgot to tell you: on Earth, 1 Nautical Mile = 1.852 m, 1 knot = 0.514 m/s.

5.3 Navigation Toolbox

The tangent plane to the sphere at a location $\{\theta, \phi\}$, noted $\mathcal{T}$, is given a 2-dimensional vector space structure with basis $\{\vec{e}_\theta, \vec{e}_\phi\}$, where both basis vectors are tangent to the sphere at location $\{\theta, \phi\}$ and tangent to the coordinate lines passing by θ and ϕ. Pictured in Figure 10, they are given by:

$$\begin{bmatrix} \vec{e}_\theta \\ \vec{e}_\phi \end{bmatrix}_{\{\theta,\phi\}} = \begin{bmatrix} \frac{1}{cos(\phi)}\ \partial_\theta \vec{X}_{\{\theta,\phi\}} \\ \\ \partial_\phi \vec{X}_{\{\theta,\phi\}} \end{bmatrix}$$

The division by $cos(\phi)$ in the first term ensures that the frame be orthonormal: $|\vec{e}_\theta| = |\vec{e}_\phi| = 1$ with $\vec{e}_\theta \perp \vec{e}_\phi$. The horizontal wind field delivered by the numerical forecasting model, described in chapter 3, is easily expressed in this frame: $\vec{W}_{\{\theta,\phi,t\}} = U_{(t)}\ \vec{e}_\theta + V_{(t)}\ \vec{e}_\phi$, where $U_{(t)}$ is the zonal component of the wind at location $\{\theta, \phi\}$ and $V_{(t)}$ its meridional component, both at time t. Time dependency will represent a next challenge! The zonal component of the wind is directed along a latitude circle, its meridional component along a meridian circle.

We calculate now the orthodromic distance on earth between two locations $\{\theta_1, \phi_1\}$ and $\{\theta_2, \phi_2\}$. Together with the centre $\vec{X}_C$ of earth, their location vectors $\vec{X}_{\{\theta_1,\phi_1\}}$ and $\vec{X}_{\{\theta_2,\phi_2\}}$ define a plane Π whose intersection with the sphere is the orthodromy, or great circle, Figure 10. The orthodromic distance is simply the angle $\angle(\vec{X}_{\{\theta_1,\phi_1\}},\ \vec{X}_C,\ \vec{X}_{\{\theta_2,\phi_2\}})$ between the two vectors, expressed in nautical miles. It is given by the ArcCosinus of their scalar product[9]:

$$\mathcal{O}_{\{\{\theta_1\phi_1\},\{\theta_2,\phi_2\}\}} = \arccos[\vec{X}_{\{\theta_1,\phi_1\}} \cdot \vec{X}_{\{\theta_2,\phi_2\}}]$$

The arc of great circle linking the two locations is parametrized according to the curvilinear abscissa $\tau \in [0, \mathcal{O}_{\{\{\theta_1\phi_1\},\{\theta_2,\phi_2\}\}}]$ and its locus is provided by[10]:

$$cos(\tau)\ \vec{X}_{\{\theta_1,\phi_1\}} + sin(\tau)\ \vec{T}_{\{\theta_1,\phi_1\}}$$

$\vec{T}_{\{\theta_1,\phi_1\}}$ is the vector tangent to the arc of grand circle at location $\{\theta_1, \phi_1\}$, indeed the track of our sailing boat. It is determined in two steps. Firstly, one computes the vector $\vec{N}$ normal to the plane Π spanned by $\vec{X}_{\{\theta_1,\phi_1\}}$ and

[9] According to section 6.1.1, $|\vec{X}_{\{\theta_1,\phi_1\}}| = |\vec{X}_{\{\theta_2,\phi_2\}}| = 1$.

[10] The great circle is generated when $\tau \in [0, 2\pi]$.

$\vec{X}_{\{\theta_2,\phi_2\}}$: $\vec{N} = \vec{X}_{\{\theta_1,\phi_1\}} \times \vec{X}_{\{\theta_2,\phi_2\}}$. Then, a further cross product delivers the track vector itself:

$$\vec{T}_{\{\theta_1,\phi_1\}} = \vec{N} \times \vec{X}_{\{\theta_1,\phi_1\}}$$

Of course, the same result holds at location $\{\theta_2, \phi_2\}$ with $\vec{T}_{\{\theta_2,\phi_2\}} = \vec{N} \times \vec{X}_{\{\theta_2,\phi_2\}}$. Thanks to the geometrical construction, both the track $\vec{T}_{\{\theta,\phi\}}$ and the wind vector $\vec{W}_{\{\theta,\phi,\,t\}}$ belong to the plane $\mathcal{T}$ tangent to the sphere at a location $\{\theta, \phi\}$ with basis $\{\vec{e}_\theta, \vec{e}_\phi\}$. Thus, a simple implementation consists in projecting the track vector onto the $\{\vec{e}_\theta, \vec{e}_\phi\}$ frame, and in making direct use the U and V components of the wind vector. Accordingly, the actual bearing of the sailing boat, expressed as an angle from north to its track, is of no use in this computation.

All these elements are implemented in the sailing program. Figure 10 provides an overview.

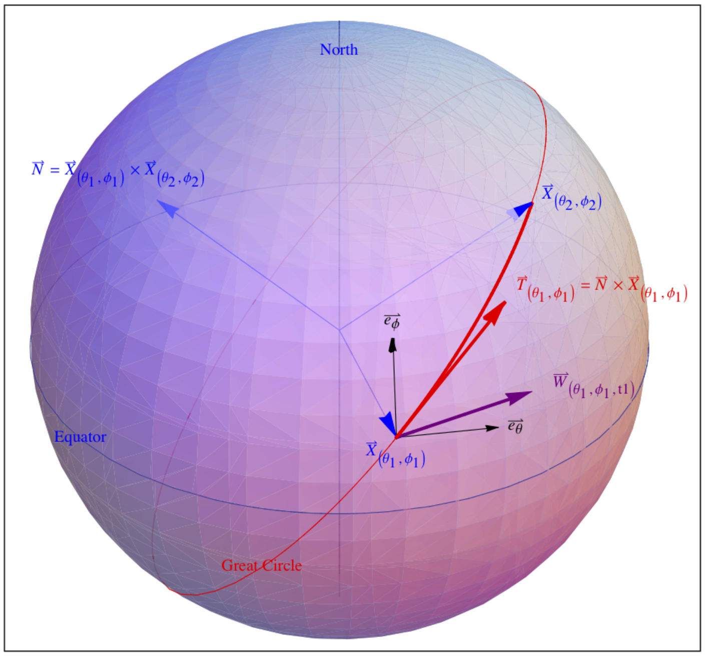
North
$\vec{N} = \vec{X}_{(\theta_1,\phi_1)} \times \vec{X}_{(\theta_2,\phi_2)}$
$\vec{X}_{(\theta_2,\phi_2)}$
$\vec{T}_{(\theta_1,\phi_1)} = \vec{N} \times \vec{X}_{(\theta_1,\phi_1)}$
$\vec{e_\phi}$
$\vec{W}_{(\theta_1,\phi_1,t1)}$
$\vec{e_\theta}$
Equator
$\vec{X}_{(\theta_1,\phi_1)}$
Great Circle

Figure 10: *Spherical navigation. Bold red curve: orthodromy between two locations* $\vec{X}_{\{\theta_1,\phi_1\}}$ *and* $\vec{X}_{\{\theta_2,\phi_2\}}$. *Track at* $\vec{X}_{\{\theta_1,\phi_1\}}$: $\vec{T}_{\{\theta_1,\phi_1\}}$. *Wind vector:* $\vec{W}_{\{\theta_1,\phi_1,t1\}}$. *Accordingly, the boat is on broad reach, port tack on a north easterly route.*

6 Time elapsed on an arc

Let we now estimate the sailing time on the arc $\vec{T}$ linking a vertex $\{i-1,k\}$ to a vertex $\{i,j\}$, see Figure 11.
Simplifying the typography, we call the former vertex 1, the later 2, located at $\{\theta_1,\phi_1\}$ and $\{\theta_2,\phi_2\}$. Their locations being close on a lake, their corresponding tracks and can be identified: $\vec{T} = \vec{T}_{\{\theta_1,\phi_1\}} \approx \vec{T}_{\{\theta_2,\phi_2\}}$. The distance between them is easily known: $\Lambda = \mathcal{O}_{\{\{\theta_1\phi_1\},\{\theta_2,\phi_2\}\}}$. Let be t_1, respectively t_2, the time at which the boat crosses vertices 1 and 2. The wind blowing at time $t = t_1$ at vertex 1 is denoted $\vec{W}_{(t_1)}$, at time $t = t_2$ at vertex 2, $\vec{W}_{(t_2)}$. The former wind is known, the latter not.

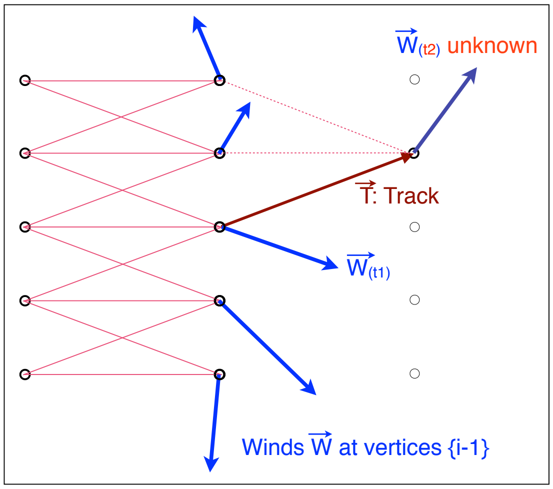

Figure 11: *Sketch of the conventions established so far.*

The speed is provided by $\mathcal{Y}_{(\alpha_{\{\vec{W},\vec{T}\}},|\vec{W}|)}$ from the polar diagram. Let us now remember the expression for Dynamical Programming:

$$D_{i,j} \;=\; \min_{k\,\in\,I-1} \;[\; D_{i-1,k} \;+\; \Delta^{i,j}_{i-1,k} \;]$$

The duration of the journey from $\{i-i,k\}$ to $\{i,j\}$, or equivalently from $\{\theta_1,\phi_1\}$ to $\{\theta_2,\phi_2\}$, is given by $\Delta^{i,j}_{i-1,k}$ or, in a simpler notation, Δ^2_1. DP provides us with $t_1 = D_{i-1,k}$. On the contrary, t_2, the arrival time at vertex 2, is unknown. Its value should be $t_2 = t_1 + \Delta^2_1$, where Δ^2_1 is our unknown. Δ^2_1 is expressed as the quotient of the length of the arc by the average speed of the yacht along it, provided by the function $\mathcal{Y}_{(\alpha_{\{\vec{W}_{(t)},\vec{T}\}},|\vec{W}_{(t)}|)}$ (section 4):

$$\Delta^{i,j}_{i-1,k} = \Delta^2_1 = \frac{2\,\Lambda}{\mathcal{Y}_{(\alpha_{\{\vec{W}_{(t_1)},\vec{T}\}},|\vec{W}_{(t_1)}|)} + \mathcal{Y}_{(\alpha_{\{\vec{W}_{(t_1+\Delta^2_1)},\vec{T}\}},|\vec{W}_{(t_1+\Delta^2_1)}|)}}$$

And now the shock: we are confronted with a circularity. The previous expression can not be solved in a straightforward manner, as Δ^2_1 is expressed in terms of itself, a feat summarized as $\Delta^2_1 = \mathcal{F}(\Delta^2_1)$. Who throws us a lifeline?

6.1 Fixed point and relaxation

Liutzen Brower developed by the beginning of the last century the 'fixed point theory', that deals with such questions. The theory has blossomed since then: fixed point theorems have found applications in pure mathematics, physics, engineering, economics, finance, game theory, to name few.

A simple formulation of a fixed point theorem, relevant for our application, is: *In the plane, every continuous application $f : D \rightarrow D$ from a convex domain D to itself leaves at least one point $x^* \in D$ fixed, such that* $x^* = f_{(x^*)}$. Formally, *given a metric $\mu : D \times D \rightarrow \mathrm{R}^+$, the contraction condition $\mu(f_{(x)}, f_{(x')}) < \tau\ \mu(x, x')$ stipulates that, for $\tau \in \mathrm{R}^+$, $\tau < 1 \Leftrightarrow \exists\ x^* \in D$ such that $x^* = f_{(x^*)}$.*

For us, the convex domain is simply the closed segment $[0, T_{max}]$ on the real line R, where T_{max} in the maximum forecasting range of the forecasting model.

Fixed point theorems are existence theorems. They tell us under which conditions solutions do exist, but usually do not provide us with methods to compute them. Two techniques are presented below to solve the issue. Both are based on relaxation.

6.2 First relaxation method

The first technique, quite trivial, makes use of the contraction condition evoked in the theorem. Firstly, on rewrites $\Delta^2_1 = \Delta_{(t_1,t_2)}$ where t_1 is known and $t_2 = t_1 + \Delta_{(t_1,t_2)}$ has to be estimated. Then one iterates on an index i: $(t_2)_{i=0} = t_1 + \Delta_{(t_1,t_1)} \ldots (t_2)_i = t_1 + \Delta_{(t_1,(t_2)_{i-1})}$. By the initial step $i = 0$, one assumes that the wind at vertex 2 does not change from its value at time t_1 during the journey from vertex 1 to vertex 2. By step $i = 1$ one extracts form the wind silo (chapter 3 and Figure 12) the wind at vertex 2 at time $(t_2)_1$ which gives us a better estimation of the duration of the journey between both vertices, and so forth. One gets, at recursivity depth four:

$$t_2 = t_1 + \Delta_{(t_1, t_1 + \Delta_{(t_1, t_1 + \Delta_{(t_1, t_1 + \Delta_{(t_1, t_1)})})})}$$

Time t_2 does no longer appear in right hand side of the expression. This formulation has been implemented at recursivity depth four in past versions of the sailing software.

6.3 Second relaxation method (newtonian descent)

Smarter, this method takes advantage of the fact that a point x^*, fixed for the expression $x = f_{(x)}$, is a zero of the new expression $\psi_{(x)} = x - f_{(x)}$. The zeroes of $\psi_{(x)}$ are classically reached with the help of a newtonian descent algorithm of the form $x_{i+1} = x_i - \frac{\psi_{(x_i)}}{\psi'_{(x_i)}}$.

In our case, $f_{(x)}$ corresponds to $t_1 + \Delta_{(t_1, t_2)}$ and $\psi_{(x)}$ to $\Psi_{(t_2)} = t_2 - (t_1 + \Delta_{(t_1, t_2)})$, so that the descent algorithm reads:

$$(t_2)_{i+1} = (t_2)_i - \frac{(t_2)_i - (t_1 + \Delta_{(t_1, (t_2)_i)})}{1 - \Delta'_{(t_1, (t_2)_i)}}$$

The derivative in the denominator is estimated as:

$$\Delta'_{(t_1, (t_2)_i)} \approx \frac{1}{2h} (\Delta_{(t_1, (t_2)_i + h)} - \Delta_{(t_1, (t_2)_i - h)})$$

with h arbitrarily set at one tenth of an hour. The initial condition is provided by $(t_2)_{i=1} = t_1 + \Delta_{(t_1, t_1)}$. Experiences indicate a satisfactory convergence within five steps beyond the initial condition. This algorithm is implemented in the latest version of the sailing software.

6.4 Control of the convergence

Either the first or the second method is called during the construction of the decision tree. It is evaluated on all the arcs linking elected vertices in the network. The contraction condition $\tau < 1$ being not explicitly verified, an exclusion clause is implemented in both methods in order to catch cases where $(t_2)_i$ leaps outside the $[0, T_{max}]$ domain. Tests demonstrate a slow convergence for $\lim_{i \to \infty} (t_2)_i = {t_2}^*$ with method 1, a much faster convergence with method 2.

7 Implementation

The abstraction named 'System' in Chapter 2 is now palpable. All elements discussed so far - lake, forecasted wind field, sailing boat, navigation, network, elected vertices, decision tree, optimal route, isochrones - are integrated in a coherent whole, in a System. The construction of its parts, the sub-systems and their mutual interactions, is accomplished. It is presented as a block diagram in following Figure 12.

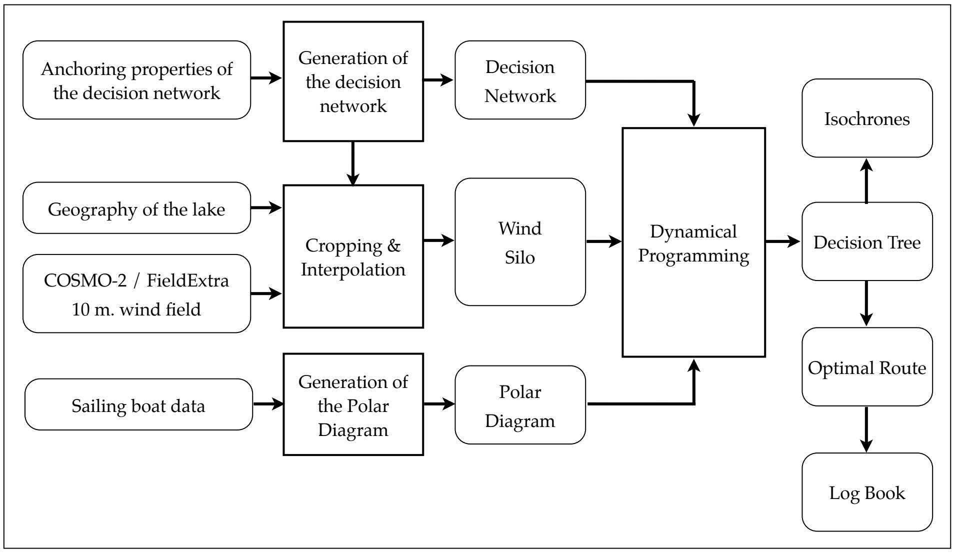

Figure 12: *Flow Diagram of the routing system.*

8 Bol d'Or 2013

Let us now contemplate Figure 13. In the upper panel, the tree gradually invades the network from Geneva eastwards towards le Bouveret and seeks any possible path. It forms the set of shortest routes linking each vertex back to the departure vertex, located at the root. The interaction of the sailing boat with the wind field, as it evolves in place and time during the journey, is taken into account in each local decision. These multiple local decisions shape the tree. Eventually, the unique path linking arrival back to departure is the optimal route sought for our yacht, indeed the solution of our problem. The isochrones are depicted in the middle and lower panels, formed of sets of red dots disposed almost perpendicularly to the main stream of the graph. Labelled in hours, they exhibit the locus that a swarm on sailing boats, all identical to our yacht (clones of it), would have reached at a given time within the tree. They allow the viewer of this diagram to grasp at a global level the tactical decision taken by DP in a regatta, in accordance with evolving weather patterns. The decision tree and the optimal route are specific to the sailing boat under consideration, as characterized by its polar diagram. Both tree and route would be different for a boat sporting another diagram.

The lower panel differs from the upper one only through the addition of a wind field. A subtlety, however, lurks. Each wind vector depicts the wind at an elected vertex at the time at which the algorithm would let a sailing boat crossing this vertex. Thus, the actual wind field is not pictured here. What is shown is a virtual, temporally evolving wind field, as it unfurls during the regatta. As for the isochrones, this dynamical representation enables a better understanding of the tactical decision taken at global level by DP. Figure 14 describes the way home, from the Bouveret back to Geneva.

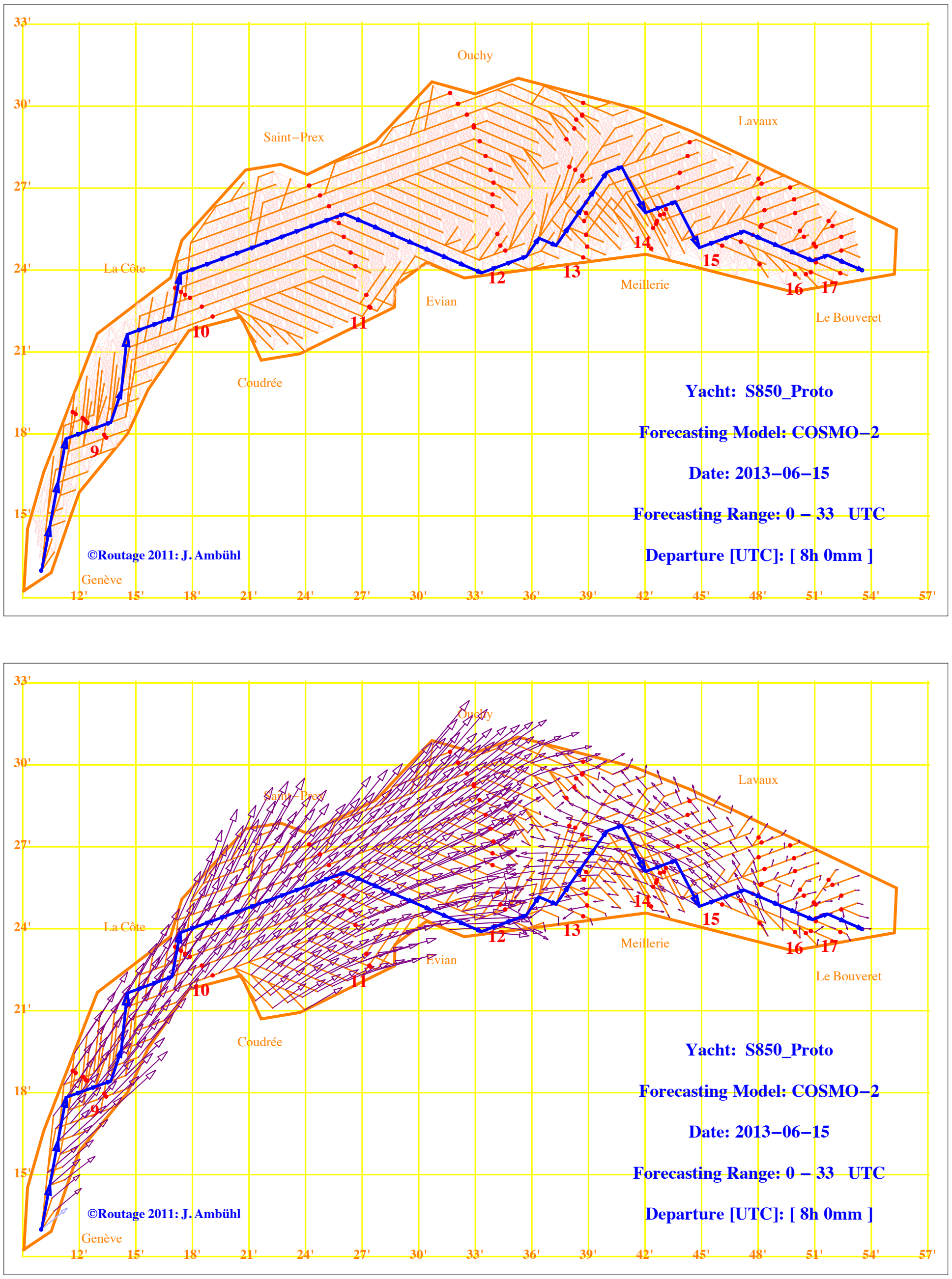

Figure 13: *Upper panel: decision tree, optimal route and isochrones. Lower panel: decision tree, optimal route, isochrones and wind field.*

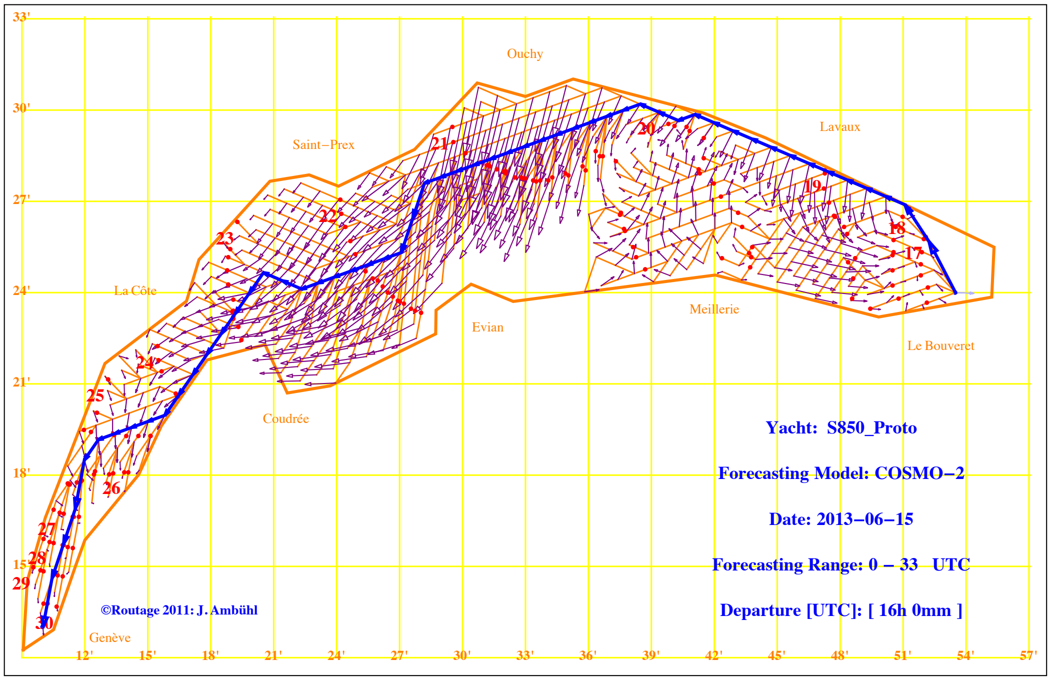

Figure 14: *Sailing back home. Departure from the Bouveret at 17.50 am local time. Arrival in Geneva on Sunday, at 4 pm local time.*

8.1 Discussion

Both routes, firstly from Geneva to the Bouveret, and then back home, have been computed using the 00 UTC forecast of COSMO-2 delivered at about 4 pm local time on saturday 15 june 2013.
Although looking like the description of a past event, these maps are forecasts, indeed option proposals for skippers. Let us now compare reality and forecast.

8.1.1 Standpoint of the machine

The machine provides its standpoint graphically. On Figure 13, the algorithm produces back wind tacks in the morning in order to keep the yacht in broad reach (it is more efficient in broad reach than on the run, Figure 9). Later, in the middle of the lake, it anticipates the weak easterly regime on the high lake (eastern part of the Léman), sends the boat to the French coast and allows it to cope optimally with the incoming easterly wind.
On the way home, from the Bouveret back to Geneva, Figure 14, the algorithm disregards in the evening the area located in front of Evian (an effect of the control of the convergence), the area located in front of Lavaux being much more favorable. This is well described by the distorted shape of the 20 UTC isochrone. Later, the 21 UTC isochrone is split in two parts with few dots left back near the french coast and a larger flock running in broad reach, starboard tack between Ouchy and St.Prex.

8.1.2 The skipper's viewpoint

The routing happened to be pertinent when forecasted wind fields were in accordance with reality. This occurred from the departure, at 10 am, up to 20 pm, 18 UTC. Later, divergences developed between COSMO-2 forecast and actual winds on the lake. Thunderstorms raised at dusk in the Rhone valley and triggered a pugnacious "Vaudaire" (easterly downhill wind) instead of the gentle easterly breeze "Vauderon" forecasted 20 hours earlier by the model. A second thunderstorm hoisted later "dans les Drances" in the vicinity of Evian, that had not been foreboded by the numerical forecast. Due to this convective activity in the first half of the night, forecasted and real wind fields did no longer match anymore[11]. This is only later in the night and at dawn that weak nighty winds were again correctly grasped by the model. We cut the arrival line after 23 hours 55 minutes, in 90th

[11] at that time, the weather forecast and the corresponding routing were 22 hours old.

position over 508 competing yachts. Summarizing, we have been positively surprised by the overall accuracy of the wind fields delivered by the model, and impressed by the pertinence of the options proposed by the algorithm[12]*. Philippe Jeanneret, 18. September 2013.*

9 Conclusion and outlook

Making use of 2 km wind forecasts provided by modern weather numerical models, the exercise was aimed at reconstructing for a lake the algorithms developed in the eighties for ocean routing.

Tremendous improvements in software techniques enabled us to enhance major parts of the algorithms implemented three decades ago. New communication systems (internet, tablets) allowed the direct transfer of numerical results gained on a maine frame, in our case CSCS, to the tablet of a sailor, this shortly before the start of a regatta.

Further improvements are announced in numerical forecasting technique. MeteoSwiss will run its COSMO model on a 1 km grid by 2015. At the same time, a new technique named ”Kalman Ensemble Assimilation” shall enhance the way the model grasps local weather events - fog patches, breezes, thunderstorms - in real time while building the initial condition of a model integration.

Regarding decision algorithms, the combination of stochastic programming and ensemble technique [10] is likely to open fascinating avenues. By that way, the natural unsteadiness of the wind field will be taken into account. However, opportunities will not only be offered to lake or ocean routing designers, but also to managers of complex, weather dependent systems. In this respect, intelligent grids fueled by renewable energies are on the verge of becoming an active field of applications for stochastic dynamical optimization.

Technology always fostered sailing. New designs for sails, hulls or keels, new materials, opened new challenges, enabled new endeavors. Were sailing authorities willing to allow ”life routing” during regattas, they would trigger a similar leap. A new era would open in sailing sports.

[12] author’s translation.

10 References

1. http://www.meteoschweiz.admin.ch/web/de/wetter/modelle.html

2. http://www.cosmo-model.org

3. http://www.cscs.ch

4. http://www.ecmwf.int

5. Eugen Lawler. 1976 Combinatorial Optimization. Networks and Matroids. (Holt, Rinehart, Winston, pp 65-77)

6. http://ig.halwe.dk/2008/10/implementing-fast-point-in-polygon.html

7. Ambühl J. Eckert P. 1986. A study of optimal routes for a sailing boat. Application to the case UBS-Switzerland during the 1985-86 Whitbread round the world race. (ECMWF Newsletter Nr 35, pp. 3-9)

8. Ambühl J. Eckert P. 1987. Calcul de la route optimale d'un voilier. Application à la course autour du monde 1985-1986 du voilier UBS-Switzerland. (Arbeitsbericht Nr. 141 der schweizerischen meteorologischen Anstalt)

9. Ambühl J. 1992. Des compétences EPFL naviguent avec Merit autour du monde. (CAST, Centre d'Appui Scientifique et Technologique, Rencontres EPFL-Economie. Notes de conférence)

10. Palmer T. N. and all. 2007. The Ensemble Prediction System - Recent and Ongoing Developments. (ECMWF Technical Memorandum no. 540)

11. Dumas F. 2010. Stochastic Optimization of Sailing Trajectories in an America's Cup Race. (Ph.D. thesis 4884. Ecole Polytechnique Fédérale de Lausanne, Switzerland)

12. Hablützel M. Yükseldi F. 2012. LakeRouting. Optimale Wegfindung anhand von Wettermodellen (Diplomarbeit ZHAW)

10.1 Author's address

Jacques Ambühl
Grosswiesenstrasse 167
CH-8051 Zürich
Switzerland
ambuhl@icloud.com

MeteoSchweiz
Krähbühlstrasse 58
CH-8044 Zürich

T +41 44 256 91 11
www.meteoschweiz.ch

MeteoSchweiz
Flugwetterzentrale
CH-8060 Zürich-Flughafen

T +41 43 816 20 10
www.meteoswiss.ch

MeteoSvizzera
Via ai Monti 146
CH-6605 Locarno Monti

T +41 91 756 23 11
www.meteosvizzera.ch

MétéoSuisse
7bis, av. de la Paix
CH-1211 Genève 2

T +41 22 716 28 28
www.meteosuisse.ch

MétéoSuisse
Chemin de l'Aérologie
CH-1530 Payerne

T +41 26 662 62 11
www.meteosuisse.ch