Time-minimal Path Planning in Dynamic Current Fields

Michaël Soulignac [1,2] Patrick Taillibert [1] Michel Rueher [2]

[1] THALES Aerospace, AI laboratory
2 Avenue Gay Lussac, 78852 Elancourt, FRANCE
{firstname.lastname}@fr.thalesgroup.com

[2] Nice Sophia Antipolis University, I3S/CNRS,
BP 145, 06903 Sophia Antipolis, FRANCE
michel.rueher@gmail.com

***Abstract*— Numerous approaches have been proposed for path planning in dynamic current fields, for a fixed departure time. However, in many applications, the departure time is not necessarily known in advance, but can vary in a time window. In this context, the choice of a good departure time is a critical issue. That is why we introduce in this paper a new approach, called *symbolic wavefront expansion*, determining both the path and the departure time minimizing the travel time of the vehicle. The key idea of this approach is to propagate and compose functions instead of numerical values, with appropriate operators.**

I. Introduction

Recent advances made in the field of autonomous vehicles suggest that, in a near future, Unmanned Air Vehicles (UAVs) or Autonomous Underwater Vehicles (AUVs) will be more and more deployed in order to achieve various missions such as surveillance, intelligence or search and rescue.

Moreover, since UAVs and AUVs may be slow, the impact of (air or water) currents is significant, and cannot be neglected.

Specific path planning approaches have been proposed to handle currents, based on evolutionary computation [1][9], wavefront expansion [4][8][10] or optimization techniques [5][7][11]. These approaches determine the path taking at most profit of currents, i.e. they maximize the parts of the path where the vehicle and the currents point in the same direction.

The main limit of these approaches is that they require fixing the departure time in advance. Indeed the departure time is required to know the state of the currents, which is necessary to evaluate the travel cost of the vehicle.

However, in many applications, the departure time may vary in a given time window. In such situations, the choice of an appropriate departure time is a critical issue: if it is improperly chosen, currents may be against the vehicle, regardless of the path planned.

That is why we propose here a new approach which generalizes the concept of wavefront expansion [3]. In this generalization, the wavefront contains functions instead of numerical values, as illustrated in figure 1. The initial algorithm is modified in order to manipulate these functions: numerical operators are replaced by symbolic ones. The symbolic operators we propose take advantage of the piecewise linearity of functions.

The resulting algorithm, that we called *symbolic wavefront expansion* (SWE), is able to determine both the path and the departure time (in a given time window), minimizing the travel time.

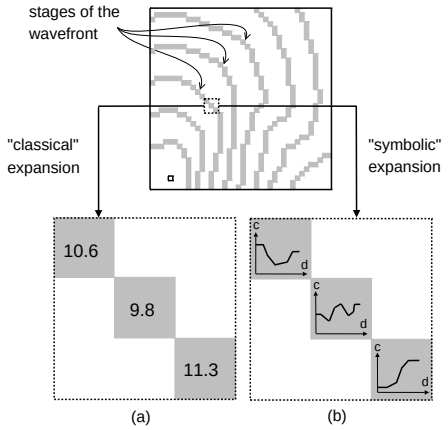

Fig. 1. Classical vs symbolic wavefront expansion. (a) costs are numbers; (b) costs are piecewise linear functions, represented by line coefficients.

At our knowledge, SWE is the only path planning algorithm designed for autonomous vehicles with this capability. Orda and Rom introduced in [6] an algorithm with similar properties in a very different context. This algorithm, called *Source Waiting 2* (SW2), has been designed to process a message in dynamic communication network.

However, experiments on actual wind chart show that, in the context in long duration missions within currents, SWE is much more efficient and scalable than SW2.

II. Problem Statement

A. Problem description

Our problem consists in finding both the path P and the departure time d^* (in a time window W) for an UAV, minimizing its travel time between two points A and B in a planar environment, containing space and time varying currents.

The magnitude of the velocity of the vehicle relative to the current is constant (cruise speed). Currents are known through k charts denoted $C_1, C_2, ..., C_k$, obtained by forecasting.

B. Problem formalization

Our problem is modeled as follows:

- The planar environment in which the vehicle evolves is modeled by a 2-D Euclidean space E. We denote

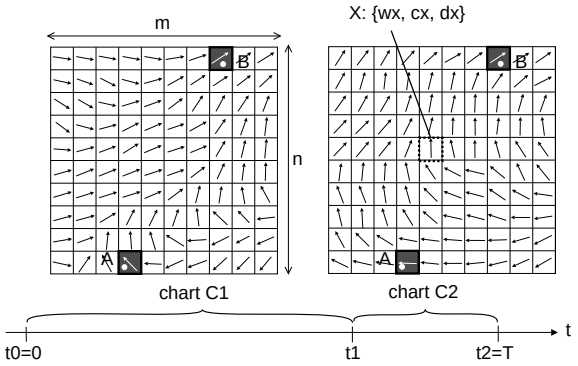

Fig. 2. A path planning problem in time-varying currents, with two currents charts C_1 and C_2.

$R = (O, \vec{x}, \vec{y})$ the frame embedded in E, (u_x, u_y) the coordinates of a vector $\vec{u}$ in R and u its magnitude.

- This environment is discretized using a grid G of size $m \times n$. Each cell X of this grid has the following attributes:
  - $\vec{w_X}$: the velocity vector of the (homogeneous) current in X.
  - c_X: the cost, i.e. the travel time, required to reach X from the start point A.
  - d_X: the best departure time (from A) to reach X, i.e. the departure time minimizing c_X.

  Note that the start and goal points A and B are approximated by the cells containing them.

- The time window W is defined by $[0, T]$. The bound 0 represents the beginning of the mission, and T the latest arrival time to B.

- The i^{th} current chart C_i defines the value of $\vec{w_X}$, applied in each cell of G, from time t_{i-1} to time t_i. By convention, we have $t_0 = 0$ and $t_k = T$. The transition between two successive charts is assumed to be instantaneous. This approximation is justified by the fact that current charts already include errors. It is thus meaningless to model a smooth transition between them. However, it would be possible to insert some artificial current charts between existing ones, using interpolation techniques.

- The vehicle's velocity relative to the current $\vec{w}$ is denoted $\vec{v}$. Its magnitude v is assumed constant and called cruise speed.

Using these notations, our problem consists in determining:

1) The departure time $d^* \in [0, T]$ from cell A,
2) The path $P = \{A, X_1, X_2, ..., X_l, B\}$, i.e. the sequence of $l + 2$ cells,

minimizing the travel time between A and B.

III. The wavefront expansion

A. Description

The wavefront expansion was first proposed by Dorst and Trovato [3]. It consists in iteratively expanding a wavefront F from the cell A, within the grid G, until it has reached the cell B. An expansion step consists in extracting the cell H for which the cost c_H is minimal and then evaluating its neighbors $\mathcal{N}(H)$, using a metric $\mathcal{M}$. The corresponding algorithm is provided in figure 3.

Once this expansion is performed, the optimal path is built by applying the hill-climbing algorithm, which is a discrete version of the steepest descent. This algorithm consists in starting from B and iteratively moving towards the cell of lowest cost, until A is reached.

Generally, $\mathcal{N}$ is the Moore neighborhood and $\mathcal{M}$ the Euclidean distance, but others choices are possible to handle constraints related to the environment.

In particular, some adaptations have been proposed to handle currents [4][8][10]. In these adaptations, the metric $\mathcal{M}$ is defined to integrate the influence of currents on the travel time. In particular, in [10], we proposed the following metric, between two cells X and Y:

$$\mathcal{M}_{X,Y}(t) = \frac{\sqrt{\Delta} - \langle \vec{m} \cdot \vec{w(t)} \rangle}{v^2 - w(t)^2} \tag{1}$$

where $\Delta = v^2 \cdot (m_x^2 + m_y^2) - (w_x(t) \cdot m_y - w_y(t) \cdot m_x)^2$, $\vec{w}(t)$ represents the velocity vector of the current between X and Y at time t, and $\vec{m} = \vec{XY}$ the move between X and Y.

WAVEFRONT_EXPANSION(A, B, G, d)
- ▷ **Input** A, B: start and goal cells
- ▷ **Input** G: grid
- ▷ **Input** d: departure time in A
- ▷ **Local** F: current wavefront
- ▷ **Local** $\underline{F}$: fully expanded cells
- ▷ **Local** H: head of F (cell of lowest cost)
- ▷ **Local** N: a neighbor of H
- ▷ **Local** c_N^{temp}: temporary evaluation of N

Begin
- $F \leftarrow \{A\},\ \underline{F} \leftarrow \emptyset$
- $c_X \leftarrow +\infty, \forall X \in G \setminus \{A\}$
- $c_A \leftarrow 0$
- **do**
  - $H \leftarrow \arg\min\{c_X, X \in F\}$
  - **for each** $N \in (\mathcal{N}(H) \setminus \underline{F})$ **do**
    - $c_N^{temp} \leftarrow c_H + \mathcal{M}_{H,N}(c_H + d)$
    - $c_N \leftarrow \min\{c_N, c_N^{temp}\}$
    - $F \leftarrow F \cup \{N\}$
  - $F \leftarrow F \setminus \{H\}$
  - $\underline{F} \leftarrow \underline{F} \cup \{H\}$
- **while** $H \neq B$

End

Fig. 3. The classical wavefront expansion, adapted from [3].

This metric incites the vehicle to point in the same direction than the current: if the scalar product $\langle \vec{m} \cdot \vec{w(t)} \rangle$ increases, the value of $\mathcal{M}$ decreases. Thus, during its expansion, the wavefront explores first the parts of the environment where the cost is the lowest, i.e. where currents push the vehicle. This leads to an anisotropic wavefront expansion which gives shorter paths (in terms of travel time) than an isotropic one.

However, at each iteration of this expansion, the arrival time t to cell H (the head if the wavefront) depends on the departure time d from cell A. We indeed have $t = c_H + d$. Thus d has to be provided to the algorithm, as an input.

In our problem, d may vary in the time window W. In other terms, the value of d is not known. Moreover, the choice of a good value for d may be critical.

That is why we introduce here a new approach, called *symbolic wavefront expansion* (SWE), which is a generalization of the algorithm of figure 3. SWE is able to determine both the path and the value of d (in W), minimizing the travel time from A to B.

IV. The Symbolic Wavefront Expansion (SWE)

In SWE, the cost c_X associated to a cell X is not a numerical value anymore, but a function of the departure time d. Therefore, the algorithm described in figure 3 has been modified to manipulate these functions instead of numerical values. The resulting algorithm is given in figure 4.

SYMBOLIC_WAVEFRONT_EXPANSION(A, B, G, W)
- ▷ **Input** A, B: start and goal cells
- ▷ **Input** G: grid
- ▷ **Input** W: time window for the departure time in A
- ▷ **Local** F: current wavefront
- ▷ **Local** H: head of F (cell of lowest cost)
- ▷ **Local** N: a neighbor of H
- ▷ **Local** c_N^{temp}: temporary evaluation of N

1 **Begin**
2 $F \leftarrow \{A\}$
3 $c_X \leftarrow +\infty, d_X \leftarrow 0, \forall X \in G \setminus \{A\}$
4 $c_A \leftarrow 0, d_A \leftarrow 0$
5 **do**
6 $H \leftarrow \arg\min\{c_X(d_X), X \in F\}$
7 **for each** $N \in \mathcal{N}(H)$ **do**
8 $c_N^{temp} \leftarrow c_H + \mathcal{M}'_{H,N} \circ (c_H + d), d \in W$
9 $c_N^{temp} \leftarrow \min\{c_N, c_N^{temp}\}$
10 **if** $c_N^{temp} \neq c_N$
11 $c_N \leftarrow c_N^{temp}$
12 $d_N \leftarrow \min\{c_N(d), d \in W\}$
13 $F \leftarrow F \cup \{N\}$
14 $F \leftarrow F \setminus \{H\}$
15 **while** $T \neq B$
16 **End**

Fig. 4. The symbolic wavefront expansion (SWE)

Note that this algorithm coincides with the one of figure 3 in the case of constant functions (since a constant function is modeled by a single numerical value). That is why we present SWE as a generalization of the wavefront expansion.

The main differences between algorithms of figures 3 and 4 are the following ones:

- Line 8 (Metric). The metric $\mathcal{M}'_{H,N}$, measuring the travel time between H and N, does not return a numerical value anymore, but a function of time.
- Line 8 (Evaluation). The previous change has two consequences on the evaluation of N:
  - The composition operator "$\circ$" is introduced,
  - The addition operator "$+$" does not add two numerical values anymore, but two function.
- Line 9 (Comparison). Similarly to the addition, the minimum operator "$\min$" compares now two functions.
- Line 12 (Minimization). This new operation consists in searching the minimum of the function c_N, in order to set the attribute d_N.
- Line 13 (Wavefront update). Contrary to the classical wavefront expansion, the cell N can be introduced several times in the wavefront.

A. Introducing a new metric

Consider the move between the head of the wavefront H and one of its neighbors N. Let us study the effect of the last current change, from the chart C_{k-1} to the chart C_k, on the travel time. If we denote t the arrival time to cell H, we have:

- If $t \geq t_k = T$ the move $H \rightarrow N$ is entirely performed in the chart C_k. Therefore, the travel time between H and N is equal to:
$$\tau_k = \mathcal{M}_{H,N}(t_k)$$
$\mathcal{M}_{H,N}$ is defined by equation 1.
- If $t \in [t_{k-1}, t_k - \mathcal{M}_{H,N}(t_{k-1})]$ the move $H \rightarrow N$ is entirely performed in the chart C_{k-1}. The travel time between H and N is thus:
$$\tau_{k-1} = \mathcal{M}_{H,N}(t_{k-1})$$
- Finally, if $t \in [t_k - \mathcal{M}_{H,N}(t_{k-1}), t_k]$, the current changes during the move $H \rightarrow N$. In other words, the beginning of the move is performed in the chart C_{k-1}, and the end is performed in C_k. The travel time between H and N is given by:
$$\tau_{k-1,k} = \left(\frac{\tau_k}{\tau_{k-1}} - 1\right) t + \left[\tau_k - \tau_k \cdot \left(\frac{\tau_k}{\tau_{k-1}} - 1\right)\right]$$
$\tau_{k-1,k}$ can be seen as a *transition phase* between τ_{k-1} and τ_k.

Then, making an induction from $i = k$ to $i = 1$, we can define a new metric $\mathcal{M}'_{H,N}$ by:

$$\mathcal{M}'_{H,N} : t \mapsto \mathcal{M}_{H,N}(t_k) \quad \text{if} \quad t \geq t_k \tag{2}$$

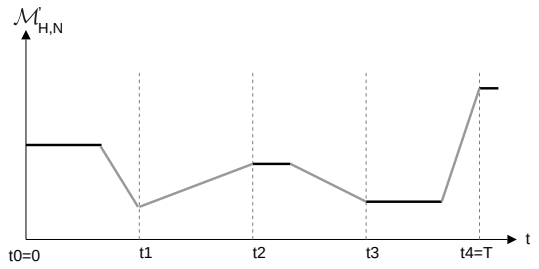

Fig. 5. Graph of the metric $\mathcal{M}'_{H,N}$, with 4 current changes in $[0, T]$.

And:

$$\forall i \in [1,k]: \quad \mathcal{M}'_{H,N} : t \mapsto \begin{cases} \tau_{i-1} & \text{if } t \in [t_{i-1}, t_i - \tau_{i-1}] \\ \gamma_i \cdot t + [\tau_i - t_i \cdot \gamma_i] & \text{if } t \in [t_i - \tau_{i-1}, t_i] \end{cases} \tag{3}$$

with $\tau_i = \mathcal{M}_{H,N}(t_i)$ and $\gamma_i = \left(\frac{\mathcal{M}'_{H,N}(t_i)}{\mathcal{M}_{H,N}(t_{i-1})} - 1\right)$.

The graph of $\mathcal{M}'_{H,N}$ is plotted in figure 5 for 4 current changes. In general, this kind of graph is made up of successive levels (in dark grey) and transition phases (in light grey). More precisely, between t_i and t_{i+1}, there is one level and one transition phase[1].

B. Redefining the evaluation operation

The evaluation operation is used to compute the cost c_N of a new cell N, coming from H (already evaluated).

In the classical wavefront expansion, c_N is computed thanks the following steps:

1) c_H represents, by definition, the cumulated travel time from A to H,
2) $c_H + d$ represents the arrival time to H (since d is the departure time from A),
3) $\mathcal{M}_{H,N}(c_H + d)$ represents the local travel time from H to N,
4) $c_H + \mathcal{M}_{H,N}(c_H + d)$ finally represents the cumulated travel time from A to N.

Since costs are numerical values, all additions are numerical operations.

In the symbolic wavefront expansion, c_N is computed in the following manner:

1) c_H represents, by definition, the cumulated travel time from A to H,
2) $c_H + d$ represents the arrival time to H (since d is the variable modeling the departure time from A),
3) $\mathcal{M}'_{H,N} \circ (c_H + d)$ represents the local travel time from H to N,
4) $c_H + \mathcal{M}'_{H,N} \circ (c_H + d)$ finally represents the cumulated travel time from A to N.

[1]If $\mathcal{M}_{H,N}(t_i)$ is bigger than the duration of the chart C_i (i.e. $t_{i+1} - \mathcal{M}_{H,N}(t_i) < t_i$), then no level will be present between t_i and t_{i+1}. In figure 5, it is the case for $i = 1$.

These steps seem very similar to those performed in the classical wavefront expansion. However, since costs are functions and not numerical values anymore, the nature of each step is very different.

First, in step 2, the addition $c_H + d$ is symbolic. It means that, if c_H is defined by $d \mapsto a \cdot d + b$, then $c_H + d$ is defined by:

$$c_H + d : d \mapsto (a+1) \cdot d + b \tag{4}$$

Next, in step 3, the evaluation of $\mathcal{M}_{H,N}$ at time $c_H + d$ is replaced by the composition of $\mathcal{M}'_{H,N}$ with the function $c_H + d$. If $\mathcal{M}'_{H,N}$ is defined by $t \mapsto a' \cdot t + b'$, then $\mathcal{M}'_{H,N} \circ (c_H + d)$ is defined by:

$$\mathcal{M}'_{H,N} \circ (c_H + d) : d \mapsto a'(a+1) \cdot d + [a'b + b'] \tag{5}$$

Finally, in step 4, the addition of the previous function with c_H is also symbolic (as in step 2). We obtain that c_N is defined by:

$$c_N : d \mapsto [a'(a+1) + a] \cdot d + [b(a'+1) + b'] \tag{6}$$

All these steps are illustrated in figure 6. Let us detail this example:

1) c_H is defined by:

$$c_H : d \mapsto \begin{cases} 1 & \text{for} \quad d \in [0,5] \\ 3 \cdot d - 14 & \text{for} \quad d \in [5,6] \\ 4 & \text{for} \quad d \in [6,10] \end{cases}$$

2) We deduce that $c_H + d$ is defined by:

$$c_H + d : d \mapsto \begin{cases} d + 1 & \text{for} \quad d \in [0,5] \\ 4 \cdot d - 14 & \text{for} \quad d \in [5,6] \\ d + 4 & \text{for} \quad d \in [6,10] \end{cases}$$

3) Composing this last function with the metric $\mathcal{M}'_{H,N}$ defined by:

$$\mathcal{M}'_{H,N} : t \mapsto \begin{cases} 3 & \text{for} \quad t \in [0,3] \\ -1/3 \cdot t + 4 & \text{for} \quad t \in [3,6] \\ 2 & \text{for} \quad t \in [6,10] \end{cases}$$

We obtain:

$$\mathcal{M}'_{H,N} \circ (c_H + d) : d \mapsto \begin{cases} 3 & \text{for} \quad d \in [0,2] \\ -1/3\,t + 4 & \text{for} \quad d \in [2,5] \\ 2 & \text{for} \quad d \in [5,10] \end{cases}$$

4) c_N is finally obtained by adding c_H:

$$c_N : d \mapsto \begin{cases} 4 & \text{for} \quad d \in [0,2] \\ -1/3 \cdot d + 14/3 & \text{for} \quad d \in [2,5] \\ 3 \cdot d - 12 & \text{for} \quad d \in [5,6] \\ 6 & \text{for} \quad d \in [6,10] \end{cases}$$

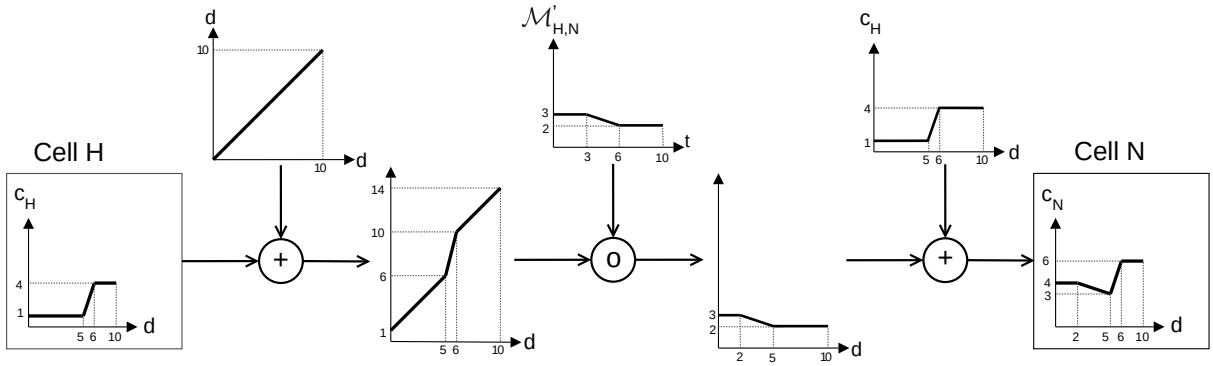

Fig. 6. Evaluation operator in the symbolic wavefront expansion.

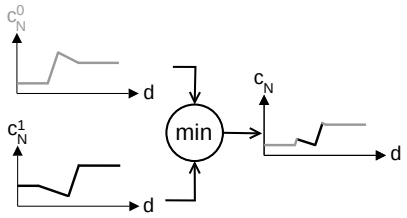

Fig. 7. Comparison operator in the symbolic wavefront expansion.

C. Redefining the comparison operation

The comparison operation is used to choose between two concurrent sources H_0 and H_1 for a same destination N, by comparing the costs c_N^0 and c_N^1 associated to the moves $H_0 \rightarrow N$ and $H_1 \rightarrow N$.

In the classical wavefront expansion, this operation consists in comparing the two numerical values c_N^0 and c_N^1. The move $H_i \rightarrow N$ leading to the smallest value c_N^i will be selected to reach N. The other move, $H_{1-i} \rightarrow N$, is definitively given up.

In the symbolic wavefront expansion, this comparison is performed for all possible departure times d. By this way, we build a new function c_N defined by $d \mapsto \min\{c_N^0(d), c_N^1(d)\}$. This combination is illustrated in figure 7. For each value of d, a backpointer towards the optimal predecessor H^i (for which $c_N^i(d)$ is minimal) is stored.

According to this principle, the optimal move to reach N can alternate. In figure 7, for instance, it is first $H_0 \rightarrow N$ (grey part of c_N), then $H_1 \rightarrow N$ (black part) and finally $H_0 \rightarrow N$.

To perform this operation, we can proceed as follows:

- Initialize c_N at $t = 0$: $c_N = c_N^i$ if $c_N^i(0) < c_N^{1-i}(0)$, else $c_N = c_N^{1-i}$.
- At each intersection at $d = d_I$, between the graph of c_N^0 and c_N^1, change the definition of c_N: if c_N was equal to c_N^i before d_I, c_N is equal to c_N^{1-i} after d_I.

D. Locating the minimum

As explained before, each function c_X is defined by successive line segments $[M_i, M_{i+1}]$, with $M_i = (d_i, c_X(d_i))$. The minimum of c_X can be easily computed by enumerating its segments, and locating the value of d_i for which the value of $c_X(d_i)$ is minimal.

E. Updating the wavefront

In our approach, each cell X of the wavefront F is represented by the minimum of its cost function $c_X(d_X)$. This allows to maintain an order in F, but this also lead to a loss of information about the initial function c_X (image domain, slope variation, etc.). In those conditions, we cannot guarantee that the cost of an expanded cell cannot be improved in the future. Therefore, contrary to the classical wavefront, a cell may be expanded several times, each time some changes in its cost function are detected.

F. Building the solution

- Optimal departure time:
  By construction, the departure time minimizing the travel time from A to B is the minimum of the cost function c_B. The location of this minimum is known, and stored in the attribute d_B. Thus, the optimal departure time is $d^* = d_B$, as illustrated in figure 8.
- Optimal path:
  Once d^* is known, the optimal path is built by following backpointers stored during the cost propagation, as performed in figure 9.

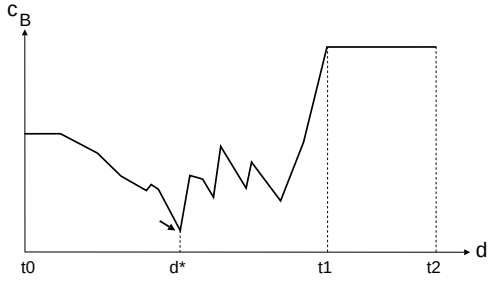

Fig. 8. The cost function c_X and the optimal departure time d^* for the problem of fig. 2

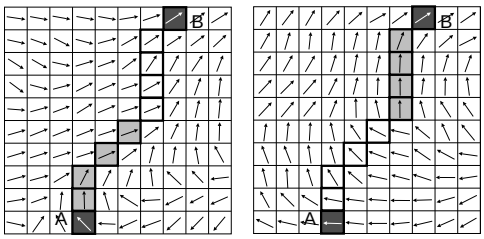

Fig. 9. The optimal path for the problem of fig. 2. The departure of the vehicle has been delayed such that, in each chart, the vehicle is "pushed" by currents (grey parts).

V. Performance Analysis

A. Complexity study

Let us evaluate the number of operations required by SWE (given in figure 4) at each iteration i:

- Line 6: extracting the cell of minimum cost from the wavefront requires $O(1)$ operations with an appropriate data structure, such as a heap [2].
- Lines 8, 9 and 12: computing the cost function of a cell and locating its minimum requires $O(k \cdot i)$ operations.
- Line 11: affecting the cost function of a cell requires $O(1)$ operations, using a pointer or a reference.
- Line 13: maintaining the order of the wavefront F, after inserting a new element, requires $O(\log i)$ operations (using a heap).
- Line 14: deleting an element in F requires $O(1)$ operations.

To sum up, the i^{th} iteration requires at most $O(k \cdot i)$ operations.

Moreover, as we can see in line 10, at the i^{th} iteration, if the cost function of a cell has changed, it is re-introduced in the wavefront. This implies that this cell can be potentially evaluated i times.

If we denote $N = n \cdot m$ the number of cells in the grid G, the maximal number of operations required by the whole algorithm is:

$$O\left(\sum_{i=1}^{N} i \cdot (k \cdot i)\right) = O(k \cdot N^3)$$

This seems much greater than the time complexity of the classical wavefront expansion, which is in $O(N \log N)$. However, it is meaningless to compare them, because they produce totally different results: the classical wavefront expansion computes the optimal path for a single (and known) departure time, whereas the symbolic wavefront expansion computes optimal paths for all possible departure times in the time window W.

At our knowledge, the only algorithm with the same capability is the SW2 algorithm, proposed by Orda and Rom [6] in the field of communication networks. SW2 has a time complexity in $O(f \cdot V \cdot E)$, where f is the number of elementary operations to evaluate the cost function of a node (by comparison and composition); E and V are respectively the number of nodes and arcs in the communication network. In our problem, grid cells play the role of communication nodes, thus we have $V = N$ and $E = O(N)$. Then, since $f = O(k \cdot N)$, we finally obtain a time complexity in $O(k \cdot N^3)$.

In the worst case, SWE and SW2 have the same behavior. However, in the context of our application (an UAV moving in forecasted winds), we observed that our approach is much more efficient than SW2. This point is shown experimentally in the next section.

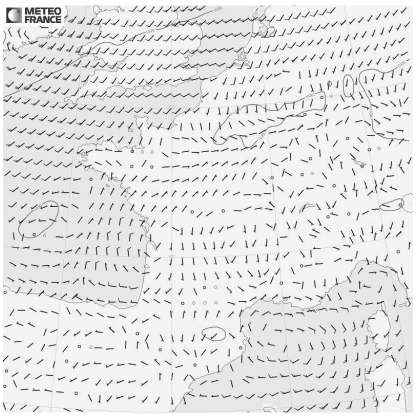

Fig. 10. An example of wind chart used to generate test-cases.

B. Experimental results

In this part, we compare the performance of SWE and SW2, in the context of a long range mission for a UAV: several hours and thousands miles of flight, in presence of time-varying winds.

1) Test-cases generation: We collected weather forecast data on the Météo France website[2] to constitute a database of 100 consecutive wind charts. Each wind chart is valid during 6 hours. Using this database, each test-case was generated as follows:

1) Start and goal points A and B were randomly placed in the environment;
2) The first wind chart and the number of wind changes $k \in [1, 5]$ were also randomly chosen. Each wind chart was applied in the environment during 6 hours. Thus the time window for departure was $W = [0, 6k]$.

2) Performance comparison: To compare the performance of SWE and SW2, we applied both approaches on 500 test-cases for different values of N (from $N = 1000$ to 10000, with a step of 1000). Then, we computed the average number of evaluated cells as a measure of the computational effort. Results are plotted in figure 11.

[2]http://marine.meteofrance.com/marine/accueil?19904.path=marine%252Fimgmervent

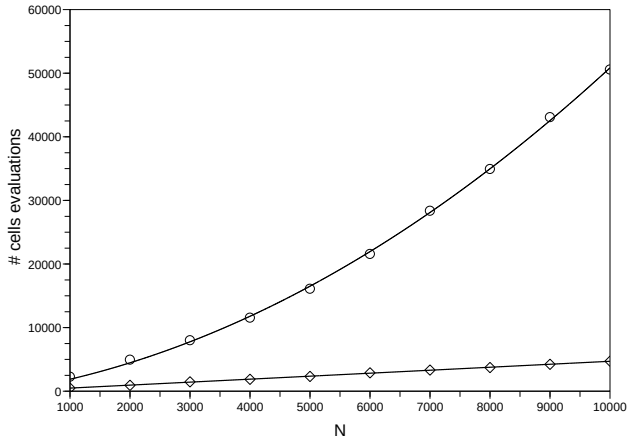

Fig. 11. Number of cells evaluations vs total number of cells in the grid (N). Diamonds = SWE, circles = SW2.

By performing regressions on data, we determined that the number of evaluated cells is in $O(N)$ for SWE, and in $O(N^2)$ for SW2 (with a coefficient of determination $R^2 > 0.999$).

SWE appears to be much more efficient than SW2 on our test-cases. Indeed, for $N = 10000$ (i.e. a 100×100 grid), our approach is 10 times faster, and this ratio seems to increase linearly with N.

This difference can be explained by the nature of the algorithms. SWE is a Dijkstra-like algorithm, whereas SW2 is a Ford-like algorithm. And the impact of a cell re-evaluation is completely different in these two families of algorithms.

Let us consider the re-evaluation of a cell X, which descendants are $X_1, ..., X_j$ (the neighbors of X, the neighbors of these neighbors, etc.).

In a Dijkstra-like algorithm, the cells of the wavefront are sorted by increasing values of cost. By construction, the cost of X is lower than the cost of its descendants, since costs are positive quantities. If X is re-injected in the wavefront, it means that its cost has been improved. Therefore, it is still lower than the cost of its descendants after re-introduction. In those conditions, the descendants of X will be "frozen" (they do not propagate any information) until the impact of the improvement of the cost of X has reached them.

In a Ford-like algorithm, the notion of wavefront does not exist. The whole grid is re-evaluated, while some changes are detected. If the cost of a cell X is improved, its descendants will continue to propagate sub-optimal costs to their neighbors, because they do not "wait" updated information about X. And all these sub-optimal costs will necessary imply some cells re-evaluations.

VI. Conclusions and future works

In this paper, we introduced the symbolic wavefront expansion, determining both the path and the departure time minimizing the travel time of a vehicle, in presence of dynamic current fields.

Our experiments on realistic environments suggest that our algorithm is significantly more efficient and scalable than a similar algorithm introduced by Orda and Rom in a very different domain.

We previously introduced in [10] the *sliding wavefront expansion*, able to handle strong currents (i.e when currents become faster than the vehicle). Further works will naturally consist in mixing the concepts of the symbolic and sliding wavefront expansions, in order to handle dynamic and strong currents in a unified way.

VII. Acknowledgements

The authors would like to thank the staff of the THALES AI laboratory for their careful reading of this paper.

References

[1] Alvarez, A., Caiti, A., and Onken, R. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *Journal of Oceanic Engineering 29* (2004), 418–429.

[2] Cormen, T., Leiserson, C., and Rivest, R. *Introduction to algorithms, Second Edition*. The MIT Press, 1998.

[3] Dorst, L., and Trovato, K. Optimal path planning by cost wave propagation in metric configuration space. In *Proceedings of SPIE-The International Society for Optical Engineering* (1988), pp. 186–197.

[4] Garau, B., Alvarez, A., and Oliver, G. Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach. In *Proceedings of the International Conference on Robotics and Automation* (2005), pp. 18–22.

[5] Kruger, D., Stolkin, B., Blum, A., and Briganti, J. Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments. In *Proceedings of the International Conference on Robotics and Automation* (2007), pp. 4265–4270.

[6] Orda, A., and Rom, R. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM 37* (1990), 607–625.

[7] Pannequin, J., Bayen, A., Mitchell, I., Chung, H., and Sastry, S. Multiple aircraft deconflicted path planning with weather avoidance constraints. *Proceedings of AIAA Guidance, Navigation and Control Conference* (2007).

[8] Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J., and Lane, D. Path planning for autonomous underwater vehicles. *Transactions on Robotics 23* (2007), 331–341.

[9] Rubio, J., and Kragelund, S. The trans-pacific crossing: long range adaptive path planning for UAVs through variable wind fields. In *Proceedings of the Digital Avionics Systems Conference* (2003), vol. 2, pp. 8.B.4.1–12.

[10] Soulignac, M., Taillibert, P., and Rueher, M. Adapting the wavefront expansion in presence of strong currents. *Proceedings of International Conference on Robotics and Automation* (2008), 1352–1358.

[11] Zhang, W., Inanc, T., Obaum, S. O.-B., and Marsden, J. E. Optimal trajectory generation for a glider in time-varying 2D ocean flows B-spline model. In *Proceedings of the International Conference on Robotics and Automation* (2008), pp. 1083–1088.