

# Rapport de projet MPA : Outil d'aide à la décision pour le routage multi-objectifs de voiliers de course

## I. Domaine

Le but de la course au large, comme toute course, est d'arriver le premier et le plus vite possible à un point d'arrivée défini en partant d'un point de départ défini. Mais contrairement aux autres types de courses, le coureur doit prendre en compte plusieurs paramètres si il veut arriver à son objectif. En effet, les caractéristiques physiques du bateau ainsi que la météo doivent être pris en compte au minimum, afin de permettre de définir la route la plus adaptée jusqu'à l'arrivée. Par exemple, il peut être plus judicieux de faire un léger détour plutôt qu'emprunter une route directe afin de bénéficier de vents favorables et ainsi être plus rapide. C'est ce qu'on appelle faire du routage. Néanmoins, emprunter une route pour être plus rapide n'est pas nécessairement le choix le plus judicieux, cela pourrait générer beaucoup de stress (fatigue physique et psychologique pour le coureur et matérielle pour le bateau) et diminuer les chances d'atteindre l'arrivée dans de bonnes conditions. Lorsque le stress est lui aussi pris en compte, on parle alors de routage multi-objectifs.

## II. Problématique

Le but principal de ce projet est d'aider les coureurs au large en rendant le routage multi-objectifs plus visuel et ainsi les aider dans le choix de la route qui leur conviendra le mieux. Cela passe tout d'abord par l'intégration d'un outil de routage déjà développé au sein d'un outil de cartographie déjà existant lui aussi. Le programme de routage

nécessitant des paramètres et des fichiers spécifiques pour fonctionner (données météo, polaires de vitesse, coordonnées et date de départ...), le but est donc de pouvoir permettre la récupération de ceux-ci directement via l'application de cartographie qui possède une interface graphique. Une fois ces différents paramètres collectés, il faudra ensuite faire appel au programme de routage puis afficher les différentes routes calculées sur l'application de cartographie. L'étape suivante consiste à permettre l'affichage de données relatives aux différentes routes calculées (comme le stress) afin de permettre aux coureurs de pouvoir choisir la route la plus adaptée.

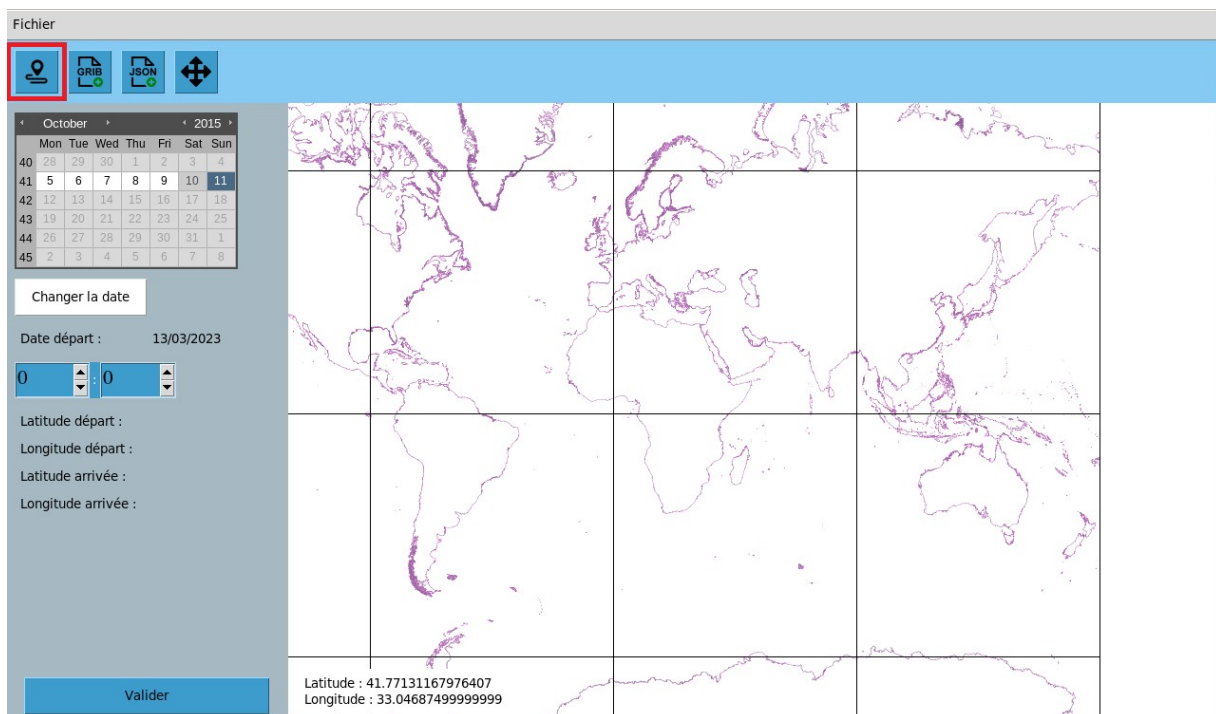
### III. Outillage et technologies

L'application de routage est développée en langage C et était au début basée sur l'API GRIB\_API, qui n'était plus maintenu à jour. Un passage vers l'API Eccodes a donc été effectué au tout début du projet. L'application de cartographie est quant à elle développée en python. La nouvelle API Eccodes nécessitant une installation qui fait appel à des outils fonctionnant mal avec le système d'exploitation Windows, il a été décidé d'utiliser une machine virtuelle Linux Debian grâce à l'outil VirtualBox afin que la partie de routage puisse fonctionner sans soucis. Les développements ont été effectués à l'aide de l'IDE Spyder.

### IV. Développement

L'application ayant été en partie remodelée par K. Le Loeuff qui travaillait aussi en parallèle dessus, elle est basée sur le design pattern composite. Ce qui veut dire que chaque grande fonctionnalité possède un fichier dédié (leafs), et que ces différentes fonctionnalités sont ensuite appelées dans des fichiers plus centraux qui permettent de faire fonctionner l'application (composite). Par exemple les fonctions de déplacement sur la carte se trouvent dans le fichier move.py. Il a donc tout naturellement été décidé de créer un fichier route.py regroupant

toutes les fonctions relatives à la partie de routage. Le premier ajout a été de permettre l'enregistrement de fichiers grib (pour les fichiers météo) et xml (pour les polaires de vitesse) directement via l'interface de l'application de cartographie. La partie routage comportant des programmes permettant la conversion de ces types de fichiers en type de fichiers acceptés pour le routage, le but était donc d'enregistrer puis de convertir les fichiers choisis. Il a fallu pour cela faire appel aux algorithmes en langage C de la partie routage directement depuis l'application Python en utilisant la fonction call de la bibliothèque subprocess. Une fois ceci effectué, un encart pour la partie routage a été ajouté (voir figure ci-dessous), ainsi que le bouton d'accès à cet encart (voir figure).



Par un simple clic sur ce bouton, l'encart de routage s'ouvre et on y retrouve :

- Un Calendrier permettant de choisir la date de départ
- Un Picker permettant de choisir l'heure de départ
- Une zone permettant de visualiser les coordonnées de départ et d'arrivée choisies. Un simple clic sur la carte permet de sélectionner des coordonnées de départ, un deuxième des coordonnées d'arrivée, puis un troisième de modifier celles de départ etc.

- Un bouton de validation permettant de lancer le routage une fois que tous les paramètres ont été sélectionnés.

Les fichiers de données météorologiques couvrant des périodes et des surfaces limitées, il a ensuite fallu mettre en place des restrictions et des vérifications afin de vérifier que différentes conditions étaient satisfaites (elles seront détaillées dans la prochaine partie). Une fois celles-ci mises en place, ce qui garantissait que le routage pouvait se dérouler sans soucis, il a été possible de mettre en place la partie appel à l'algorithme de routage. Cet algorithme générant des fichiers en sortie, il a ensuite fallu mettre en place une fonction permettant de lire dans ces fichiers les différentes routes calculées puis de les dessiner à l'écran. Pour cela l'outil de conversion de coordonnées en pixels déjà présent car développé par K. Le Loeuff s'est avéré très utile. Avec cet ajout, la partie routage est maintenant fonctionnelle.

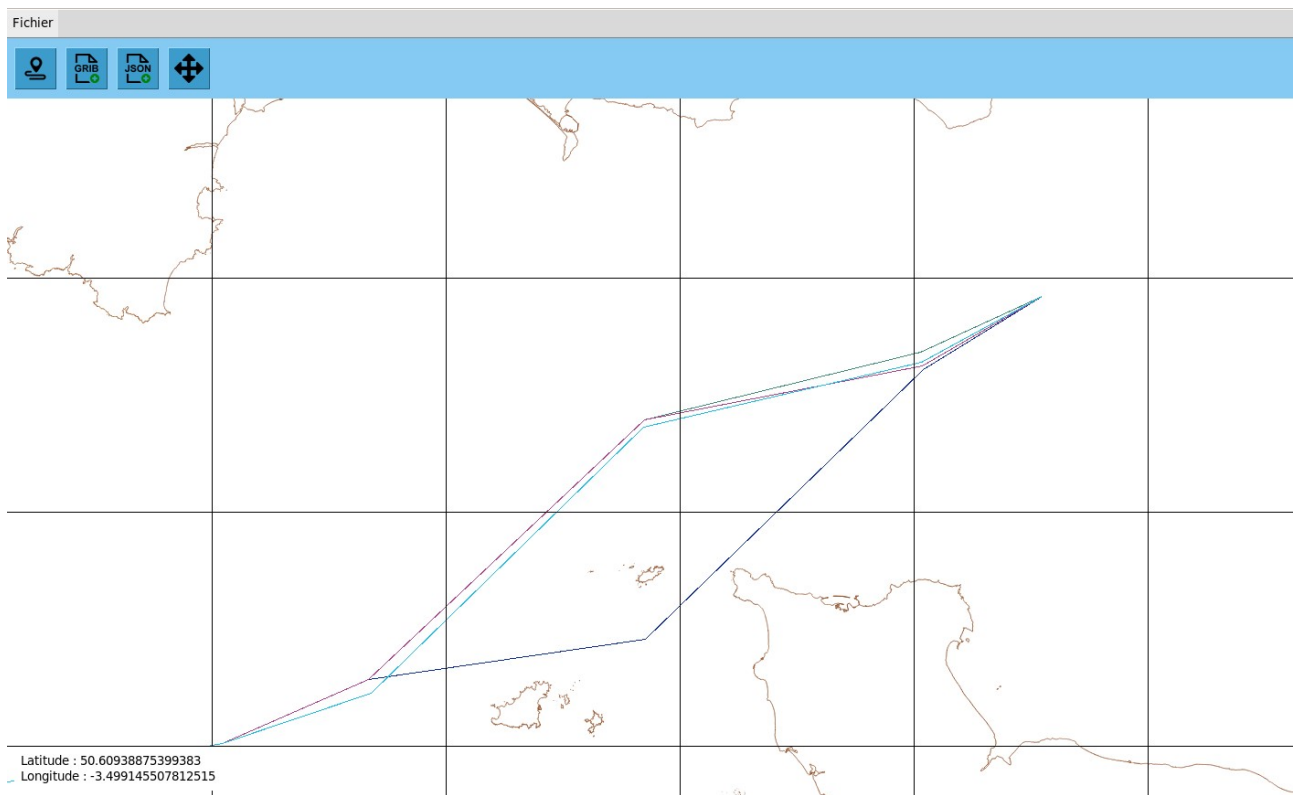
## V. Résultats

Le projet final rendu permet donc d'ajouter des fichiers météo et de polaires de vitesse et de les convertir au bon format, de sélectionner une date et une heure de départ pour le routage ainsi que des coordonnées de départ et d'arrivée. Une fois ceux-ci sélectionnés, il permet d'afficher les routes calculées sur la carte et des les différencier via des couleurs différentes (voir figure plus bas).

Afin de garantir le bon fonctionnement de l'application sans bugs. Des restrictions et vérifications ont été mises en place afin que le routage ne se lance pas dans de mauvaises conditions. Ainsi, il n'est pas possible de lancer un routage lorsque :

- Il manque un fichier de données météorologique
- Il manque un fichier de polaires de vitesse
- Des coordonnées ne sont pas sélectionnées
- La date choisie n'est pas couverte par le fichier météo
- L'heure choisie n'est pas couverte par le fichier météo

- Les coordonnées de départ/d'arrivée ne sont pas couvertes par le fichier météo



## VI. Discussion

Au cours de ce projet plusieurs difficultés ont été rencontrées, dont certaines ont causé du retard dans les développement. Par exemple un soucis de compilation de la partie routage qui n'a jamais pu être résolu. Au final, seule la partie routage a pu être développée. Il resterait donc à développer la partie aide à la décision, qui consisterait à afficher les données relatives à chaque chemin, soit dans l'encart de routage déjà développé soit via une fenêtre pop up. Ces deux solutions s'afficheraient lors du clic sur un chemin, qui serait lui affiché en surbrillance, ce qui ferait gagner en lisibilité et aiderait le marin dans son choix. Il faudrait aussi rajouter la conversion en .isoc lors de l'enregistrement de l'édition

d'un fichier grib. Il resterait aussi des améliorations possibles, comme par exemple afficher le nom des fichiers de données météorologiques ou de polaires de vitesse, rajouter des paramètres supplémentaires dans l'encart de routage pour le rendre plus modulable ou encore permettre la saisie de coordonnées de départ/d'arrivée à la main (car les déterminer lors du clic sur la carte peut parfois manquer de précision). Au total, deux livraisons ont été effectuées lors du projet, l'une à la fin du deuxième jalon et l'autre pour le rendu final. Chaque livraison comportait le code ainsi qu'un manuel d'installation et un manuel d'utilisation de l'application.

## VII. Summary

The main goal of this project is to help sailors to choose the best path possible during an off shore race. During this particular type of race, there are multiple factors to take into consideration to determine the best path to the finish line. The first one is the physical features of the boat that is being used during the race, and the second and most important one, the weather, and especially the wind. Sometimes taking the most direct way is not the fastest solution as taking a little longer path can put the sailor in a position where the wind increase considerably his speed. Determining a path based on this parameters is called routing. Another objective can be considered to help choosing the best path, it's the physical and psychological exhaustion for the sailor and the material damage for the boat that choosing a certain path could cause. For example, choosing a way with really strong winds could certainly be the fastest one, but it could also be dangerous for both the sailor and his boat. Taking stress into account adds another objective to the routing, which is now called multi-objective routing.

The purpose of the project was to integrate a routing application, developed in C language to a mapping one, developed in Python. In this perspective, a routing section has been added to the mapping application, which allows users to :

- add data files that are mandatory for the routing (weather data from grib files and speed polar data from xml files)
- choose the date and hour of the beginning of the routing
- Choose coordinates of start/finish of the routing
- Check that the dates and coordinates that have been chosen are covered by the weather file chosen before
- Perform the routing by calling the routing algorithm
- Draw the pathes that got calculated by the algorithm