

Optimization of Genetic Material Collection Missions Using A Gis Approach Based on the Dijkstra Algorithm and its Implementation on A Web Platform

Fall El Hadji Malick Sy^{1,*}, Kehel Zakaria², Hajji Rafika³, Tine Moustapha Gning⁴

¹SATPORT Senegal, Dakar, Senegal

²International Center for Agricultural Research in the Dry Areas (ICARDA), Rabat, Morocco

³Hassan II Agronomic and Veterinary Institute, Rabat, Morocco

⁴Laboratory of Mechanics and Modeling L2M, University Iba Der Thiam of Thiès, Thiès, Senegal

*Corresponding author: syfall@satport.ma

Received December 15, 2024; Revised January 16, 2025; Accepted February 23, 2025

Abstract Spatial Distribution Modeling and Gap Analysis of species are based essentially on the environmental and socio-economic characteristics of the specimen in question. It is therefore essential to have a mechanism for visiting a large number of sites with a high probability of the presence of genetic material in the shortest possible time. The specific aim of this project is to address the optimization of itineraries for a genetic material collection mission, based primarily on the use of shortest-path search algorithms, in particular Dijkstra's algorithm. The results of this study make it possible to ensure access to the germplasm while rationalizing the resources deployed and, consequently, the financial costs incurred. The whole process of finding the optimal route was made viable by its integration into a platform as a Web application serving as a guide for a genetic material collection mission.

Keywords: Genetic material collection, Optimization, Gap Analysis, Route, Spatial model

Cite This Article: Fall El Hadji Malick Sy, Kehel Zakaria, Hajji Rafika, and Tine Moustapha Gning, "Optimization of Genetic Material Collection Missions Using A Gis Approach Based on the Dijkstra Algorithm and its Implementation on A Web Platform." *Journal of Geosciences and Geomatics*, vol. 13, no. 1 (2025): 23-30. doi: 10.12691/jgg-13-1-2.

1. Introduction

The International Center for Agricultural Research in the Dry Areas (ICARDA) is aware of the progressively significant loss of biodiversity caused by a variety of factors over the last few decades, and has taken a keen interest in the issue. Safeguarding germplasm has become a necessity to offset this loss. The strategy adopted is to target collection sites that contain the greatest number of species and the greatest genetic diversity. Their studies are now focusing on establishing mechanisms and procedures to guarantee rational access to germplasm for the various stakeholders. Implementing this mission will make it possible to add new local plant varieties and then fill in the gaps in existing collections. However, this preambular phase requires the deployment of significant material resources and is proving very costly. Optimizing it therefore becomes a priority. The planning of a collection mission takes into account every aspect of the search for and collection of genetic material. It begins with a strategy for identifying targeted collection regions, which is then put into practice.

By ICARDA, by designing a model to represent the geographical distribution of germplasm. This model

could be obtained using gap analysis which, by definition, is a specific tool guaranteeing the identification of geographical areas where there is a high probability of genetic diversity that does not exist in collections, but also of adaptation traits of interest.

Nevertheless, to explore the collection points, the collectors' journeys are generally based solely on the maps resulting from the pre-established model, which can also be a source of additional costs. This study aims to answer the following question: "How can we optimize the task of collecting genetic material, and in particular the routes taken when exploring collection sites?"

Based on the gap analysis, the general aim of our approach is to focus on the rationalization offered by models designed to predict the spatial distribution of germplasm with traits of interest, to optimize collection routes. By optimal route, we refer to the adoption of the shortest route ensuring maximum harvest of genetic material.

Apart from the introduction, which sets out the context and issues of the study, this document is structured in two main parts. The first part is devoted to the methodology and hardware (software) used in this study. The second is dedicated to the presentation and discussion of the results obtained. This section summarizing the methodological approach is provided, along with recommendations for subsequent research.

2. Materials and Methods

Analysis of the problem leads us to choose some of the methods developed in the literature, based on their suitability for each situation [1]. Beforehand, a set of processing sets is necessarily applied to the geographical distribution of the material to be collected in order to prepare the execution of the Dijkstra shortest path algorithm. [2] Implemented a geographic information system with Dijkstra's algorithm based on a mobile app: A model for disaster evacuation routes in the city of Padang, Indonesia.

The figure below shows the methodological flow chart for the study.

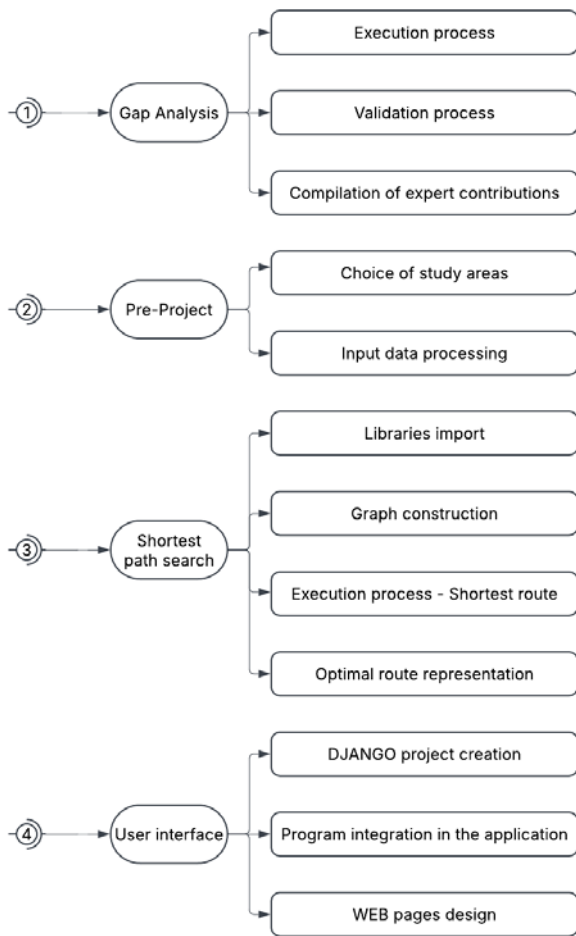


Figure 1. Methodology flow chart (picture changed)

2.1. Gap Analysis Procedure

The first step in optimizing the collection mission is to model the geographical distribution of the germplasm of interest. In the case of our project, this phase is carried out using the Gap Analysis approach. The results are a probability of presence map and a gap map on which further processing will be based.

This analysis tool starts with a determination of the so-called primitive breed groups and is preceded by a definition of the study area and a collection of data relating to the spatial prediction and characterization of accessions. This determination of races is essentially

based on a study of the existing literature and the classification resulting from the superposition of different results [3]. Figure 2 below illustrates the methodological procedure to be followed for a gap analysis intended for modelling as part of the optimization of a collection mission.

The modelling phase of the geographical distribution prediction is carried out using the **MaxEnt** [5]. Environmental and geographical gap scores are then calculated. It should be noted that the study area is generally considered on a large scale and may even cover the whole world. These scores are determined using specific scripts in **R** software, which will be presented in the following paragraphs. The Gap Score is then deduced from these previously calculated scores.



Figure 2. Gap analysis methodological process diagram [4]

$$Gap\ score = SDM * \left(\frac{Geo\ score + Env\ score}{2} \right)$$

The equation above is an illustration of the formula used to calculate the Gap Score. The first term of the product, SDM, represents the probability of finding a landrace in the region concerned. The second term characterizes the probability of finding a gap measured by the average between the geographical and environmental scores of equal weights.

Once the "Geo" and "Env" scores have been determined using the functions described in the previous paragraphs, the gap score, which we can call the global score, is deduced. The calculation stage, using the `calc_gap_score` function set up in the Gap Analysis program and all the required parameters, is of vital importance. This function represents the algorithmic translation of the gap score calculation based on the formula mentioned above

(equation 1). However, as a logical consequence, it must always be preceded by the determination of the probability of finding a landrace in the region concerned, noted SDM. The latter provides information on the prediction of the spatial distribution of the material being sought. However, it should be made clear that there are two possible scenarios when modeling the SDM. The approach to be adopted for calculating the SDM is influenced by the verification of a function called "*use.maxnet*".

If *use.maxnet* is set to "TRUE", then a function called *sdm_maxnet_approach_function* is executed: an approach considered to be only slightly faster than the next. This allows a complete evaluation of the models through cross-validation.

Secondly, if *use.maxnet* is set to "FALSE", then the MaxEnt Java application is launched. This application is slower and involves not only installing the correct version of Java but also an additional R package is required. However, the Java application must be stored in a specific folder to avoid a launch error.

2.2. Gap Analysis Validation Process

Any application that makes it possible to model a given phenomenon, particularly its probabilistic spatial distribution, needs to be validated. Gap analysis has played an important role in assessing the priorities for establishing new protected areas that will best fill the gaps in existing areas. The figure below gives an overview of the function used to validate it.

```
validation_process(occName = occName,
  gap_valDir = gap_valDir,
  buffer_radius = 1, # Radius of 100 km for ex
  density_pattern = 3, # Density pattern (1: 1d
  geo_score = c("cost_dist", "deLaunay"),
  use.Arcgis = FALSE,
  n.points = 5,
  doPar = TRUE, # whether or not parallelize
  use.maxnet = use.maxnet)
```

Figure 3. Validation process function for the Gap Analysis

This function creates all the files needed for the validation stage. It has to run the entire script more than five times. The time allowed is therefore quite considerable. However, it can be reduced if you have access to remote servers or a computer with several kernels and a large RAM. The next step is to reconfigure the *doPar* parameter to "TRUE". If the user only has a laptop with limited resources, we recommend setting the *doPar* to "FALSE".

2.3. Preliminary Draft of the Germplasm Collection Mission

2.3.1. Choice of Study Area

Because it is on a large scale, the modeling resulting from the gap analysis generally concerns the whole world or a fairly large area. While the collection mission can only concern a fairly restricted area about the global distribution obtained, it is, therefore, necessary to define a study area that will be the exclusive focus of the collection of genetic material. In practice, this choice is made according to several criteria. The first condition is the favorable agreement of the host country. A request for a

visit in the search for germplasm based on the "Gap Analysis" is sent to the administration of the land of destination. The result of the gap analysis must also be satisfactory at the level of the zone where the exploration has been programmed. It should be noted, however, that it is essential to define the main crop beforehand, as well as the type of germplasm desired for that crop.

In the case of our study, we theoretically chose a study area that does not meet all the above criteria but does allow us to implement the programs used in the rest of the project. We based ourselves on the "SDM" and the gap map, while identifying an area with a high probability of material presence. We also took into account the opinions of collectors from previous missions to some of the countries that had once agreed to be visited. In selecting the study area, these territories were given priority.

2.3.2. Input Data Processing

Although the results of gap analysis do not need to be georeferenced because they are in the same global WGS84 system, they do need to be adapted to the input parameters of the algorithms that follow them in the processing process. Gap Analysis returns raster data. The associated attribute tables can be converted to Excel tables in ArcGIS using the procedure shown in the figure opposite.

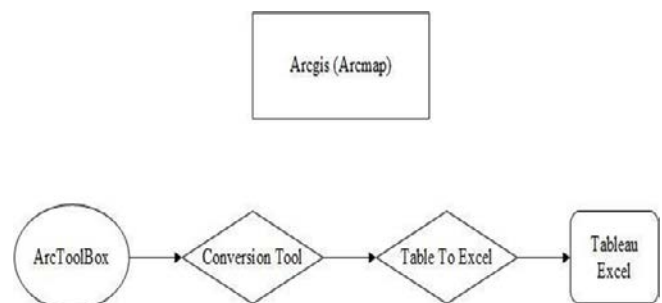


Figure 4. Converting an attribute table into an Excel table

Note: it is very important to check in the newly designed Excel table that the "id", "Latitude" and "Longitude" columns have been correctly written in the same way. It should also be noted that the "id" column should be generated starting from zero (0) up to the last entry in step one (1). This will prevent an error variable recognition during the execution of the algorithmic program for finding the optimal path.

The application of this program requires the installation of a set of packages that allow, among other things, the import of the necessary libraries and the execution of certain functions. The figure below gives an overview of the packages installed as part of our work.

Installations nécessaires

```
[ ] apt-get install libgeos-3.5.0
[ ] apt-get install libgeos-dev
[ ] pip install https://github.com/matplotlib/basemap/archive/master.zip

[ ] pip install ortools==6.7.4973

[ ] pip install openrouteservice
```

Figure 5. Installing the necessary packages

2.4. Application of the Dijkstra Algorithm

The Dijkstra algorithm commonly known as Dijkstra is used to solve the shortest path problem [6]. For a good adaptation, the backbone of the algorithm requires the pre-processing mentioned above. Indeed, the Dijkstra function has as input parameters a graph, the starting point, and the arrival point. The graph as defined in the literature is a set of nodes connected, but not necessarily all, by arcs. The distance between a node and its spouses is also inserted. It should also be noted that the characteristic points (start and finish) must belong to the graph. By way of illustration, we present two examples of numerical and graphical graphs.

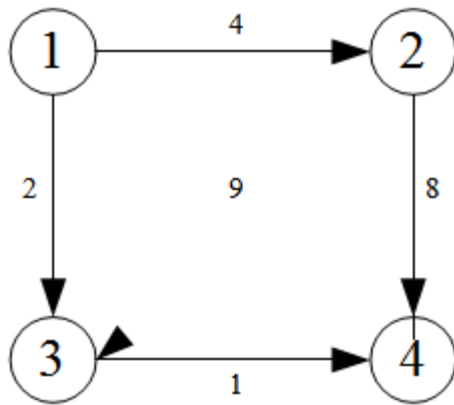


Figure 6. Example of a weighted directed graph [7]

In the case of our study, the programming of the algorithm in question was carried out using **PYTHON** scripts in Google Colab, following a certain methodological procedure which we will elucidate below. This choice was made to be able to test all of the scripts, which are divided into cells that depend on each other. Added to this is the ease of handling and the possibility of representing dynamic Leaflet maps that this platform offers.

2.4.1. Importing libraries

In IT terminology, a library can be defined as a collection of instructions that support a certain number of operations and produce a given result. This set of instructions is called a routine.

They are generally compiled for use in specific programs. Each library has a specific index enabling it to be called. Several libraries have been used for this project. Each of these libraries is used in the execution of a well-defined program. By way of illustration, the Basemap library (from "mpl_toolkits.basemap"), the matplotlib.pyplot library and the folium library are intended for the visualization and/or representation of data in a georeferenced geographical environment. As for the math, NUMPY, and PANDAS libraries, they were used respectively to use trigonometric formulas and to read an Excel file.

2.4.2. Constructing the Graph

The graph must be designed based on data relating to the geographical distribution of genetic material. In this case, the collection points are considered to be the nodes of the graph. The graph is obtained by calling up an Excel

file containing the identifier and geographical coordinates (longitude and latitude) of each collection point. In reading the Excel file, in particular, the identifier (id) and the geographical coordinates are followed simultaneously by the filling in of the graph dictionary. To do this, a sub-dictionary is created and noted $t\{\}$ is filled in for each node of the graph by all the nodes associated with it and also the distance separating them. The set of sub-dictionaries forms the graph referred to as the $\{\}$ graph in the code. The distance between two nodes is calculated using a function whose input parameters are their latitudes and longitudes.

2.4.3. Running the Shortest Path Algorithm

The algorithm for calculating the shortest path is applied by considering the graph thus designed. The literature has enabled us to distinguish between the different means of solving the shortest path problem and to choose the one best suited to our study. The authors' ideas converge on the fact that the Dijkstra algorithm is better than the Bellman-Ford algorithm in terms of complexity [8]. One of its advantages is the total exploration of all possible combinations of the graph starting from a source node. Its operating principle is considered simple. After initializing the counters, the algorithm starts from a source, explores all the nodes joining it, and saves the distances for later comparison. Gradually, from one node to another, the shortest distance is determined and the closed node is removed from the graph dictionary. The shortest route to take is then the smallest sum of the arcs that can be taken from the source to the final destination.

2.4.4. Representation of the Resulting Route

To make the results obtained viable, it is advisable to represent the route adopted as optimal on a map. This modeling enables collectors to find their way around the site and makes it easier for them to move around.

This involves reading the same Excel file used to create the graph. The folium function first generates the world map, using the coordinates of one of the points on the graph to zoom in on the location of the study area. The identifiers of the nodes selected from the result of the compilation of the Dijkstra algorithm are recorded in a list. They are then used to retrieve the corresponding coordinates from the Excel file, enabling them to be stored in a given variable. This variable is the main input parameter for the "Folium.Polyline" function. Formatting characteristics can also be specified: the color, size, and opacity of the polyline forming the route. However, by simply applying the Dijkstra associated with the

In the "Folium.Polyline" model, the optimal route found follows a "flight of birds", i.e. a direct link between the various nodes making up the optimal path. Whereas the objective of this work is to find the optimum taking into account the accessibility of the routes taken.

The algorithmic program has therefore been reconsidered and new functions have been added. These include the ability to import all the communication routes available in the study area. This is done using the "Folium.map" class, which generates a map from the Leaflet library. The optimally defined route then selects the roads, streets, and tracks to connect the various

collection points. This process, which includes these and other features, is commonly referred to as solving the vehicle routing problem. The steps by which this process is executed and brought to completion will be detailed in the third paragraph of the following chapter.

Google Colab certainly allows us to run the PYTHON scripts and visualize the results, as shown in Figures 11 and 14 in the chapter on results and discussions, but as a deliverable, the project is not suitable. The user should not have direct access to the pre-established algorithmic codes to obtain the expected results, as these could be modified during the manipulation. To this end, an interface must be put in place to ensure interaction between the user and the program being developed.

2.4.5. Setting up a Graphical Interface

For this project, we opted for an application based on web development. Although there are many different ways of implementing a graphical interface, this alternative was adopted because of the flexibility offered by web development for generating the types of data we have as outputs. The resulting maps can be viewed via an HTML page. To do this, we used Django as the Framework, an open-source web development framework based on the Python language.

2.4.6. Creating a Django Project

Creating a project in Django is the first step in setting up a web application. It is done at the command prompt, commonly known as cmd or console. A specification of the access path where we want to list the project is optional but important for organizing files properly; this is created using the command "Django-admin start project" followed by the name of the project.

The application folder thus created contains Python files. The settings.py file manages all the application's basic configurations; it is defined by default and can be modified according to the project's requirements. The urls.py file handles all the URL routing operations that the application needs to run. To launch the project locally, the manage.py file is called at runtime in the command prompt. A Hypertext Transfer Protocol (HTTP) link is then generated to start the application in a browser.

2.4.7. Integration of Optimal Path Search Programs

The main purpose of this graphical interface is to run scripts to solve the problem of finding the shortest path. At the start of the website, the user has the option of choosing the Excel file containing the collection points. After choosing and saving the file using the **Choose File** and **Upload File** buttons respectively, the **Import Gap Points** button is used to import all the collection points and display them on the OpenStreetMap map. All that remains at this stage is to specify certain input parameters and draw the route in question using the **Draw Optimal Itinerary** button. The figure below provides an overview of this final stage.

This procedure is carried out using the "views.py" module in the application. The programs relating to each operation are entered in this module and the path by which the result is to be displayed is clearly defined in the URL files as specified above. However, care must be taken to ensure that the names of the call variables are identical in

the HTML and the **views.py** and **urls.py** files. For each view function, you also need to specify the rendering and the web page that will support viewing it. In fact, for greater organization and easier handling of the whole program, the views.py file has been split into several functions, each of which is generally associated with a button for executing a specific operation and therefore resolving a given stage in the overall process. By way of example, we will present the practical approach to creating the **file download** button.

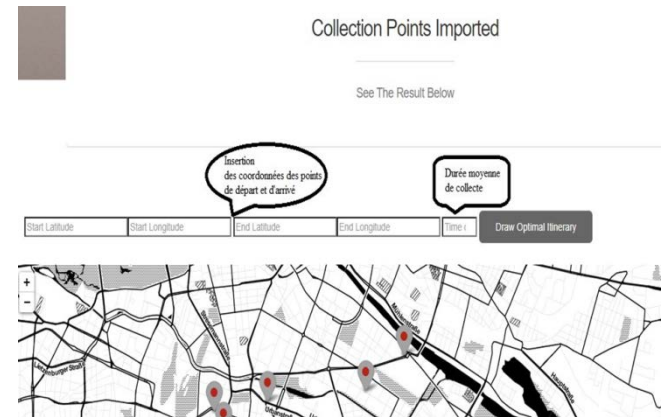


Figure 7. Route design process

2.4.8. Example of a Button Design

In the HTML page, the style of the button to be created must be defined: its location, size, color, and shape. Buttons that perform a specific action, in particular the execution of a script, are inserted in the "form" tag, where their name is specified along with the name of the function to be called using the "action" attribute. The action attribute indicates where to send the form data when it is submitted. For this example, the POST is used. Once the file has been selected, bearing in mind that .xlsx and .xls extensions are accepted, it is saved in a directory in the application. This file can then be called up when the user imports points.

To this end, we have also added a feature to meet the same requirements if the user wants to change the file. Using the "**os.path.getmtime**" command, the last file to be entered will always be considered for the next operation, which is to import the collection points. However, the disadvantage of this approach is that all the files set up by the user are stored in a folder. This results in a significant loss of memory. So, we've added a script that deletes the entire contents of the folder when a new file is uploaded.

It should be noted that for any action defined for the resolution of a given step, the result is assigned to one of the HTML pages of the application site for display. When a map is rendered, it is saved and a new file (.html) is created. This file contains all the scripts in JavaScript format, as well as the style links for displaying the map.

3. Results and Discussions

3.1. Gap Analysis

Gap analysis is used in a modeling framework aimed at predicting the distribution of interesting cultivated plant

breeds and identifying gaps in existing collections, particularly ex-situ. This approach is essentially based on the study of not only geographical but also environmental variations using specific factors, particularly accessibility and connectivity, known as scores. We have deduced from the literature and the opinion of professionals that the presence of gaps does not necessarily mean a total absence of genetic material in the location in question. Thus, during collection missions, a visit that is judged to be unprofitable in terms of germplasm collection is hardly considered useless because of its contribution to complementing existing databases.

The results of the study based on the gap analysis consist mainly of a map showing the probability of presence (Figure 8 - Left) and a gap map (Figure 8 - Right).

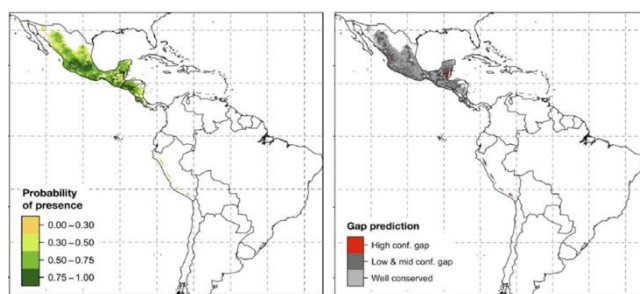


Figure 8. Example of Gap Analysis results (Ramirez-Villegas et al., 2020)

3.2. Gross Result Without Management Import

The optimal path as defined above refers to the choice of the shortest route ensuring passage over as many genetic material collection points as possible. To do this, the principle of the Dijkstra algorithm is applied while integrating the collection points as nodes in the graph. However, the simple application of Dijkstra does not provide a satisfactory return of the expected result. From source to destination, it selects a few collection points that constitute the shortest path and links them together using the *folium.Polyline* function, which does not take into account the accessibility of the routes taken. In other words, a route following the result is a "bird's eye view". The figure below illustrates the representation of the paths resulting from this approach. The need to reconsider the tracing of routes, this time following the usual communication routes, is obvious.



Figure 9. Representation of resulting routes following "bird's eye view"

3.3. Application of Dijkstra with import of communication channels

Integrating the available directions in the study area into the optimal path search process ensures that the various stakeholders have access to the collection points. There are several ways of importing communication routes. In the case of our study, the "OpenRouteService" platform was used. Using a secret code called "Api_Key" which we obtained beforehand, it allows us to import directions and apply the Dijkstra algorithmic principle to calculate the distance between points. It is also possible to introduce into the body of the algorithm the estimated calculation of the time, distance, and cost of the journey. This manipulation enables the time and financial savings to be evaluated, i.e. the added value of the optimization. Other platforms such as Google API can be used for the same purpose. But our choice was motivated by the accessibility of the ORS and the flexibility of the services provided. Other platforms such as Google

API can be used for the same purpose. However, our choice was motivated by the accessibility of the ORS and the flexibility of the services provided. Obtaining the "Api_Key" is also free.

However, ORS only considers land communication routes available on the World Street Map using the Leaflet library. Program execution is therefore limited when the characteristic points are far from the roads, streets, lanes, and tracks recognized in Leaflet.

This restriction can be circumvented by pre-processing the data in the input file. All you need to do is set up a Buffer zone so that only accesses within a 350 m radius of the directions allowed on the platform can be retrieved. Communication routes (roads, streets, tracks, alleyways, etc.) can be downloaded from www.Openstreetmap.org. They are then converted into a shapefile format (SHP). This conversion is carried out using Spatial Manager Desktop software, following a very specific procedure. After exporting a portion of the map to OpenStreetMap, the file extension obtained is "osm", a format that is not compatible with GIS platforms, which is why it cannot be read by ArcGIS. The result of this process is a vector file of all the roads in the study area in question, to which the Buffer operation is applied.

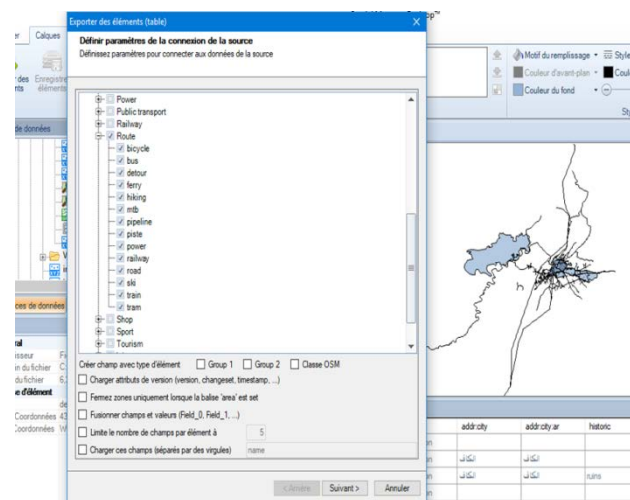


Figure 10. Converting and exporting a route layer in shp format to SMD

The Vehicle Routing Problem (VRP) is a combinatorial optimization and programming problem that asks the following question: "What is the optimal set of routes that a fleet of vehicles must take to deliver a given set of customers? In the context of this project, it is not a question of delivering customers but of exploring an environment made up of a set of gap points. Solving this problem is carried out using the ORS as described above. The following link: <https://colab.research.google.com/drive/15dLV1OgM6QOOQakS9EQJTSmweTA28cd#scrollTo=EDt1UP2Xzobl&line=4&uniqifier=1> shows the cell in the script specifically programmed for this resolution. The figure below shows the result obtained after drawing up the optimal route, taking into account the accessibility of the sites visited by the collectors.



Figure 11. Route representation with VRP resolution process

3.4. Integrating the Model Into a Web Application

To make the process of finding the optimal path more viable, it was decided that it would be more appropriate to integrate the model into a platform. This will allow users to manipulate the model more easily while retaining some degree of flexibility. The scripts designed for this purpose will be protected so that they can withstand errors caused by incorrect handling. With this in mind, we have set up a web application in the form of a graphical interface. The figure below shows the home window of the web application.

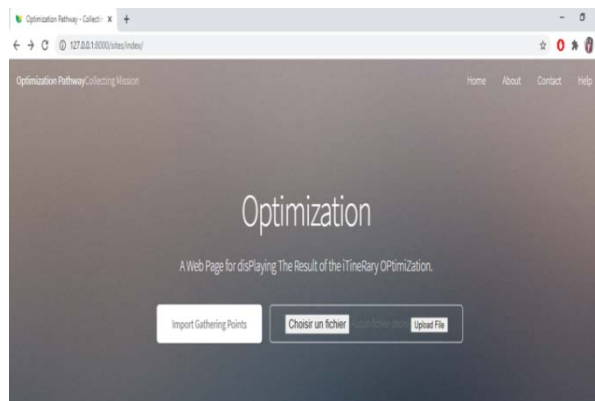


Figure 12. Web application home

To achieve the desired results, a certain procedure must be followed, as defined below.

- (i) Select the Excel file of collection points before saving it

- (ii) Importing these points and displaying them on the map
- (iii) Specification of start, finish and intermediate points
- (iv) Definition of average collection time (Figure 13)
- (v) Plotting the optimum route (Figure 14)

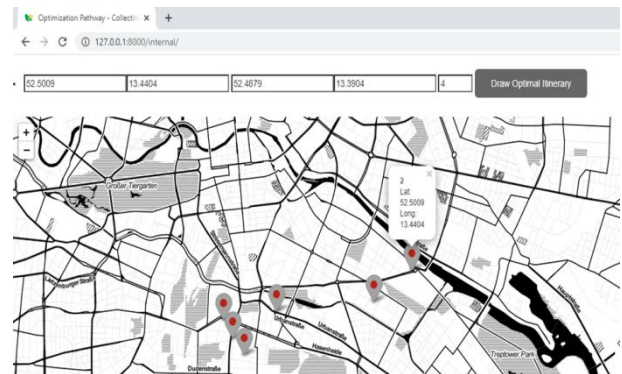


Figure 13. Inserting input parameters and plotting the route

To see the characteristics of the journey (Distance, Non-Stop Time, Total Time and Cost), drag the cursor over the route. The figure below gives an overview of the result obtained.

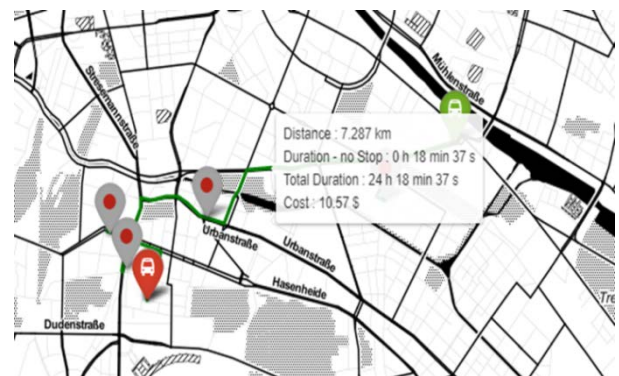


Figure 14. Optimal Route with route characteristics

The Non-Stop duration represents the duration of the journey, taking into account the negligible number of stops at collection points or gaps. The total duration refers to the duration of the journey, taking into account the average duration of each stop that the collection project planner specifies in advance (Figure 14).

To make it easier to use the application, we have set up a help platform and a facility for interaction via email conversations. A help window has been set up to explain how the application works, listing the various stages in the process. The figure below shows this window.

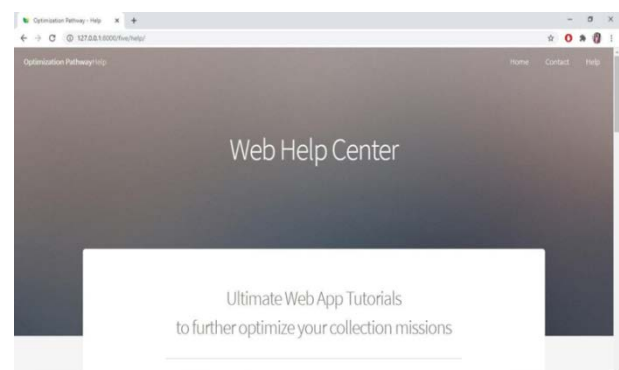


Figure 15. Help window - Operating tutorials

3.5. Contribution of the Application

Although the results obtained are satisfactory, the stage of evaluating the model's contribution is essential because it ensures a certain reliability. In the context of our project, this process consists of repeating a previous data collection mission to Tajikistan in the Central Asian region, using the approach we have proposed. The aim is to compare the various parameters that characterize the routes chosen in the two cases (with and without a model): cost, distance and duration. The decision-making tool provided by this application gives the decision-maker the opportunity to master the various choices to be made in order to optimize his mission.

It should be noted that the cost of the journey is estimated by taking into account a number of determining factors. The fundamental parameter is the distance of the journey, which is directly correlated with fuel consumption, the average price of which is within the reach of the general public. Then there's the topography of the region, the traffic and the types of roads used within it. All these parameters are directly or indirectly linked to each other.

ACKNOWLEDGEMENTS

This project is part of the ICARDA research program led by Mr KEHEL Zakaria. Carried out in the context of a diploma thesis by Mr. FALL El Hadji Malick Sy under the supervision of Ms. HAJJI Rafika, professor at IAV

Hassan II. I would like to thank all those mentioned, as well as Professor TINE Moustapha Gning, who corrected this paper, and Ms DIENG Sokhna Assietou Fall.

References

- [1] A. Jarvis, K. Williams, D. Williams, L. Guarino, P. J. Caballero, et G. Mottram, «Use of GIS for Optimizing a Collecting Mission for a Rare Wild Pepper (*Capsicum flexuosum* Sendtn.) in Paraguay», *Genet Resour Crop Evol*, vol. 52, n° 6, p. 671-682, sept. 2005.
- [2] I. E. Putra et K. Rohendi, «Implementation of Geographic Information System with Dijkstra Algorithm Base On Mobile Application: A Model for Disaster Risk Evacuation Route in Padang City Indonesia», in *Proceedings of the 1st International Conference on E-commerce, E-Business and E-Government*, in ICEEG '17. New York, NY, USA: Association for Computing Machinery, juin 2017, p. 30-34.
- [3] M. D. Jennings, «Gap analysis: concepts, methods, and recent results*», *Landscape Ecology*, vol. 15, n° 1, p. 5-20, janv. 2000.
- [4] J. Ramirez-Villegas *et al.*, «A gap analysis modelling framework to prioritize collecting for ex situ conservation of crop landraces », *Diversity and Distributions*, vol. 26, n° 6, p. 730-742, 2020.
- [5] Elith, «A statistical explanation of MaxEnt for ecologists Diversity and Distributions - Wiley Online Library». Consulté le: 8 décembre 2024. [En ligne].
- [6] E. W. Dijkstra, «A note on two problems in connexion with graphs», *Numer. Math.*, vol. 1, n° 1, p. 269-271, déc. 1959.
- [7] M. M. Hizem, *Path search in a dynamically weighted graph: Application to route optimization in road networks*, École Centrale de Lille. in PhD thesis. 2011.
- [8] A. Iglesias, «Calcul d'itinéraire multicritère en transport multimodal», phdthesis, Université de Lyon, 2017. Consulté le: 10 décembre 2024. [En ligne]. Disponible sur: <https://theses.hal.science/tel-01848737>.



© The Author(s) 2025. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).