

IPC04-0482

SAAAP – AN AUTOMATED PIPELINE ROUTING SYSTEM

Wilson José de Oliveira
PETROBRAS / ENGENHARIA / IEGEN / EGE / EAMB
Rua General Canabarro, 500
Rio de Janeiro, RJ - Brasil - CEP 20271

Patrícia Pereira Porciano

Beatriz V. Alvarez
Grupo de Tecnologia em Computação Gráfica / TeCGraf
Departamento de Informática - Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225
Rio de Janeiro, RJ - Brasil - CEP 22453-900

Marco A. Casanova

Marcelo Tílio M. de Carvalho

Anselmo Antunes Montenegro
IMPA – Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110
Rio de Janeiro, RJ - Brasil - CEP 22460-320

Paulo Cezar Pinto Carvalho

ABSTRACT

SAAAP – Routing Alternatives Environmental Evaluation System – was designed to select the best alternative route, within an area of interest, to implement a new pipeline project. The system takes into account economic, environmental and engineering factors, according to an optimality criterion that combines several variables, such as vegetation coverage, soil type and declivity. The system is operational and has been tested in several realistic projects.

1. INTRODUCTION

SAAAP – Routing Alternatives Environmental Evaluation System – was designed to help an interdisciplinary team of experts select the best alternative route, within an area of interest and connecting user defined points, to implement a new pipeline project. The team of experts usually covers several disciplines – Forest Management, Geology, Biology and Engineering.

The system is based on an optimization procedure that takes into account thematic data that represent economic and environmental aspects, such as vegetation coverage and soil type, as well as terrain slope and elevation. The experts are responsible for defining how to weight thematic data in such a way as to reflect their relative importance for the area of interest.

The user may invoke SAAAP to:

- compute the cost of a pipeline route he manually defined;

- automatically generate the best route connecting points he defined, based on a cost function that takes into account thematic data and terrain slope and elevation data, if available;
- generate the elevation profile of a given route, if the project includes elevation data.

The system features:

- storage of geographic data through a freeware library, TerraLib [CÂMARA];
- an optimized routing algorithm;
- a cost functions that reflect the experience of EAMB (PETROBRAS Environmental Engineering Department);
- additional routing heuristics designed by EAMB;
- generation of the combined 2D map and elevation profile of a route;
- 3D visualization of a route.

To compute the best route, SAAAP implements an algorithm similar to classical graph traversal techniques [GIBBONS]. The problem is also similar to determining the optimal route in a mesh (see, for example [SHEKHAR]). An approach based on genetic algorithms would also be a viable alternative [KIM].

This paper is organized as follows. Section 2 presents the data organization adopted. Section 3 briefly describes how to use the system. Section 4 discusses procedures to generate an optimal route. Finally, Section 5 contains the conclusions.

2. ORGANIZATION OF THE DATA

A *vector layer* is a set of geo-referenced objects whose geometric representations are vectorial (i.e, a set of points, lines, polygons, etc.) and use the same geo-referencing scheme.

A *grid layer* is a geo-referenced grid of rectangular cells, in a given geo-referencing scheme. Each cell has a value and represents a rectangle, in the layer's geo-referencing scheme.

A *thematic layer* is a grid layer whose cell values are classified into distinct *classes*, each class having a user-defined *weight*. Figure 1 schematically shows a thematic grid layer.

A *slope layer*, an *elevation layer* and an *image layer* are grid layers whose cell values respectively capture the slope, elevation and reflectance of the rectangle of the Earth surface the cell represents.

A *project* is a set of *homogeneous* layers, that is, a collection of layers that cover the exact same area, have the same geo-referencing scheme and the same cell geometry, for the grid layers. The *project grid* is a grid of cells with the same geometry as the grid layers in the project.

It will become clear in Section 4 why we favor grid layers.

A *pointwise cost function* for a project \mathcal{P} is any function $COST$ that maps each cell c of the project grid into $COST(c)$, called the *cost* of c , and defined as the weighted sum of the values of the cells of the thematic layers, the slope layer and the elevation layer in \mathcal{P} which correspond to c (the weight of a cell in a thematic layer is the product of the weight of its class by the weight assigned to the layer it belongs to).

Pointwise cost functions are so-called because the cost of a cell does not depend on the cost of the neighboring cells. Naturally, non-pointwise functions are possible, but they will be ignored in what follows. Hence, we refer to a pointwise

cost function simply as a cost function.

The *cost grid* for a project \mathcal{P} and a cost function $COST$ is a grid of rectangular cells such that:

- the grid has exactly the same geometry as the grid layers in \mathcal{P} ;
- the value of a cell c is $COST(c)$.

Figure 2 schematically illustrates how the cost grid is created.

Note that, in the current implementation, the vector layers are not used to create the cost grid, by definition. The reasons will become clear in the Section 4. The elevation layer is also used to generate the *elevation profile* of a route. An image layer is used only to provide a visual background.

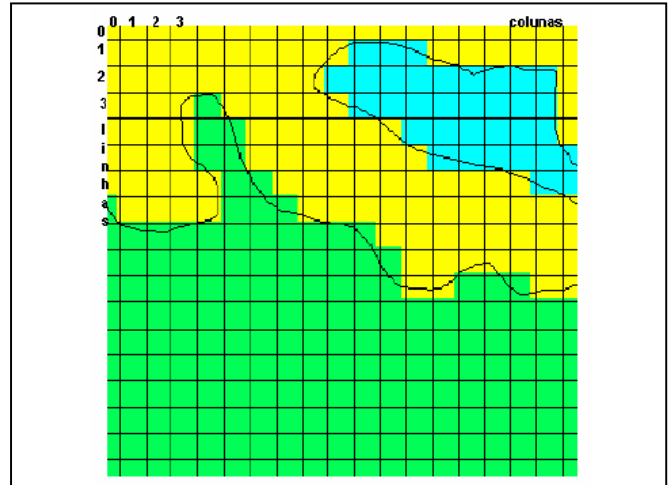


Figure 1 – Schematic representation of a thematic layer.

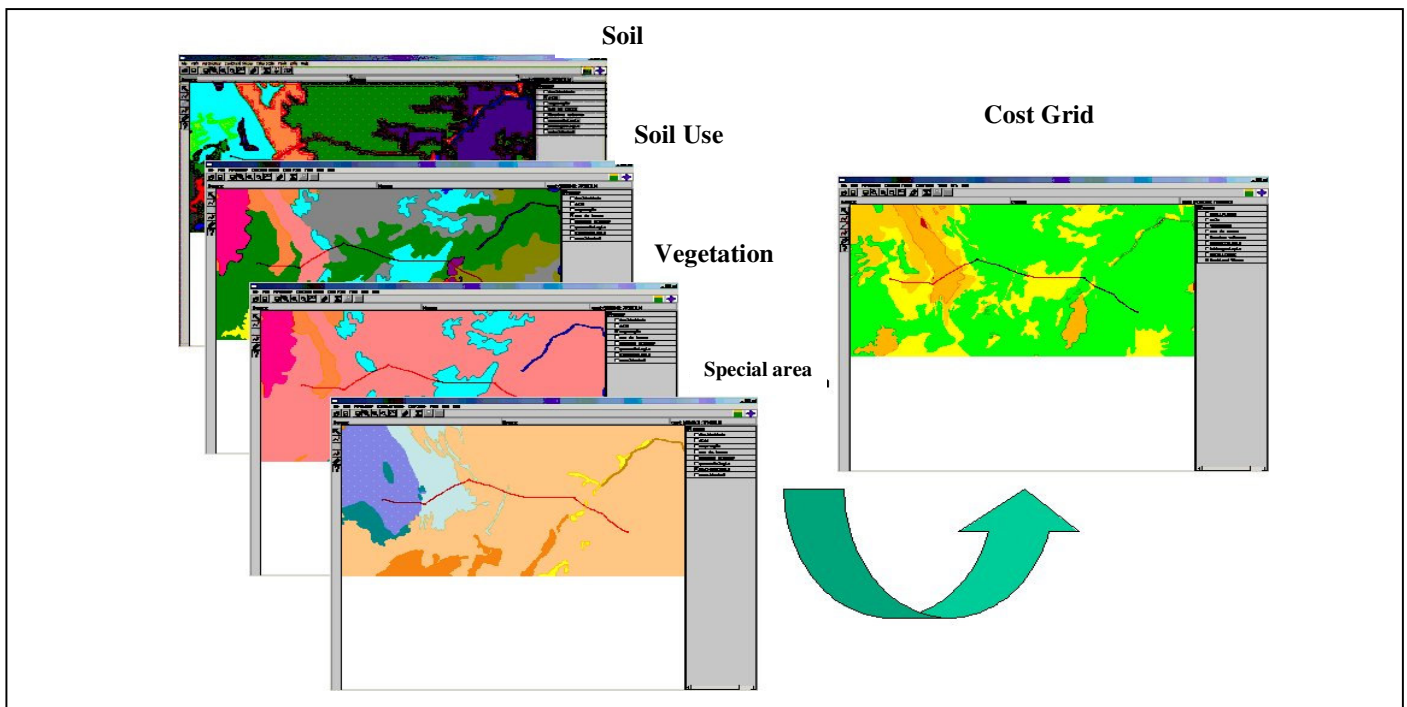


Figure 2 – Schematic definition of the grid cost.

3. DESCRIPTION OF THE SYSTEM

3.1 INTERFACE

The system's user interface is organized as follows (see Figure 3): a menu bar exposing the system's functions; an icons bar with shortcuts to some of the functions; a message area; a canvas toolbar; a panel listing the layers; a visualization canvas.

The major functions the interface exposes are:

- Manual route generation: permits defining a route by manually plotting points directly on the canvas.
- Automatic route generation: generates the best route connecting points manually plotted on the canvas.
- Generation of an elevation profile: generates the elevation profile for the selected route, provided that the project includes an elevation layer. The X axis corresponds to the distance traveled along the route and the Y axis, to the elevation of a point. By default, the graph covers the entire route selected. However, it is possible to generate the elevation profile just for a route segment. It is also possible to click on a point on the route to show its position of the elevation profile, and vice-versa.
- Route analysis: returns a set of vectors, each corresponding to a thematic layer and its classes. Let V be the vector associated with thematic layer L. Then, element V(i) indicates the percentage of the route that crosses cells belonging to the ith class of L.
- Point analysis: permits selecting a point on the canvas and asking to which class of each thematic layer the point belongs to.

The basic tools to manipulate the canvas are:

- Route manipulation: permits selecting, manually defining, manually editing and removing a route.
- Canvas tools: zoom in, zoom out, refresh, and fit.
- Distance measurement: permits measuring the distance between points on the canvas.

3.2 USING THE SYSTEM

The user should start by creating a new project or selecting an existing project. Recall from Section 2 that the project defines the area and the geo-referencing scheme, for all layers, and the cell geometry, for the grid layers.

The user then moves on to add layers to the project, either by importing new layers, or by selecting layers that already exist in the database. For a thematic layer, the user must also define the classes and their weight, and an overall weight for the layer.

Once this data preparation phase is concluded, the user may start experimenting with alternative routes, either by manually plotting them, or by invoking the automatic route generator. He can analyze each alternative route using the elevation profile tool, or the route analysis tool, as explained in Section 3.1.

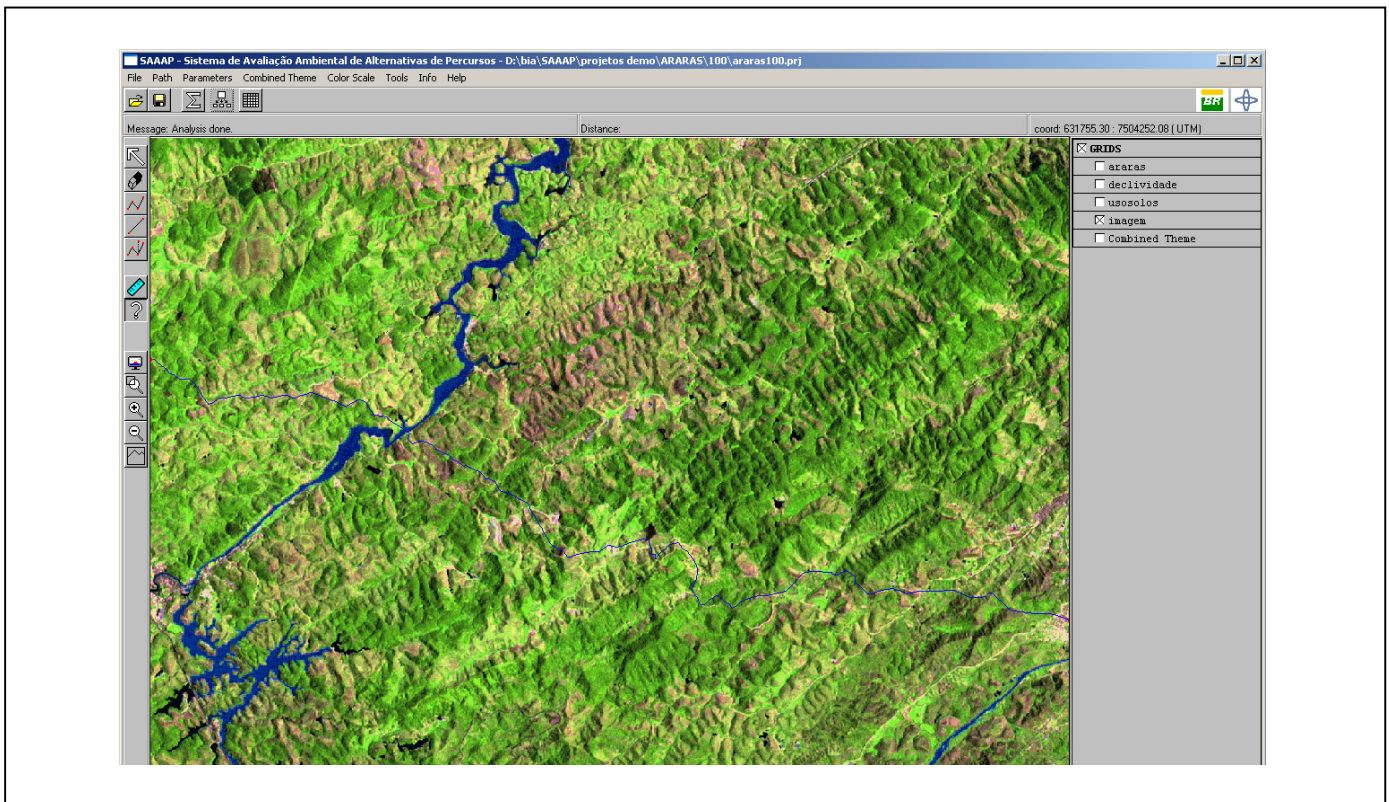


Figure 3 – SAAAP user interface.

4. GENERATING ALTERNATIVE ROUTES

We discuss in this section how to compute alternative routes over a project.

Let \mathcal{P} be a project and $COST$ be a (pointwise) cost function for \mathcal{P} . Let G be the cost grid for \mathcal{P} and $COST$.

A route r in G is any sequence of points $r = (P_1, \dots, P_n)$ such that P_i, P_{i+1} are centroids of neighboring cells in G , for $i \in [1, n-1]$. We denote by $COST(r)$ the sum of the cost of the cells that the route traverses.

The core of SAAAP is an optimization procedure that takes as input:

- a project \mathcal{P}
- a cost function $COST$ for \mathcal{P}
- a sequence of points P_1, \dots, P_n which are centroids of cells of the cost grid G

and produces as output:

- a route r that visits all points P_1, \dots, P_n , in this order (in particular, it starts on P_1 and ending on P_n), and that minimizes $COST(r)$

The simplest implementation is the *exhaustive search procedure* which, very briefly, generates all possible routes and returns the route r such that r traverses all input points in the correct order and r minimizes $COST(r)$.

However, the exhaustive search procedure is unfeasible when the cost grid G is reasonably large. We therefore discuss several improvements in what follows.

The *sub-sampling exhaustive search procedure* is basically exhaustive search, except that it does not visit every cell in the project grid G , but rather just a sub-sample of G , based on a spacing parameter s . Briefly, a coarser resolution grid G' is constructed by taking just the cells $c_{1+p.s, 1+q.s}$ of G , for $p \in [0, (m-1)/s]$ and $q \in [0, (n-1)/s]$, where $m \times n$ is the size of G . Then, exhaustive search is applied to G' and the route returned accepted as the best route.

The system currently implements a second variant, the *look-ahead exhaustive search procedure*, which combines sub-sampling and a look-ahead heuristics to extend the routes.

Given a grid G and a non-negative integer d , we say that two cells $c_{i,j}$ and $c_{p,q}$ in G are *d-neighbors* iff

$$|i - p| = d \text{ and } |j - q| \leq d \text{ or } |i - p| \leq d \text{ and } |j - q| = d.$$

that is, roughly speaking, iff $c_{i,j}$ and $c_{p,q}$ are d cells apart in each direction.

The look-ahead exhaustive search procedure has two input parameters – spacing s and radius d – and is very briefly described in pseudo-code as follows:

EXHAUSTIVE_L_AHEAD(\mathcal{P} , $COST$, (P_1, \dots, P_n) , s , p ; r)

1. Initialization:
 - a. Initialize a set of routes S with $\{(P_1)\}$.
 - b. Construct a cost grid, G' , by taking just the cells $c_{1+p.s, 1+q.s}$ of the original cost grid G of \mathcal{P} and $COST$, for $p \in [0, (m-1)/s]$ and $q \in [0, (n-1)/s]$, where $m \times n$ is the size of G .
 - c. Consider all cells of G' as unvisited.
2. While there is an unvisited cell c in G' do:
 - a. For each route $r = (R_1, \dots, R_k)$ in S such that R_k is the centroid of c do:

- i. For each unvisited cell c' that is a d -neighbor of c in G' , construct a new route $r' = (R_1, \dots, R_k, R_{k+1}, \dots, R_{k+q})$ such that R_{k+q} is the centroid of c' and r' is the best route from R_k to R_{k+q} that touches all input points in the d -neighboring area of c .
 - ii. Delete r from S and add to S all routes constructed in the previous step.
 - b. Consider c as visited.
3. Return the route r in S such that r traverses all input points in the correct order and minimizes $COST(r)$.

Note that this procedure reduces to the sub-sampling exhaustive search procedure when the radius r is equal to 1. If the radius is greater than 1, it indeed reduces the search space, when compared to sub-sampling, because it discards all routes from R_k to R_{k+q} , except the best one.

Other variants are naturally possible. The *bounded look-ahead exhaustive search procedure* assumes that the user manually limits look-ahead exhaustive search by defining a buffer area that covers all input points. The look-ahead exhaustive search procedure is then applied as before, but the search is limited to cells inside the buffer area.

The *multi-level look-ahead exhaustive search procedure* is a variant of bounded look-ahead exhaustive search, where the buffer area is automatically generated. Briefly, look-ahead exhaustive search is first applied at a coarse resolution grid, G_1 . The route r_1 returned is treated as a first approximation of the best route. Route r_1 is then projected to a finer resolution grid, G_2 , and a buffer area B_2 is defined around r_1 , based on a user-defined buffer width. Bounded look-ahead exhaustive search is then applied to G_2 , bounded by B_2 , to generate a refined route r_2 . This step can be repeated until the desired resolution is reached and the final route returned.

5. CONCLUSIONS

To conclude, it is worth mentioning that the SAAAP system has been in use for about 2 years now. It has been used at Petrobras, with realistic data, to help design new pipeline routes. The results indicate that the look-ahead exhaustive search procedure, even working with a coarse-grained grid, produces routes that are compatible with those manually designed by expert engineers. In fact, in many tests, the routes automatically obtained were almost identical to those manually designed.

Not surprisingly, performance is an issue. We therefore plan to experiment with the multi-level look-ahead exhaustive search procedure as a strategy to extend the system to very large project grids.

We also plan to expand the functionality of the system to include:

- advanced cost functions that take into account vector layers, such as hydrographic and road network maps (that is, the cost function would take into account the cost of crossing rivers and roads);
- 3D visualization of routes and simulation of 3D “flights” over the terrain surrounding a route land stripe;
- functions to generate advanced route elevation maps.

ACKNOWLEDGMENTS

The authors wish to thank Paula Frederick who implemented the final version of the system.

REFERENCES

CÂMARA, G. et. alli. "TerraLib: Technology in Support of GIS Innovation". Proc. II Brazilian Workshop on GeoInformatics, São Paulo, SP, June 2000.

GIBBON, A. Algorithmic Graph Theory. Cambridge University Press, Great Britain, 1985.

KIM, D. G.; Corne, D. W.; Ross, P. Industrial plant pipe-route optimisation with genetic algorithms. In Voigt et al. [273], pages 1012-1021.

SHEKHAR, S.; Fetterer, A. "Path Computation in Advanced Traveler Information Systems". In Proc. Intelligent Transportation Systems, 1996.
<http://citeseer.nj.nec.com/shekhar96path.html>