Ecole Centrale Paris

Laboratoire Génie Industriel

Cahier d'Études et de Recherche / Research Report

A Three-Phase Approach and a Fast Algorithm to Compute Efficient Corridors within GIS Framework

Vincent Mousseau, Hassène Aissi, Salem Chakhar

CER 10 – 07, juin 2010

A Three-Phase Approach and a Fast Algorithm to Compute Efficient Corridors within GIS Framework

Vincent Mousseau
Laboratoire de Génie Industriel Grande, École Centrale Paris,
Grande Voie des Vignes, 92295 Châtenay-Malabry Cedex, France.
Email: vincent.mousseau@ecp.fr
Hassène Aissi
LAMSADE, Univerisité Paris-Dauphine,
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France.
Email: aissi@lamsade.dauphine.fr
Salem Chakhar
Univerisité Paris-Dauphine,
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France.
Email: salem.chakhar@dauphine.fr

Abstract

The paper introduces a three-phase approach based on a coupling between a geographical information system (GIS) and multicriteria evaluation and devoted to handle bi-objective corridor siting problems. The corridors are evaluated on two criteria: (i) a quantitative criterion (such as length), and (ii) a qualitative criterion measuring the quality of a corridor with respect to the suitability of crossing its component polygons. To identify the efficient corridors, we designed a very fast algorithm that exploits the qualitative dimension (suitability) of each polygon in order to identify a set of efficient corridors with different suitability levels. The proposed approach and algorithm are validated through the development of a prototype and illustrated on a real-world instance application. The paper also discusses the behavior of the algorithm with large datasets.
Keywords—GIS, Corridor siting, Multicriteria evaluation, Spatial analysis, Multi-objective shortest path

1 INTRODUCTION

CORRIDOR siting is a frequent problem in many real-world applications concerning the identification of cost efficient routes for highways [2], roadways [3][10], railways [11], pipelines [34] and transmission lines [17]. Most of previous works use or extend traditional shortest-path or least-cost-path algorithms to solve these problems. Traditional algorithms have polynomial computational complexity and are intended to generate a single global optimal solution [34][36]. However, corridor siting problems often imply several conflicting evaluation criteria that it is difficult to find a single "optimal" solution. Accordingly, the use of a multi-objective approach seems to be more appropriate.

The criteria considered in multi-objective corridor siting problems can be divided into two categories representing two types of generally conflicting dimensions: (i) the first dimension captures the length (or cost) of a linear facility, and (ii) the second dimension accounts for the suitability of the corridor in terms of environmental impacts, quality of life, security and so on. The first dimension is structurally cardinal. The second dimension can hardly be "monetarized" and is often evaluated on a qualitative ordinal scale. Hence, rather than aggregating these two dimensions, it is preferable to proceed in a bi-objective analysis in which the decision maker can identify the best compromise solution(s) between quality criteria and cost/distance criteria.

In this paper, we propose an approach based on a coupling between a geographical information system [9] (GIS) and multicriteria evaluation [31]. Note first that in this paper we assume a vector data model where the study area is defined as a collection of polygons. The approach is composed of three consecutive phases: data transformation, qualitative assessment and corridor identification. The first phase is based on an intensive use of GIS to construct and then combine different criteria maps describing the study area. The second phase aims to evaluate the overall suitability of each polygon of the study area. The third phase makes use of a graph, called connectivity graph, in which vertices correspond to polygons and edges to adjacent polygons in order to identify corridors that compromise at best the cost and suitability dimensions.

To identify the efficient corridors, we designed a very fast algorithm that exploits the qualitative dimension (suitability) of each polygon in order to identify a set of efficient corridors with different suitability levels. The proposed resolution algorithm uses the conventional Dijkstra [16]'s algorithm to solve a number of mono-objective shortest path problems only—instead of resolving one bi-objective shortest path problem. This idea permits a substantial time saving in comparison to the idea of using a multi-objective shortest path algorithm such as the revised version of Martins [27]'s algorithm proposed by [20] to handle multi-objective shortest path problems.

The proposed approach and algorithm are validated through the development of a prototype and illustrated on a real-world instance application. The paper also discusses the behavior of the algorithm with large datasets. The results show the high performance of the proposed algorithm in comparison to other ones.

The paper is organized as follows. Section 2 presents the proposed corridor siting approach. Section 3 details efficient corridors computing algorithm. Section 4 addresses some implementation issues. Section 5 presents a step-by-step illustrative application. Section 6 studies the computational behavior of efficient corridors computing algorithm. Section 7 concludes the paper.

2 CORRIDOR SITING APPROACH

The proposed corridor siting approach is composed of three consecutive phases: data transformation, qualitative assessment and identification of corridors. The rest of this section presents each of these phases.

2.1 Phase 1. Data transformation

To evaluate the suitability of the study area, we need to assess in a qualitative way each polygon of the study area. Assessment can be grounded on several qualitative criteria. Each of these criteria is represented in terms of a map. The objective of this first phase is to construct and then combine different criteria maps describing the study area. Naturally, this phase requires the use of a GIS, which is, as stated by different authors [7][12][25], particularly useful for the construction of criteria maps.

2.1.1 Construction of criteria maps

In multicriteria evaluation [31], criteria are factors on which alternative solutions are evaluated and compared. The evaluation of an alternative a in respect to criterion g_j is denoted $g_j(a)$. In spatial multicriteria decision making, attributes are often associated with geographical entities and relationships between entities and therefore can be represented in the form of maps [7][25]. Within this context, a *criterion map* is a collection of polygons, each of which is characterized by its value on the corresponding criterion. Let $\mathcal{E}_j$ be the evaluation scale for criterion g_j. A criterion map, denoted $\mathbf{c}_j$, is the set $\{(s, g_j(s)) : s \in U_j\}$ where U_j is a set of polygons and g_j is a criterion function defined as follows:

$$\begin{array}{cccc} g_j & : & U_j & \rightarrow & \mathcal{E}_j \\ & & s & \rightarrow & g_j(s) \end{array} \qquad (1)$$

2.1.2 Combination of criteria maps

Let **G**={$\mathbf{c}_1$, $\mathbf{c}_2$,$\cdots$ $\mathbf{c}_m$} be a set of m criteria maps defined on the study area. The second step of data transformation phase is to overlay the different criteria maps in **G**. The output is a new *combined map*. Each polygon of the combined map **c** corresponds to the intersection of the criteria maps $\mathbf{c}_1$, $\mathbf{c}_2$,$\cdots$,$\mathbf{c}_m$ as follows (see Figure 1):

$$\mathbf{c} = \{u^1_{i_1} \cap u^2_{i_2} \cap \cdots \cap u^m_{i_m} \neq \emptyset, i_j \in \{1, 2, \cdots, n_j\}, j \in \{1, 2, \cdots, m\}\} \quad (2)$$

where n_j denotes the number of polygons in the criterion map $\mathbf{c}_j$, and i_j is an index varying in the range $\{1, 2, \cdots, n_j\}$. Hence, u^j_k is the k^{th} polygon of criterion map $\mathbf{c}_j$. The non-empty intersection involved in (2) corresponds to the overlay of the polygons such that the intersection of their interior is non-empty. Each polygon u in the combined map is associated with the vector $\mathbf{g}(u) = (g_1(u), \cdots, g_m(u))$ which represents the evaluations of u with respect to the evaluation criteria $g_1, g_2, \cdots, g_m$ associated with the criterion maps in **G**. In the following we will denote by U the set of polygons of the combined map.

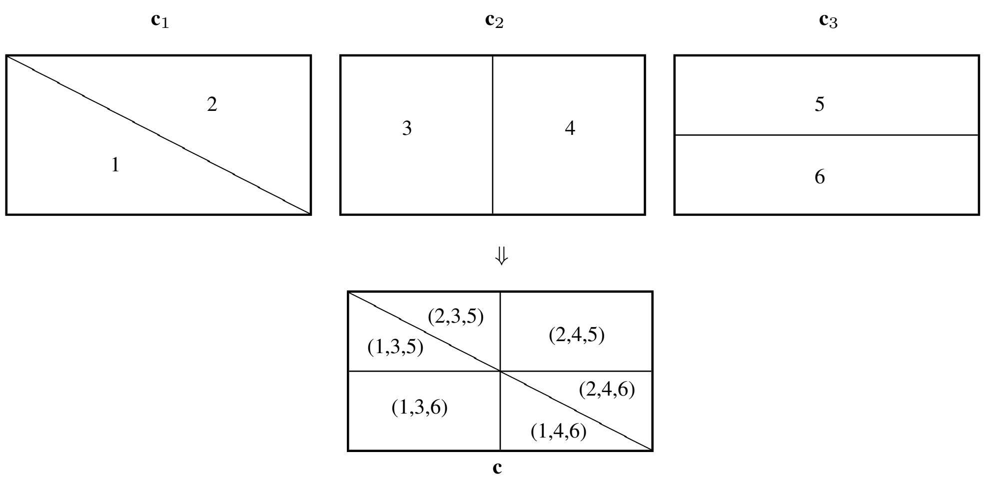

Fig. 1. Combination of criteria maps

Algorithm 1 below formalizes the combined map construction process. The procedure **Transform-ToPolygonal** permits to transform non-polygonal criteria maps into polygonal ones. The transformation of criteria maps is required because input datasets may be sample data points, raster maps, contour lines, digital terrain models, etc., while the proposed approach assume a polygonal representation of landscape. The **Union** procedure implements the union version of the GIS overlay operation and produces an initial version of the combined map. Procedure **RemoveSilverPolygons** permits to eliminate silver polygons from the combined map. Instructions in lines 8-15 permit to set the partial (i.e. in respect to a single criterion) performances of the polygons in the combined map. The intersection in line 11 permits to identify the polygon in criterion map $\mathbf{c}_j$ from which polygon u is issued.

Algorithm 1 runs in $\theta(m+m\cdot\sum_{j=1}^{m} n_j)$ where m is the number of criteria and n_j is the number of polygons in criterion map $\mathbf{c}_j$. The complexity of **Union**, **TransformToPolygonal** and **RemoveSilverPolygons** are not included in the complexity of Algorithm 1 since they correspond to built-in GIS functions.

Algorithm 1 **CombinedMapConstruction**
Input: $\mathbf{c}_1,\cdots,\mathbf{c}_m$: Criteria maps
Output: $\mathbf{c}$: Combined map
for $j = 1$ to m **do**
    if $\mathbf{c}_j$ is not polygonal **then**
        $\mathbf{c}_j \leftarrow$ **TransformToPolygonal**($\mathbf{c}_j$)
    end if
end for
$\mathbf{c} \leftarrow$ **Union**($\mathbf{c}_1,\cdots,\mathbf{c}_m$)
$\mathbf{c} \leftarrow$ **RemoveSilverPolygons**($\mathbf{c}$)
for each u in $\mathbf{c}$ **do**
    $s \leftarrow \emptyset$
    for $j = 1$ to m **do**
        $s \leftarrow s_i \in \mathbf{c}_j$ such that $(u \cap s_i) \neq \emptyset$
        $g_j(u) \leftarrow g_j(s)$
    end for
    $\mathbf{g}(u) \leftarrow (g_1(u), \cdots, g_m(u))$
end for
$\mathbf{c} \leftarrow \{(u, \mathbf{g}(u)\}$
return $\mathbf{c}$

2.2 Phase 2. Qualitative assessment

The objective here is to evaluate the suitability for the corridor to cross each polygon of the study area. As stated earlier, the formal model used to specify the evaluation of polygons is grounded on a multicriteria classification method. The output of this phase is a map in which each polygon is assigned a suitability level on an ordinal scale $\mathcal{E}$ composed of a finite set of p evaluation levels: $\varepsilon_1 \prec \varepsilon_2 \prec \cdots \prec \varepsilon_p$. Let Γ denotes a multicriteria classification model. Then, to evaluate the global suitability of polygons, we need to apply the multicriteria classification model Γ to assign to each polygon $u \in U$ a level on $\mathcal{E}$:

$$\begin{array}{rcccl} \Gamma & : & U & \longrightarrow & \mathcal{E} \\ & & u & \longrightarrow & \Gamma(u) \end{array} \quad (3)$$

The map obtained in terms of this phase is called *decision map* [5][8] and denoted by **M**=$\langle U, \Gamma(U)\rangle$. Different multicriteria classification models or any kind of approach that produces homogeneous polygons could be applied. In this paper we used the multicriteria classification model ELECTRE TRI [19]. It assigns objects described by several criteria to ordered categories. ELECTRE TRI uses multidimensional profiles $b_1, b_2, \cdots, b_{p-1}$ to define the limits of p consecutive categories. In our approach the categories correspond to the levels of the evaluation scale $\mathcal{E}$. Figure 2 shows the definition of an ordinal scale with three levels.

ELECTTRE TRI has two assignment procedures: optimistic and pessimistic; see [19][29] for details. The pessimistic assignment procedure of ELECTRE TRI, which is used in this paper, is given in Algorithm 2. Algorithm 2 compares each polygon u to each of the profiles limits staring from the highest one and assign u to the first category such that its lower profile limit verifies an *assignment rule*; see Section 4.7. The function **AssignmentTest** (given in Section 4.7) in Algorithm 2 permits to check if the assignment rule holds or not. The boolean variable `assigned` in Algorithm 2 is used to avoid unnecessary loops. Algorithm 2 runs in $\theta(r \times p)$, where r is the number of polygons in $\mathbf{c}$ and p is the number of levels.

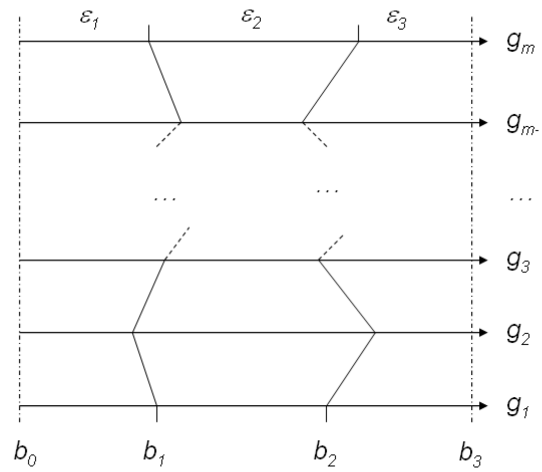

Fig. 2. The definition of an ordinal scale with three levels

Algorithm 2 **QualitativeAssessment**
Input: **c**: Combined map, $\mathcal{E}$: Suitability measurement scale
Output: **M**: Decision map

$p \leftarrow$ number of levels in $\mathcal{E}$
for each u in **c** **do**
  $h \leftarrow p$
  assigned $\leftarrow$ False
  while $h \geq 0$ and not(assigned) **do**
    if AssignmentTest(u, b_h) **then**
      $\Gamma(u) \leftarrow h + 1$
      assigned $\leftarrow$ True
    end if
    $h \leftarrow h - 1$
  end while
end for
M$\leftarrow \langle U, \Gamma(U) \rangle$
return **M**

2.3 Phase 3. Identification of corridors

The third phase explicitly concerns the identification of corridors. These corridors correspond to a collection of adjacent polygons specifying a "path" from origin s to destination t in the connectivity graph $G = (U, E)$ defined such that:

- $U = \{$polygons defining the study area$\}$ and
- $E = \{(u_i, u_j) \in U : u_i$ and u_j are adjacent polygons$\}$.

It is easy to see that the connectivity graph is planar by construction. Note also that a corridor corresponds to a chain from s to t in the connectivity graph. Furthermore, two types of evaluations are considered:

- The qualitative evaluation $\Gamma(u)$ associated with each vertex $u \in U$.
- The length $l(u_i, u_j)$ associated with each edge $(u_i, u_j) \in E$.

The length of an edge is defined as the Euclidian distance between the centroïds of adjacent polygons. The qualitative evaluations are obtained in terms of qualitative assessment phase. These evaluations are meant to be used in a MinMax criterion to evaluate corridors. Hence, corridors are evaluated on the second criterion by the maximum qualitative evaluation (worst evaluation) among the polygons composing the corridor. The next section presents the algorithm designed to identify efficient corridors.

3 COMPUTING EFFICIENT CORRIDORS

We now model the identification of corridors as a bi-objective shortest path problem. This requires to transform the connectivity graph in such a way that both evaluations $\Gamma(.)$ and $l(.,.)$ are associated with edges. For this purpose, we will use transformation schema shown in Figure 3. Hence, for each edge (u_i, u_j) we consider a vector of two evaluations $\mathbf{e}(u_i, u_j) = (l(u_i, u_j), \tau(u_i, u_j))$ where $l(u_i, u_j)$ corresponds to the original length of (u_i, u_j) and $\tau(u_i, u_j) = \max\{\Gamma(u_i), \Gamma(u_j)\}$. The reason for defining $\tau(u_i, u_j)$ in this way is related to the fact that $\Gamma(.)$ is considered as a MinMax criterion. In fact, any path which includes the edge (u_i, u_j) should take into account for both $\Gamma(u_i)$ and $\Gamma(u_j)$; hence the maximum.

The construction of the connectivity graph according to this transformation schema is formalized in Algorithm 3. Function **Neighbors** in Algorithm 3 returns all the neighbors of the polygon given as parameter and function **Distance** computes the Euclidian distance between the centroïds of the two polygons provided as parameters. The "`if···then···`" test in Algorithm 3 permits to eliminate redundancy by removing any edge (u, v) such that the edge (v, u) is already in E. This permits to reduce the size of the connectivity graph and therefore CPU time. Algorithm 3 runs in $\theta(n^2)$ where n is the cardinality of U. Note that the complexities of functions **Neighbors** and **Distance** are not included in the definition of the complexity of Algorithm 3 since they act directly on the database level and both of them are available as built-in functions in most of commercial GIS.

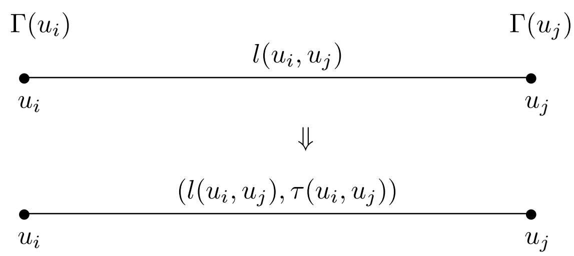

Fig. 3. Transformation schema

Algorithm 3 **ConnectivityGraphConstruction**
Input: **M**: Decision map
Output: G: Connectivity graph

```
E ← ∅
Z ← ∅
for each u in U do
  V ← Neighbors(u)
  for each v in V do
    if v ∉ Z then
      e_1 ← Distance(u, v)
      e_2 ← max(Γ(u), Γ(v))
      E ← E ∪ (u, v) with e(u, v) = (e_1, e_2)
    end if
  end for
  Z ← Z ∪ {u}
end for
G ← (U, E)
return G
```

To solve the resulting bi-objective shortest path problem, we can use a revised version of Martins [27]'s algorithm proposed by [20] to handle multi-objective shortest path problems. However, since each edge (u_i, u_j) is evaluated by the vector $\mathbf{e} = (l(u_i, u_j), \tau(u_i, u_j))$, the complexity of the computation of the efficient set can be strongly reduced by solving a number of mono-objective shortest path problems only.

Clearly, the number of efficient solutions is at most equal to $\min\{p, (n-1)l\}$, where p is the number of evaluation levels in the qualitative scale $\mathcal{E}$, $n = |U|$ and l denotes an upper bound of $l(u_i, u_j)$ for all $(u_i, u_j) \in E$. The efficient set is constructed by solving the following bottleneck shortest path problems: find if there is a shortest path $\mathcal{P}$ from s to t using distance $l(u_i, u_j)$ such that $\max_{(u_i,u_j)\in\mathcal{P}} \tau(u_i, u_j) \preceq v$, for $v = 1, \ldots, p$. To solve each of these problems, edges (u_i, u_j) such that $\tau(u_i, u_j) \succ v$ are deleted from the graph and the classical Dijkstra [16]'s algorithm is applied.

The resolution procedure discussed above is summed up in Algorithm 4. Function **Dijkstra** in this algorithm implements Dijkstra [16]'s algorithm and returns the shortest path (of level v) in graph $G_v = (U_v, E_v)$, which is provided as a parameter to this function. The complexity of the resolution algorithm is $pD(r, n)$, where $r = |E|$ and $D(r, n)$ is the complexity of computing a shortest path. Note that $D(r, n)$ strongly depends on the data structure used in the Dijkstra [16]'s algorithm (see Sections 4.3-4.5).

Algorithm 4 **EfficientCorridors**
Input: G: Connectivity graph, $\mathcal{E}$: Scale, O, D: Origin and destination nodes
Output: $\mathcal{S}$: Efficient corridors
01. $p \leftarrow$ number of levels in $\mathcal{E}$
02. $\mathcal{S} \leftarrow \emptyset$
03. **for** $v = 1$ to p **do**
04. $U_v \leftarrow U$
05. $E_v \leftarrow E \setminus \{(u_i, u_j) \in E : \tau(u_i, u_j) \succ v\}$
06. $G_v \leftarrow (U_v, E_v)$
07. $\mathcal{P}_v \leftarrow$ **Dijkstra**(G_v, O, D)
08. $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{P}_v$
09. **end for**
10. return $\mathcal{S}$

4 IMPLEMENTATION ISSUES

A prototype—called SAConstructor, for spatial alternatives constructor—has been developed to implement the proposed approach. Some implementation issues are discussed in the rest of this section.

4.1 Prototype

The general architecture of the prototype is given in Figure 4. The prototype was developed using ArcGIS® 9.2 Desktop [18] as the main software component and VBA as the development language for the graphical user interface. The prototype allows the construction of criteria maps, their combination, the qualitative assessment of polygons, and the computing of efficient corridors. The prototype is linked to an existing software called IRIS [13] used to infer the parameters required by the multicriteria classification method as explained later in Section 4.8.

Algorithms 1, 2 and 3—for combined map construction, qualitative assessment and connectivity graph construction, respectively—are coded in VBA. This also the case with Algorithm 5 (presented latter in Section 4.7) which implements the function **AssignmentTest** in Algorithm 2. In turn, Algorithm 4 for computing the efficient corridors is coded in C++. The main argument in favor of using C++ instead of the VBA language, which is supported by the used GIS, is the high competitivity of C++ compared to VBA—in terms of CUP time. The "Resolution Routine" component in Figure 4 is the executable version of the resolution algorithm. The GIS and this routine are *loosely* coupled (see [4][6][25]) and the dialogue between them is handled through ".txt" files.

4.2 Database

Maps are defined through ArcGIS shape files. Evaluation matrix and parameters required for ELECTRE TRI (e.g. profiles limits, list of criteria, weights of criteria maps, etc.) are defined separately as Microsoft Office Access database.

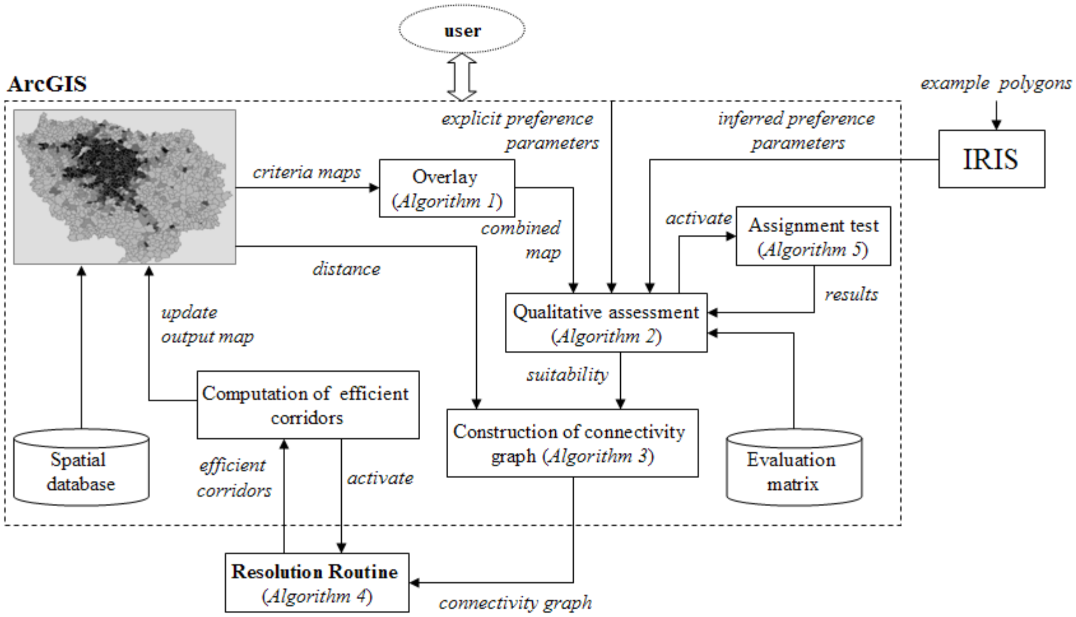

Fig. 4. Architecture of the prototype

4.3 Data structure

The connectivity graph is represented by an adjacency list. The node adjacency list $A(u_i)$ of node u_i is a linked list having $|A(u_i)|$ cells; each cell corresponds to an arc (u_i, u_j). One data field stores the node u_j, and two other fields store the distance $l(u_i, u_j)$ and the ordinal value $\tau(u_i, u_j)$. A d-ary heap is used to implement the Dijkstra [16] algorithm, where parameter $d \simeq \lceil \frac{r}{n} \rceil$; see [1] for more details.

4.4 Running time of Dijkstra algorithm

As stated above, the complexity of Dijkstra's algorithm depends on data structure adopted. For any implementation of set Q of all nodes in a graph $G = (U, E)$, the the running time of Dijkstra's algorithm is $O(|E| \cdot dk_Q + |U| \cdot em_Q)$, where dk_Q and em_Q are times needed to perform decrease key and extract minimum operations in set Q, respectively. The simplest implementation of the Dijkstra's algorithm stores vertices of set Q in an ordinary linked list or array, and extract minimum from Q is simply a linear search through all vertices in Q. In this case, the running time is $O(|U|^2 + |E|) = O(|U|^2)$. For sparse graphs, that is, graphs with far fewer than $O(|U|^2)$ edges, Dijkstra's algorithm can be implemented more efficiently by storing the graph in the form of adjacency lists and using a binary heap, pairing heap, or Fibonacci heap as a priority queue to implement extracting minimum efficiently. With a binary heap, the algorithm requires $O((|E| + |U|)\log |U|)$ time (which is dominated by $O(|E|\log |U|)$, assuming the graph is connected), and the Fibonacci heap improves this to $O(|E| + |U|\log |U|)$.

4.5 Running time of resolution algorithm

As established in the previous section and since we used a d-ary heap to implement the graph, the running time of Dijkstra's algorithm is $O(|E|\log |U|)$. Hence, the overall running time of the resolution algorithm is $O(p|E|\log |U|)$. This is straightforward since Dijkstra's algorithm is called p times in the resolution algorithm.

4.6 Multicriteria classification model

The multicriteria classification model used is ELECTRE TRI [19][29]. It assigns objects described by several criteria to ordered categories. ELECTRE TRI is based on the outranking relation [31], which is more general than the dominance relation and which is shown to be more appropriate in practice as advocated by different authors [24][25][26][32]. As explained earlier in Section 2.2, ELECTRE TRI uses multidimensional profiles $b_1, b_2, \cdots, b_{p-1}$ to define the limits of p consecutive categories, which in the proposed approach correspond to the levels of the evaluation scale $\mathcal{E}$. Further and in order to represent decision maker's preferences, ELECTRE TRI uses weights $w_1, w_2, \cdots, w_m$, indifference thresholds $q_1, q_2, \cdots, q_m$ preference thresholds $p_1, p_2, \cdots, p_m$ and veto thresholds $v_1, v_2, \cdots, v_m$, all of them are associated to criteria. Threshold $q_j(b_h)$ represents the largest difference $g_j(u) - g_j(b_h)$ preserving an indifference between u and b_h in respect to criterion g_j. Threshold $p_j(b_h)$ represents the smallest difference $g_j(u) - g_j(b_h)$ compatible with a preference in favor of u in respect to criterion g_j. Threshold $v_j(b_h)$ represents the smallest difference $g_j(b_h) - g_j(u)$ incompatible with the assignment of u to level h.

4.7 Assignment rule

Each multicriteria classification model is characterized by its *assignment rule*. As underlined earlier in Section 2.2, ELECTRE TRI has two assignment procedures, optimistic and pessimistic. The pessimistic assignment rule in ELECTRE TRI requires the computing of the *credibility indexes* $\sigma(u, b_h) \in [0, 1]$ measuring the level to which a given polygon u *outranks* (i.e. at least as good as) the profile b_h. As far as this present paper is concerned, it is sufficient to know that a polygon u is assigned to a level h only and only if $\sigma(u, b_h)$ is greater or equal to $\lambda \in [0.5, 1]$, where λ is a *cutting level* representing the lowest value for the credibility indexes $\sigma(u, b_h)$ to validate the outranking situation of u upon b_h. Mathematically, the assignment rule of ELECTRE TRI is defined as

$$\sigma(a, b_h) \geq \lambda \quad \Leftrightarrow \quad u \in C_{h+1}, \tag{4}$$

where C_{h+1} is the category delimited by profiles b_h and b_{h+1}, and corresponds to level ε_{h+1} on scale $\mathcal{E}$. This assignment rule is implemented by Algorithm 5 below, which also corresponds to function **AssignmentTest** in Algorithm 2 introduced in Section 2.2. Algorithm 5 runs in $O(m)$, where m is the number of evaluation criteria.

Algorithm 5 **AssignmentTest**
Input: u: Polygon, b_h: Profile, λ: Cutting level, F: List of criteria
Output: Boolean: *True* if sentence (4) holds; *False* otherwise
for each $j \in F$ **do**
  if $g_j(b_h) - g_j(u) \geq p_j(b_h)$ **then** $c_j \leftarrow 0$
  else if $g_j(b_h) - g_j(u) \leq q_j(b_h)$ then $c_j \leftarrow 1$
    else $c_j \leftarrow (p_j(b_h) - g_j(b_h) + g_j(u))/(p_j(b_h) - q_j(b_h))$
    end if
  end if
end for
$c \leftarrow \sum_{j \in F} w_j \cdot c_j$
for each $j \in F$ **do**
  if $g_j(u) \leq g_j(b_h) + p_j(b_h)$ **then** $d_j \leftarrow 0$
  else if $g_j(u) > g_j(b_h) + v_j(b_h)$ then $d_j \leftarrow 1$
    else $d_j \leftarrow (v_j(b_h) - g_j(u) + g_j(b_h))/(v_j(b_h) - p_j(b_h))$
    end if
  end if
end for
$F' \leftarrow \{j \in F' : d_j > c\}$
$d \leftarrow \prod_{j \in F'} (1 - d_j)/(1 - c)$
if $c \cdot d \geq \lambda$ **then** return true **else** return false **end if**

4.8 Inference procedure

As shown earlier, the application of ELECTRE TRI model requires the definition of several preference parameters. In practice, assigning precise values to these parameters requires an important cognitive effort from the decision maker. To reduce this effort, one possible solution is to proceed by an *aggregation/disaggregation* approach [15] as follows. First, the decision maker is required to provide the evaluation on the scale $\mathcal{E}$ of a limited number of polygons $u \in U^* \subseteq U$. An inference procedure is then used to infer values for the parameters b_h $(h = 1, \cdots, p-1)$, w_j, v_j, p_j and q_j $(j = 1, \cdots, m)$ which restore the assignments provided by the decision maker. The obtained values are then used to apply ELECTRE TRI. Note that inferring simultaneously all preference parameters is computationally difficult. For this purpose, different *partial inference procedures* have been designed; see [14][15][29][30]. In the illustrative example discussed in Section 5, we have applied a partial inference procedure to infer values for the weights. For more details concerning the aggregation/disaggregation approach, see [15][28].

5 ILLUSTRATIVE APPLICATION

In this section, we provide a step-by-step illustrative application of the proposed approach to a hypothetical corridor siting problem using real-world data relative to the Ile-de France region (Paris and its suburbs) in France.

5.1 Problem description

The aim of this illustrative application is to identify efficient corridors joining the counties (*communes*) Saclay and Roissy-en-France. Saclay is a county in the South-West of Paris where a important research/eduction pole will be located (this pole will be operational in a five years horizon). Roissy-en-France, located in the North-East of Paris, is one of counties where the Charles-de-Gaulle airport is located. In terms of distance, the shortest path joining these two counties leads to cross Paris. To avoid crossing zones with a high population density, it is necessary to by-pass Paris and therefore to accept a longer linear facility.

5.2 Dataset

The data used are essentially of socio-economic nature relative to the counties of the study area. We have used this data set in a previous application [5] to solve a hypothetical corridor siting problem. In this previous research, we have applied an *ad hoc* aggregation rule to evaluate corridors. Note that the application described in this paper and the one in [5] have different origin and destination nodes. One important shortcoming of the application in [5] is the relatively high CPU time in comparison to the resolution algorithm proposed in this paper.

5.3 Data transformation

5.3.1 Construction of criteria maps

The first step of data transformation phase is to construct the criteria maps. In our case, three criteria maps are elaborated (`DemographicDensity`, `EmploymentLevel`, and `Surface`) based on the following three definitions:

$$g_1(u_i) = \frac{\texttt{Total\_Population}(u_i)}{\texttt{Total\_Surface}(u_i)} \quad (5)$$

$$g_2(u_i) = \frac{\texttt{Active\_Population}(u_i)}{\texttt{Total\_Population}(u_i)} \quad (6)$$

$$g_3(u_i) = \frac{\texttt{Available\_Land}(u_i)}{\texttt{Total\_Surface}(u_i)} \quad (7)$$

First, to avoid nuisances, the corridor should avoid highly populated counties. Second, as the corridor refers to a transportation facility, it should cross counties with high employment level. Third, the corridor should cross counties in which the proportion of available land is high. Therefore, the first criterion should be minimized, and the two others are to be maximized. The criterion map given in Figure 5 represents the demographic density for the study area.

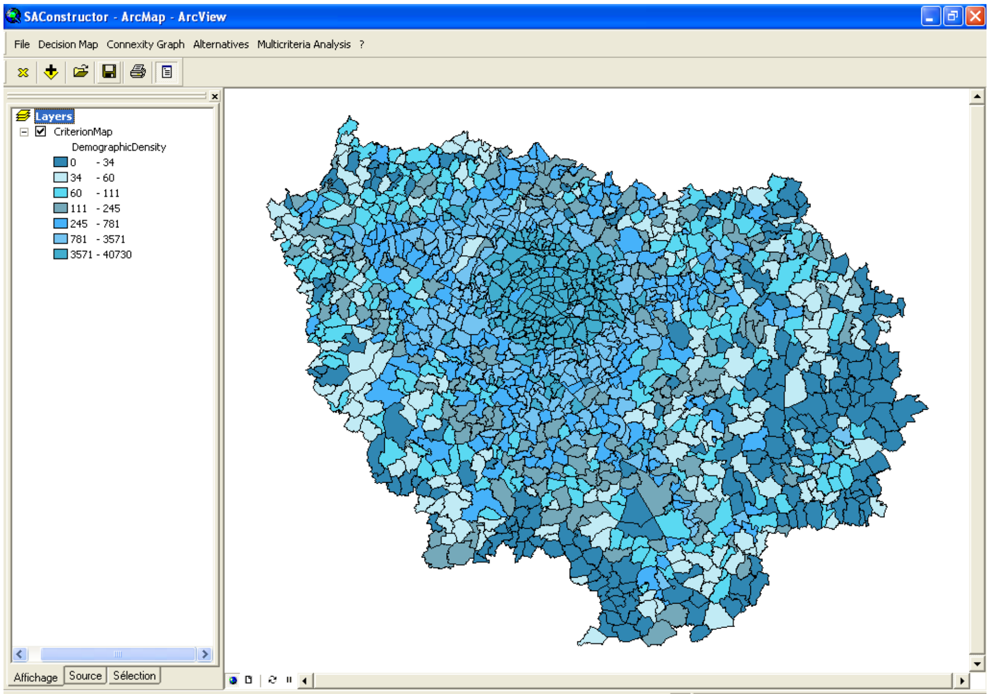

Fig. 5. Criterion map "`DemographicDensity`"

5.3.2 Combination of criteria maps

The next step in data transformation phase consists in combing the three criteria maps; these amounts at proceeding the overlay operation to combine these maps. It is important to note that the criteria maps in this example have all the same topology. Thus, the application of the GIS union operation implies only the descriptive attributes of these maps.

5.4 Qualitative assessment

The second phase aims at classifying each polygon of the study region into one of seven suitability levels. As mentioned earlier, the classification model used in this paper is ELECTRE TRI. To apply ELECTRE TRI, it is necessary to specify the values of the preference parameters: category limits b_h ($h = 1, \cdots, 6$), veto thresholds (v_1, v_2, v_3) and weights (w_1, w_2, w_3). For the purpose of this application, indifference and preference thresholds are fixed to zero. The profiles limits and the associated veto thresholds are given in Table 1. The values for these parameters are fixed by the authors after several simulations.

We have used the aggregation/disaggregation approach of [15] to specify the values for the weights. For this purpose, we require from the decision maker typical polygons that s/he assign to the seven suitability levels. From these typical polygons, values for the weights have been inferred through a partial inference procedure. For this purpose, the IRIS system (see Section 4.1) supports the inference procedure. The typical polygons are given in Table 2, which includes also the partial evaluations of the different typical polygons. The columns "Category min" and "Category max" correspond to the lowest and highest categories to which polygon u_i should be assigned, respectively. Inferred values are given in Table 3.

TABLE 1
Preference parameters

g_j	$g_j(b_1)$	$v_j(b_1)$	$g_j(b_2)$	$v_j(b_2)$	$g_j(b_3)$	$v_j(b_3)$	$g_j(b_4)$	$v_j(b_4)$	$g_j(b_5)$	$v_j(b_5)$	$g_(b_6)$	$v_j(b_6)$
g_1 (-)	3571	20	781	20	245	10	111	10	60	7	34	7
g_2 (+)	.384	.200	.412	.200	.432	.200	.445	.200	.460	.200	.479	.200
g_3 (+)	.33	.03	.48	.03	.65	.04	.83	.04	.90	.05	.95	.05

TABLE 2
Assignment examples

District Name (u_i)	$g_1(u_i)$	$g_2(u_i)$	$g_3(u_i)$	Category Min	Category Max
CHAPELLE-EN-VEXIN	87.912	0.437	0.36	2	4
NEUILLY-EN-VEXIN	70.000	0.500	0.30	3	5
GOUZANGREZ	219.231	0.468	0.78	1	3
TOURNAN-EN-BRIE	485.521	0.463	0.15	3	6
VERT-ST-DENIS	461.960	0.427	0.16	3	5
ST-GERMAIN-EN-LAYE	37.745	0.278	0.15	5	7
ULIS	4766.174	0.460	0.54	1	1

TABLE 3
Criteria weights

Weight	w_1	w_2	w_3
Inferred value	0.39	0.28	0.33

The application of Algorithm 2 using these preference parameters leads to the suitability map depicted in Figure 6. In this Figure, it appears that polygons with high suitability level correspond to counties near to Paris city.

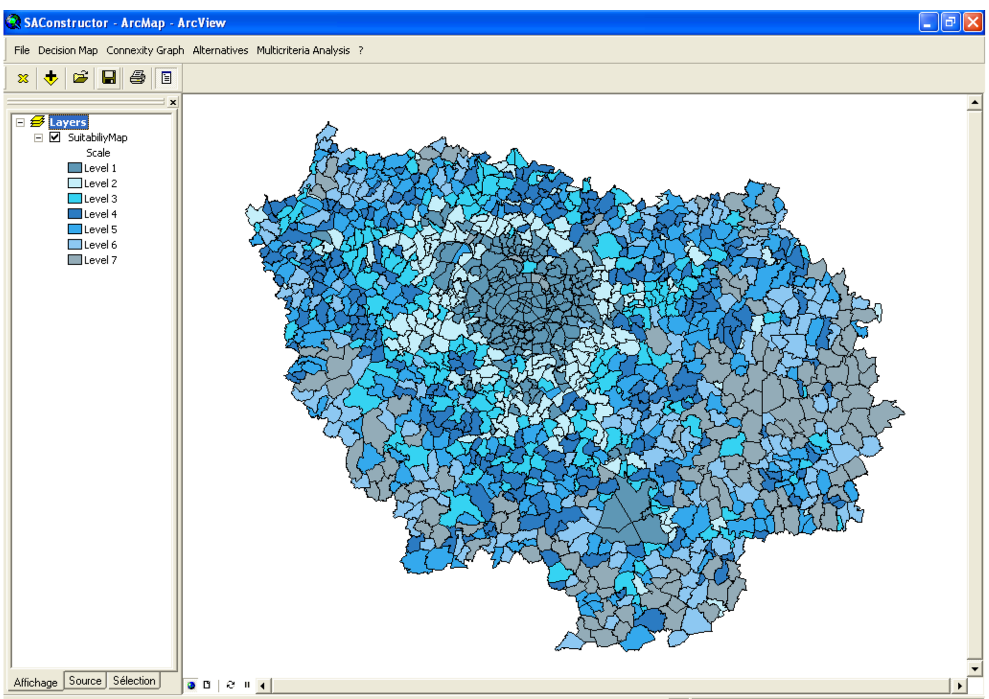

Fig. 6. Suitability map

5.5 Computing of efficient corridors

The map of Figure 6 is the input of the third phase. The connectivity graph derived (by Algorithm 3) from this map contains 1356 vertices and 3965 edges. The application of Algorithm 4 to identify the bi-objective efficient corridors leads to three solutions ((103.3,5), (60.4,6) and (41.0,7)), which are represented in Figures 7 and 8. Note that for this particular problem there does not exist corridors from Saclay to Roissy-en-France with suitability less than 5. As is shown in Figure 8, the shortest corridor crosses Paris city. The two other ones by-pass Paris city, avoiding dense counties and resulting in a longer corridors.

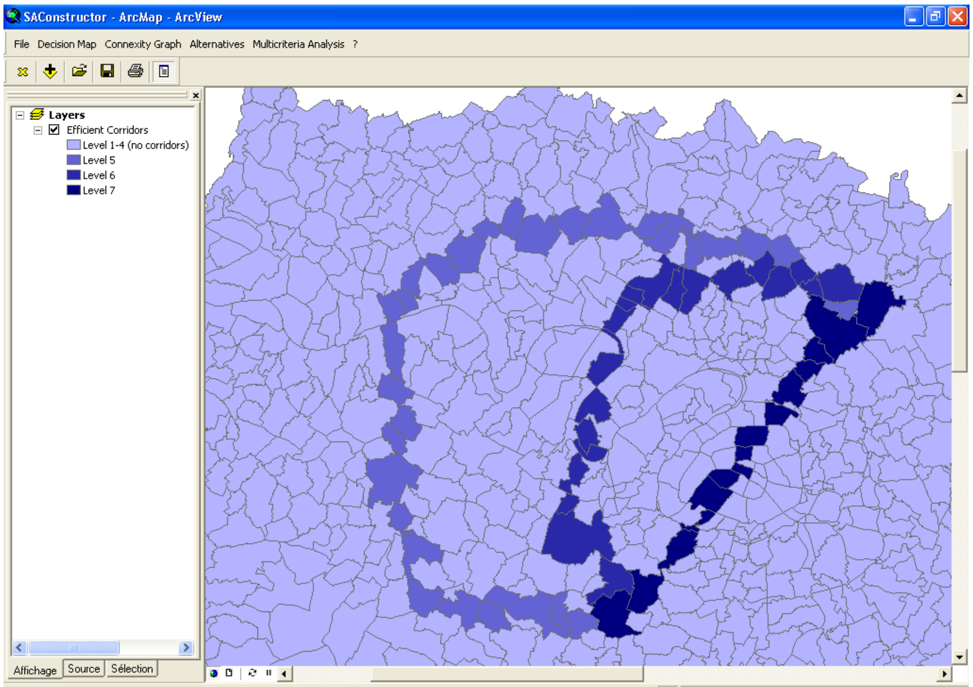

Fig. 7. Efficient corridors

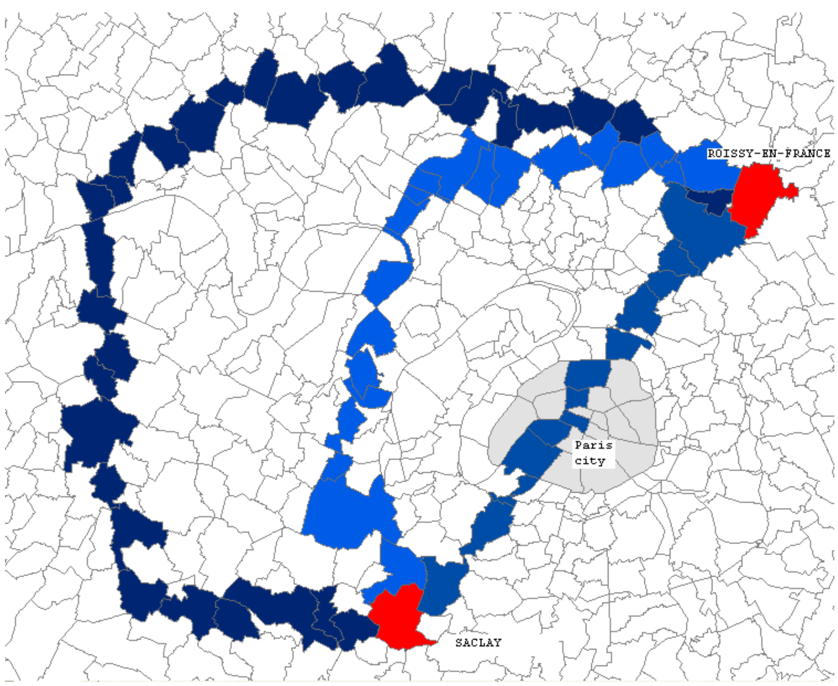

Fig. 8. A zoomed view of efficient corridors

For better illustration, consider the objective space given in Figure 9. The possible solutions is the set of points with coordinates (`Suitability`,`Distance`). At this level, it is important to note that since the `Suitability` axis is ordinal, the interval between the levels is arbitrary chosen. In the application described above, three efficient solutions have been identified (see Figure 9):

- point S_1 with coordinates (5,103.3) dominates all the other solutions in terms of suitability,
- point S_2 with coordinates (7,41.0) dominates all the other solutions in terms of distance, and
- point S_3 with coordinates (6,60.4) is not dominated neither by S_1 nor by S_2.

These three solutions represent the non-dominated points that achieve the best compromise between distance and suitability. All the other solutions are dominated by at least one of the points $\{S_1, S_2, S_3\}$.

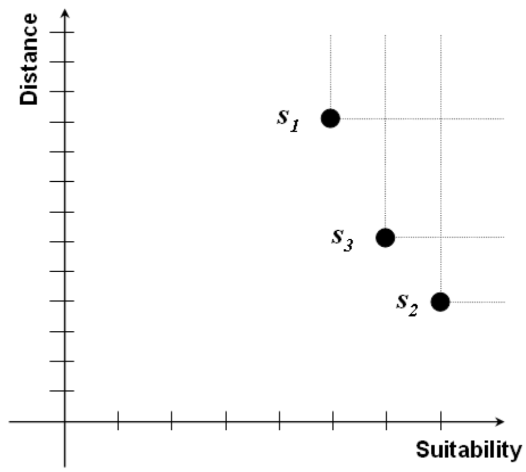

Fig. 9. Graphical representation of efficient set

6 COMPUTATIONAL PERFORMANCE

To study the behavior and the computational performance of Algorithm 4, we generated a set of random connectivity graphs. In order to do so, we used a raster representation with m rows and n columns. There exist different ways to define the neighborhood structure of a given pixel, namely Rook, Queen and Knight models; see, e.g. [21][35]. In our case, we used the Queen model. The distance value is picked-up randomly in the range 1 to 100, and suitability value is picked-up randomly in the range 1 to 7. As shown in Figure 10, the CPU time varies relatively slowly with the size of the connectivity graph. Even with large connectivity graphs, the computing time remains reasonable and makes it possible to proceed interactively with the decision maker.

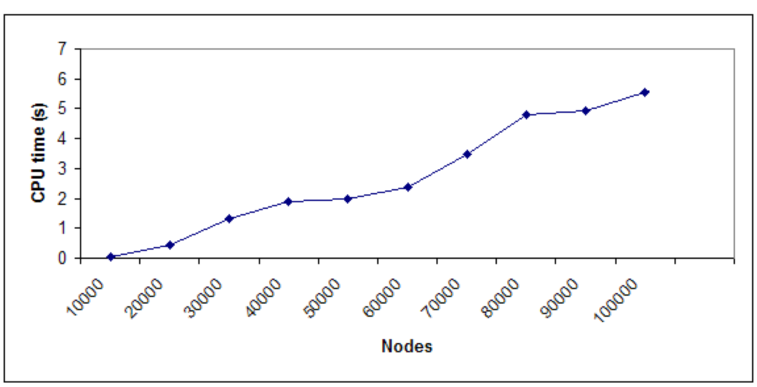

Fig. 10. Evolution of CPU time versus the number of nodes

In addition to CPU time, we summarized in Table 4 hereafter the number of solutions obtained for each connectivity graph. The limited number of the obtained solutions make it possible for the decision maker to identify the solutions corresponding to the best compromise in his/her point of view.

TABLE 4
Number of nodes versus number of solutions

Number of Nodes	CPU time	Number of solutions
1000	0.047	2
10000	0.453	7
20000	1.312	6
30000	1.875	6
40000	1.969	4
50000	2.390	7
60000	3.500	3
70000	4.781	5
80000	4.937	5
90000	5.546	7
100000	6.156	5

We also compared the proposed resolution algorithm to two other algorithms, namely SPAM and MOGADOR. The SPAM systematically varies the weights of objectives to obtain different aggregate cost surfaces [23]. A classical shortest-path algorithm is then applied on each cost surface to generate an alternative solution. MOGADOR is a genetic algorithm-based multiobjective algorithm proposed by [36]. Table 5 provides the CPU time (in seconds) and number of solutions for the algorithm proposed in this paper and SPAM and MOGADOR algorithms for two-objective shortest-path problems. Table 6 is similar to the previous one but it considers the case of three-objective shortest-path problems. Both tables show that the proposed algorithm performs better in all cases.

TABLE 5
CPU time (in seconds) and number of solutions for the proposed algorithm and SPAM and MOGADOR algorithms for two-objective shortest path problems

Number of	This paper		SPAM		MOGADOR	
Nodes	Time	Nb. opt. solutions	Time	Nb. opt. solutions	Time	Nb. opt. solutions
100	0.007	3	4	9	3	12
2500	1.523	7	11	31	20	122
62500	4.089	5	6827	66	534	173

TABLE 6
CPU time (in seconds) and number of solutions for the proposed algorithm and SPAM and MOGADOR algorithms for three-objective shortest path problems

Number of	This paper		SPAM		MOGADOR	
Nodes	Time	Nb. opt. solutions	Time	Nb. opt. solutions	Time	Nb. opt. solutions
100	2.300	5	94	12	3	18
2500	10.632	7	672	380	26	571
62500	20.422	7	$1.6\ 10^6$	?	2009	765

7 CONCLUSION AND FUTURE WORK

The paper presents a GIS-based multicriteria evaluation approach to construct a restricted set of corridors representing paths through which a linear facility can be constructed. The corridors are constructed as a collection of contiguous polygons and evaluated on two dimensions: (i) a qualitative dimension measuring the suitability for the corridor to cross a given polygon, and (ii) a quantitative dimension representing the length of the corridor. The corridor siting problem is hence modelled as a bi-objective shortest path problem: one MinMax criterion and one distance criterion to be minimized. The algorithm used to solve this bi-objective problem permits to substantially reduce the complexity of efficient set computing. The approach is validated through a prototype and illustrated to a corridor siting problem using real-world data.

In comparison to other proposals, the proposed approach has several merits. First, we think that the most innovative part of the approach is the way the corridors are defined. Conventionally, corridors are modelled by constraint based-suitability analysis. In this paper, corridors are constructed by combining different polygons. The main advantage of this idea is that it produces a limited number of solution alternatives, which is very useful in practice, as underlined by different authors [5][22][24][33]. Second, the bi-objective formulation of the problem obtained by considering two distinct dimensions—one quantitative and one qualitative—seems to us more appropriate in a preliminary study in a corridor siting problem. We think that this distinction is an important aspect of the approach because it avoids the “artificial” aggregation of quantitative and qualitative measures. Third, the resolution algorithm produces a reduced number of solutions, which makes it possible for the decision maker to (i) easily identify the solutions corresponding to the best compromise in his/her point of view; and (ii) to proceed interactively with the system.

Two points require more investigation. First, it could be interesting to extend the approach to the case in which more than two criteria are involved. This would lead, however, to providing to the decision maker pareto-front whose interpretation would be more difficult. Second, the use of a MinMax criterion is rather radical. The use of some more refined criteria would be necessary to capture in a more subtitle way the data concerning the qualitative dimension.

REFERENCES

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network flow theory, algorithms, and applications*. Prentice-Hall, Englewood-Cliffs, NJ, 1993.

[2] J. Akinyede. A geotechnical GIS concept for highway route planning. *ITC Journal*, 3:262–269, 1990.

[3] D. Atkinson, P. Deadman, D. Dudycha, and S. Traynor. Multi-criteria evaluation and least cost path analysis for an arctic all-weather road. *Applied Geography*, 3:287–307, 2005.

[4] S. Chakhar and J.-M. Martel. Enhancing geographical information systems capabilities with multi-criteria evaluation functions. *Journal of Geographic Information and Decision Analysis*, 7(2):47–71, 2003.

[5] S. Chakhar and V. Mousseau. GIS-based outranking multicriteria spatial modeling framework. *International Journal of Geographical Information Science*, 22(11-12):1159–1196, 2008.

[6] S. Chakhar and V. Mousseau. Multicriteria spatial decision support systems. In S. Shehkar and H. Xiong, editors, *Encyclopedia of Geographic Information Science*, pages 753–758. Springer-Verlag, 2008.

[7] S. Chakhar and V. Mousseau. Spatial multicriteria decision making. In S. Shehkar and H. Xiong, editors, *Encyclopedia of Geographic Information Science*, pages 747–753. Springer-Verlag, 2008.

[8] S. Chakhar, V. Mousseau, C. Pusceddu, and B. Roy. Decision map for spatial decision making. In *The 9th International Computers in Urban Planning and Urban Management Conference (CUPUM'05)*, July 29 - August 1, London, UK, 2005.

[9] N. Chrisman. *Exploring geographic information systems*. John Wiley & Sons, New York, second edition, 2002.

[10] W. Collischonn and J. Pilar. A direction dependent least-cost-path algorithm for roads and canals. *International Journal of Geographical Information Science*, 14(4):397–406, 2000.

[11] D. Cowen, J. Jensen, C. Hendrix, M. Hodgson, and S. Schill. A GIS-assisted rail construction econometric model that incorporates LIDAR data. *Photogrammetric Engineering & Remote Sensing*, 66(11):1323–1328, 2000.

[12] D. Cowen and W. Shirley. Integrated planning information systems. In D. Maguire, M. Goodchild, and D. Rhind, editors, *Geographical Information Systems: Principles and Applications*, volume 2, pages 297–310. Longman, London, UK, 1991.

[13] L. Dias and V. Mousseau. IRIS: A DSS for multiple criteria sorting problems. *Journal of Multi-Criteria Decision Analysis*, 12:285–298, 2003.

[14] L. Dias and V. Mousseau. Inferring ELECTRE's veto-related parameters from outranking examples. *European Journal of Operational Research*, 170(1):172–191, 2006.

[15] L. Dias, V. Mousseau, J. Figueira, and J. Clímaco. An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI. *European Journal of Operation Research*, 138:332–348, 2002.

[16] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[17] B. Dillon. GIS-based line-siting methodology. Technical report, Georgia Transmission Corp, 2005.

[18] ESRI. (Environmental Services Research Incorporation), ArcGIS® v. 9.2. Technical report, Redland, CA, 2006.

[19] J. Figueira, V. Mousseau, and B. Roy. *Multiple criteria decision analysis: State of the art surveys*, chapter ELECTRE methods, pages 133–162. Springer-Verlag, New York, 2005.

[20] X. Gandibleux, F. Beugnies, and S. Randriamasy. Martins's algorithm revisited for multi-objective shortest path problems with a MaxMin cost function. *4OR: A Quarterly Journal of Operations Research*, 4(1):47–59, 2006.
[21] M. Goodchild. An evaluation of lattice solutions to the corridor location problem. *Environment and Planning*, A9:727–738, 1977.
[22] G. Hall, F. Wang, and Subaryono. Comparison of boolean and fuzzy classification methods in land suitability analysis by using geographical information systems. *Environment and Planning A*, 24:497–516, 1992.
[23] D. Huber and R. Church. Transmission corridor location modeling. *Journal of Transportation Engineering*, 11:114–130, 1985.
[24] F. Joerin, M. Thériault, and A. Musy. Using GIS and outranking multicriteria analysis for land-use suitability assessment. *International Journal of Geographical Information*, 15(2):153–174, 2001.
[25] J. Malczewski. *GIS and multicriteria decision analysis*. John Wiley & Sons, New York, 1999.
[26] O. Marinoni. A discussion on the computational limitations of outranking methods for land-use suitability assessment. *International Journal of Geographical Information*, 20(1):69–87, 2006.
[27] E. Martins. On a special class of bicriterion path problems. *European Journal of Operation Research*, 17:85–94, 1984.
[28] V. Mousseau and R. Slowinski. Inferring an ELECTRE TRI model from assignment examples. *Journal of Global Optimization*, 12(2):157–174, 1998.
[29] V. Mousseau, R. Slowinski, and P. Zielniewicz. A user-oriented implementation of the ELECTRE TRI method integrating preference elicitation support. *Computers & Operations Research*, 27(7-8):757–777, 2000.
[30] A. Ngo The and V. Mousseau. Using assignment examples to infer category limits for the ELECTRE TRI method. *Journal of Multi-Criteria Decision Analysis*, 11(1):29–43, 2002.
[31] B. Roy. *Multicriteria methodology for decision aiding*. Kluwer Academic Publishers, Dordrecht, 1996.
[32] B. Roy and D. Bouyssou. *Aide multicritère à la décision: Méthodes et cas*. Economica, Paris, 1993.
[33] F. Wang. The use of artificial neural networks in geographical information system for agricultural land-suitability assessment. *Environment and Planning A*, 26:265–284, 1994.
[34] S. Wasi and J. Bender. Spatially enabled pipeline route optimization model. In *Proceedings of the International Pipeline Conference*, pages 699–707, Calgray, Alberta, Canada, October 4-8 2004.
[35] J. Xu and R. Lathrop. Improving simulation accuracy of spread phenomena in a raster-based geographic information system. *International Journal of Geographical Information Systems*, 9:153–168, 1995.
[36] X. Zhang and M. Armstrong. Genetic algorithms and the corridor location problem: Multiple objectives and alternative solutions. *Environment and Planning B: Planning and Design*, 35:148–168, 2008.