Using data analysis to approximate fastest paths on urban networks

Anjali Awasthi, Yves Lechevallier, Michel Parent, Jean-Marie Proth

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Using data analysis to approximate fastest paths on urban networks

Anjali Awasthi — Yves Lechevallier — Michel Parent — Jean-Marie Proth

N° 4961

October 2003

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4961--FR+ENG

Using data analysis to approximate fastest paths on urban networks

Anjali Awasthi*, Yves Lechevallier †, Michel Parent ‡, Jean-Marie Proth §

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet IMARA

Rapport de recherche n° 4961 — October 2003 — 19 pages

Abstract: Estimating shortest paths on large networks is a crucial problem for dynamic route guidance systems. The present paper proposes a statistical approach for approximating fastest paths on urban networks. The network data for statistical analysis is generated using a macroscopic traffic flow based simulation software. The input to the software are the input flows and the arc loads or the number of cars in each arc and the outputs from the software are the various paths joining the origins and the destinations of the network.

The network data obtained from the simulation software is subjected to hybrid clustering followed by canonical correlation analysis. The hybrid clustering comprises of two methods namely k-means and ward's hierarchical agglomerative clustering. The results of the data analysis are decision rules containing arc loads and input flows that govern the fastest paths on the network. These rules are used for predicting the paths to follow while arriving at the entrances of the network. Before entering the network, the arc loads and input flows provided by the rules are checked inside the network. If agreement is found, then the path obtained from the data analysis is the fastest path otherwise the shortest path is chosen as the fastest path.

Key-words: Traffic flow , Traffic Simulation, Fastest paths, Hybrid clustering, correlation analysis

* INRIA Rocquencourt, Project IMARA, domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex, France
† INRIA Rocquencourt, Project AXIS, domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex,France
‡ INRIA Rocquencourt, Project IMARA, domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex,France
§ INRIA / SAGEP, Bureau 410, UFR Scientifique, Université de Metz, Ile du Saulcy, 57012 Metz, France

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 —Télécopie : +33 1 39 63 53 30

Estimation de chemins les plus rapides sur les réseaux urbains en utilisant l'analyse de la donnée

Résumé : La recherche du chemin le plus rapide sur les grands réseaux est un problème crucial pour la conception de systèmes dynamiques d'aide au transport. Ce rapport de recherche propose une approche statistique pour calculer des chemins les plus rapides sur les réseaux urbains. Les données du réseau pour l'analyse statistique sont générées en utilisant un logiciel de simulation basé sur un modèle macroscopique de trafic. Les entrées du logiciel sont les flux à l'entrée du réseau et le nombre de voitures dans chaque arc, et les résultats du logiciel sont les chemins les plus courts entre les origines et les destinations du réseau.

Une approche de classification hybride, suivie d'une analyse de corrélation canonique, est appliquée aux données obtenues par simulation. L'approche de classification hybride est constituée de deux méthodes : la méthode connue sous le nom de K-mean analysis et la méthode de Ward. Le résultat de l'analyse des données est constitué de règles de décision. Chaque règle indique la décision à prendre en fonction de l'état de certains arcs et des flux d'entrée. Ces règles sont utilisées pour prédire les chemins à suivre en arrivant aux entrées du réseau. Avant d'entrer dans le réseau, l'état des arcs et des flux d'entrée sont vérifiés, et si ils correspondent aux conditions d'une règle, alors le chemin correspondant est retenu, sinon le chemin le plus court est retenu comme le chemin le plus rapide.

Mots-clés : Flux de véhicules, Simulation du trafic, Plus court chemin, Classification hybride, Corrélation canonique

1 Introduction

Finding shortest paths on real networks is a challenging problem. A number of methodologies have been proposed by researchers over recent years. Zhan [1997] presents a set of three shortest path algorithms that run fastest on real road networks. These are the graph growth algorithm implemented with two queues, the Dijkstra algorithm implemented with approximate buckets and the Dijkstra algorithm implemented with double buckets.

Smith, Chou and Romeijn [1998] introduced a hierarchical approach for solving shortest paths in a large scale network. Their main idea is to decompose the network into several smaller (low level) sub-networks. Given this decomposition, they approximate a shortest path between two arbitrary nodes of the original network by constraining the path to pass out of the sub-networks linking the origin and the destination.

Randomly varying input flows and arc states lead to dynamically changing states of the network. As a result, the fastest paths on the network do not remain constant and keep changing with time. Under this condition a methodology is required to determine the factors governing the fastest path. A statistical approach can be applied on the network data to find the relationship among the various network parameters. The network data for conducting the statistical study can be obtained in real-time or *in situ*.

Several researchers have used clustering techniques such as Kohonen Self-Organizing Feature Maps (SOFM) [Kohonen, 1998] with advanced neural networks for freeway travel time prediction. Jain *et al*[1999] present an overview of pattern clustering methods from a statistical pattern recognition perspective. Park and Rillett [1992], Dougherty [1995] and Faghri *et al*[1992] use Kohonen SOFM to transform an incoming signal pattern of multiple dimensions into a one- or two- dimensional discrete map and to perform this transformation adaptively in a topologically ordered fashion. Kisgyorgy & Rillett [2001] use Kohonen SOFM to classify the input vectors into different clusters where the vectors associated with each cluster have similar features.

We adopted a hybrid clustering statistical approach for finding the relationship among input flows, arc states and path travel times. The network data for the statistical study was obtained from a simulation software developed by us. The input to the simulation software is the network under varying input flows and arc states and the outputs are the fastest paths to join the origins and the destinations of the network.

The objective of this paper is to predict which one of the path between an input and an output node will be fastest on real networks under given input flows and states. The real networks are usually large in size and therefore the first step would be to reduce the complexity of the huge network by decomposition into smaller sub networks. These sub-networks will be linked to each other through common exit and entrance nodes. The second step would be to approximate the fastest paths in each of the sub-networks using hybrid clustering. Assuming that the first step has already been achieved, the present paper proposes a hybrid clustering based statistical approach to approximate the fastest paths on networks obtained from step 1.

This paper has been organized into 7 sections. In section 2, we introduce the notations and present the problem. Section 3 describes the hybrid clustering approach. Section

4 presents the research methodology used for approximating fastest paths on a network. Section 5 presents the application of method. Section 6 presents the validation of hybrid clustering results. Finally, the conclusion with future scope of research is presented in section 7.

2 Problem Setting

Let us represent a traffic network by a directed graph G = (N,A) where N is the set of nodes and A is the set of directed arcs. Let us denote the set of entrance nodes by E and set of destination nodes by D. An arc is denoted by *(i,j)* with tail node i and head node j. Each arc *(i,j)* has three attributes: its length l_{ij}, its traffic accommodation capacity c_{ij} and the number of vehicles inside the arc at time t denoted by $n_{ij}(t)$.

At time t_0, the network is in initial state. At time $t \geq t_0$, varying input flows start arriving at the entrance nodes of the network. This will lead to change in the states of the arcs i.e. in the number of cars in the arcs. The changing arc states combined with varying input flows generate continuously changing fastest paths and therefore, dynamic travel times on the network. The problem lies in predicting which one of these paths between an input and an output will be the fastest according to the input flows and the state of the network. The following parameters define the state of the network and act as an input to the computation of fastest paths:

- The number of cars $n_{ij}(t_0)$ in each arc at time t_0 (also called initial state or initial load of the arcs).
- The input flows ϕ_e for all the entrance nodes e. We assume that the input flows are piecewise constant on time intervals $[t_0, t_1], [t_1, t_2], [t_2, t_3], ..$ etc.

The outputs of the computation are the fastest paths that join the entrances to the exits of the network. All the input and the output parameters of the network own a rank depending upon their value. The ranks range from 1 to 3. The ranking of paths is done on the basis of travel times. A path is assigned rank 1 if it is the fastest, 2 if it is the second fastest and 3 otherwise. Input flows are ranked 1 if they lie between $0-60\%$ of the maximum input flow, 2 if they lie between $60-90\%$ and rank 3 if they lie between $90-100\%$. The arcs are ranked on the basis of their initial states. An arc is given rank 1 if it lies between $0-60\%$ of the maximal capacity, 2 if it lies between $60-90\%$ and 3 if it lies between $90-100\%$.

For determining the network parameters that govern the fastest paths on a network, we need to find those arcs and input flows whose ranks decide the paths of rank 1. Once these arcs and input flow ranks are found, then predictions about the fastest paths between origins and destinations can be made at the entrances itself. This would mean that at the network entrances, we will verify the ranks of the input flows and the states of arcs with those obtained from statistical analysis. If similarities are found then the path obtained from the statistical analysis will be chosen, otherwise the shortest path will be the best path to follow.

The problem is to approximate the best paths to join the origins and the destinations of the network under varying input flows and arc states.

3 Hybrid Clustering

Hybrid clustering is a statistical approach combining the partitional and hierarchical clustering methods. Hybrid algorithms have been proposed by Cutting *et al*(1992) as in the famous 'Scatter/Gather' approach for reducing the computational complexity. A similar idea has also been used by Murtagh(1995), Meila and Heckerman (1998), and Ambroise *et al*(2000) where Hierarchical Agglomerative Clustering is used to supply initial cluster centers for subsequent Expectation-Maximization (EM) based partitional clustering.

In the remaining of this paper we first apply a partitional method called k-means clustering followed by a hierarchical clustering method called Hierarchical Agglomerative Clustering (HAC). The goal of k-means clustering is to decompose the initial set of elements into k^o clusters. The HAC method then iteratively merges clusters that are closest to each other. The result is $k^1 < k^o$ clusters.

Let us explain these two methods. The k-means clustering name comes from the representation of each of the k clusters of a sample by their mean or weighted average of points [Diday *et al*, 1976, 1978]. Consider a graph G = (N,A) with N as the set of nodes and A as the set of arcs. An arc is represented by (i, j) and has a value $d(i, j)$. The k-means clustering algorithm can be summarized as:

Step 1: Initialization
Let N = {1,2,3...n}. We select $k^o < n$ elements from N at random. These elements represent the initial centres of the partition and are denoted by U_z^o. The set containing the centres $U_1^o, U_2^o, ...$ etc. is represented by S^o. In other words $S^o = \{U_1^o, U_2^o, U_3^o, ..U_{k^o}^o\}$ where $S^o \subset N$.

Step 2: Assignment
For i = 1,2,..n we assign an element i to cluster C_z^o if

$$d(i, U_z^o) = Min_{\{j=1,2,..k^o\}} d(i, U_j^o)$$

Step 3: Computation of S^1
For each C_z^o, we compute the mean value of the elements of C_z^o. Let U_z^1 be this value and $S^1 = \{U_1^1, ...U_{k^1}^1\}$. Note that we may have $k^1 < k^o$.

Step 4: Test
If $S^1 \equiv S^o$ stop, otherwise set $k^o = k^1, S^o = S^1$ and go to step 2.

It has been proved that this algorithm converges. The second method of hybrid clustering is based on agglomerative hierarchical clustering and is called the Ward's method. Ward (1963) proposes that at any stage of an analysis the loss of information which results from the

grouping of individuals into clusters can be measured by the total sum of squared deviations of every point from the mean of the cluster to which it belongs. At each step in the analysis, union of every possible pair of clusters is considered and the two clusters whose fusion results in the minimum increase in the Error Sum of Squares (E.S.S) are combined. The E.S.S is given by:

$$E.S.S. = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (x_{ij} - \overline{x_j})^2$$

where k represents the total number of clusters obtained from k-means method. The total number of elements in cluster j are n_j where $j = 1, 2, 3..k$. The elements of cluster j are represented by x_{ij} where $i = 1, 2, 3...n_j$. The mean $\overline{x_j}$ of the cluster j is represented by $\overline{x_j} = (1/n_j)\sum_{i=1}^{n_j} x_{ij}$. At stage one, each individual is regarded as a single member group and so E.S.S. is 0. The two individuals whose fusion results in the minimum increase in the E.S.S form the first group and so on.

There are several benefits resulting from this methodology. First, it can alleviate to some extent the initialization problem associated with partitional methods since the second step is a fine-tuning step and prunes bad initial clusters. It can also form more complex cluster shapes. Second, we can get around the high computational complexity of the HAC algorithm by running it on K1 clusters instead of starting from N singleton clusters. This hybrid approach has a complexity of $O(K_1^2 N)$ which is linear in N.

4 Methodology

The methodology used to approximate the fastest paths between the origins and destinations of a network can be summarized in the following four steps:

- Storage of network data for statistical analysis.
- Using cluster analysis to find the relationship among the network parameters namely path travel times, arc states and input flows.
- Using canonical correlation analysis to determine the strength of relationship among the network parameters.
- Validation of the results obtained from the statistical method.

We use a simulation software to emulate the evolution of the state of the network with regard to the parameters of the system. The input to the simulation is a network with varying input flows and arc states that are generated randomly. Both the input and the output parameters are ranked between 1, 2 and 3: the higher the ranking of the input flow the greater the flow, the higher the state of an arc the greater the number of vehicles present inside it, and the higher the ranking of a path the greater the travel time on the path.

The relationship among the ranked paths, arcs and input flows was found using hybrid clustering. The hybrid clustering process divides the network data into clusters that contain

Table 1: Input Flow vs Rank

Input Flow ϕ_e	*Rank*
$0 < \phi_e \leq 0.6$* Maximum Input Flow	1
0.6* Maximum Input Flow $\leq \phi_e \leq$ 0.9* Maximum Input Flow	2
0.9* Maximum Input Flow $\leq \phi_e \leq$ Maximum Input Flow	3

sets of variable values which are close to each other and distant from sets belonging to other clusters. Each cluster contains sets made of paths, arcs and input flows with their rank. Since we are interested in studying only those network parameters that govern the fastest paths, we retain only the clusters containing paths of rank 1 and the rest are discarded.

To measure the strength of relationship among arc states, input flows and paths, we subjected the data obtained from hybrid clustering to canonical correlation analysis. A threshold value was set for the correlation coefficient. Variables having correlation coefficient value greater than or equal to the threshold value were retained and the others were rejected.

Finally, the results obtained from the statistical analysis were validated by comparison with other sources. In our case, we compared the statistical results with simulation results.

5 Application

Let us consider the network depicted in Figure 1. To compute the output paths on the network, it was subjected to randomly varying input flows and arc states under the ranges presented in Table 1 and 2. Each output path was assigned a ranking using the criteria presented in Table 3. The values for maximum input flows are presented in Table 4. The 4^{th} column of table 4 presents the identities of the ranks assigned to the input flows. For instance, I1, I2 and I3 represent rank 1,2 and 3 for the input flow at node 1 and I4, I5 and I6 represent the ranks identities for input flow at node 2. The lengths l_{ij}, traffic accommodation capacities c_{ij}, probabilities p_{ij} that is the proportion of flow arriving at node i that wants to be directed to node j, initial state $n_{ij}(t_0)$ and the speed ν_{ij} of the vehicles for all the arcs $(i, j) \in (N, A)$ are presented in Table 5. The 7^{th} column of table 5 presents the identities assigned to various arcs of the network on the basis of their ranks. For instance, arc (1,3) is denoted S1, S2 or S3 if its rank is 1,2 or 3 respectively. The same naming principle applies for other arcs.

Table 6 presents the different paths of the network connecting the origins and the destinations. The 4^{th} column of table 6 presents the identities of the ranked paths. For instance, P1, P2 and P3 represent the rank 1,2 and 3 for path 1-3-7-10. The total number of simulations conducted were 10,000. The number of iterations carried out per simulation were 200. The simulation data was generated by subjecting the network to randomly varying input flows and arc states for obtaining the various ranked paths of the network. Hybrid clustering was performed on the simulated data using SPAD statistical software (CISIA, 1997). Four

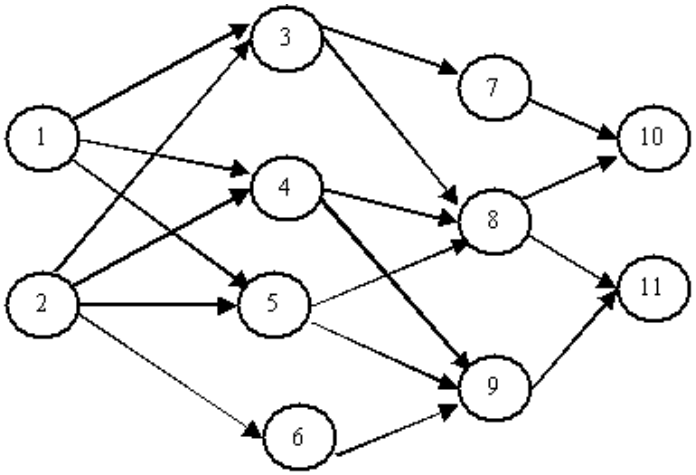

Figure 1: A Multiple Origin Destination Network

Table 2: Arc State vs Rank

Arc State $n_{ij}(t_0)$	*Rank*
$0 < n_{ij}(t_0) \leq$ 0.6* Maximum State	1
0.6*Maximum State$\leq n_{ij}(t_0) \leq$ 0.9* Maximum State	2
0.9* Maximum State$\leq n_{ij}(t_0) \leq$ Maximum State	3

Table 3: Path Travel Time vs Rank

Paths	*Rank*
$1^{st} FastestPath$	1
$2^{nd} FastestPath$	2
$\geq 3^{rd} FastestPath$	3

Table 4: Input Flows

Origin Node	*Time Interval*	*Maximum Input Flow*	*Input flow Identities for ranks 1,2 & 3*
1	0-200	70	I1,I2,I3
2	0-200	70	I4,I5,I6

Table 5: Network Parameters at time t_o

Arc Id (i,j)	p_{ij}	ν_{ij}	c_{ij}	l_{ij}	$n_{ij}(t_0)$	*Arc identities as per ranks 1,2 & 3*
(1,3)	0.4	1	25	7	175	S1,S2,S3
(1,4)	0.3	1	30	7	210	S4,S5,S6
(1,5)	0.3	1	30	8	240	S7,S8,S9
(2,3)	0.2	1	20	10	200	S10,S11,S12
(2,4)	0.4	1	20	8	160	S13,S14,S15
(2,5)	0.2	1	25	6	150	S16,S17,S18
(2,6)	0.2	1	30	6	180	S19,S20,S21
(3,7)	0.5	1	20	13	260	S22,S23,S24
(3,8)	0.5	1	18	10	180	S25,S26,S27
(4,8)	0.4	1	15	11	165	S28,S29,S30
(4,9)	0.6	1	12	11	132	S31,S32,S33
(5,8)	0.3	1	16	12	192	S34,S35,S36
(5,9)	0.7	1	20	9	180	S37,S38,S39
(6,9)	1.0	1	14	10	140	S40,S41,S42
(7,10)	1.0	1	17	7	119	S43,S44,S45
(8,10)	0.5	1	15	8	120	S46,S47,S48
(8,11)	0.5	1	13	10	130	S49,S50,S51
(9,11)	1.0	1	12	11	132	S52,S53,S54

predefined methods of SPAD were used while conducting the statistical analysis. These four methods are CORMU, DEFAC, SEMIS and PARTI/DECLA. Figure 2 depicts the four methods of SPAD.

The first method entitled CORMU represents Multiple Correspondence Analysis (MCA) and was used to examine patterns in the data. Before subjecting to MCA all the variables were classified as nominal active and variables contributing less than 2% of the total mass were discarded. Figure 3 presents the results obtained from MCA. The paths, arcs and input flows are represented in figure 3 by their identities.

It can be seen in figure 3 that the paths and arcs are well-separated with most of the arcs and input flows located around the centre and the paths spread far away. Only few arcs are situated close to the paths and can be seen around the centre (for instance P39 and S39). The MCA findings yield that a total of 59 factors or axes were required to explain 100% of the total inertia.

The second method entitled DEFAC was used to describe the details of the factors obtained from the Multiple Correspondence Factor Analysis. Since only first 10 axes or factors seemed to contribute significantly out of the 59 factors obtained from MCA, we decided to retain the first ten factors for further study. The variance explained by the first

Table 6: Path Details

O-D Pair	*Paths*	*Length*	*Path identities for ranks 1,2 & 3*
1-10	1-3-7-10	27	P1,P2,P3
	1-3-8-10	25	P4,P5,P6
	1-4-8-10	26	P7,P8,P9
	1-5-8-10	28	P10,P11,P12
1-11	1-3-8-11	27	P13,P14,P15
	1-4-8-11	28	P16,P17,P18
	1-4-9-11	29	P19,P20,P21
	1-5-8-11	30	P22,P23,P24
	1-5-9-11	28	P25,P26,P27
2-10	2-3-7-10	30	P28,P29,P30
	2-3-8-10	28	P31,P32,P33
	2-4-8-10	27	P34,P35,P36
	2-5-8-10	26	P37,P38,P39
2-11	2-3-8-11	30	P40,P41,P42
	2-4-8-11	29	P43,P44,P45
	2-4-9-11	30	P46,P47,P48
	2-5-8-11	28	P49,P50,P51
	2-5-9-11	26	P52,P53,P54
	2-6-9-11	27	P55,P56,P57

ten factors are 7.55%,7.35%,6.33%,3.59%, 2.59%, 2.01%, 2.0%, 1.86%, 1.85% and 1.84% which contributes to 36.96% of the total inertia.

The third method entitled SEMIS was used to perform the k-means clustering on the variables obtained from the 10 factors of MCA. For generating the clusters in a partition, 10 centres were randomly chosen and 20 iterations of k-means clustering method were conducted. Likewise three partitions each containing 10 clusters were generated. These three partitions were then intersected to obtain the final 26 stable classes that acted as input for the Ward's agglomerative hierarchical clustering.

The fourth method PARTI/DECLA was used to perform the Ward's method. The 26 stable classes obtained from SEMIS were subjected to Ward's agglomerative hierarchical clustering. The output of the hierarchical clustering is a dendogram (Figure 4) that is later partitioned into 7 clusters.

Figure 5 presents the 7 clusters obtained from PARTI/DECLA. Each cluster contains individuals called parangons that are normally the closest points to the centre of each class and best represent the cluster. The parangons can be seen at the centre of each cluster in figure 5.

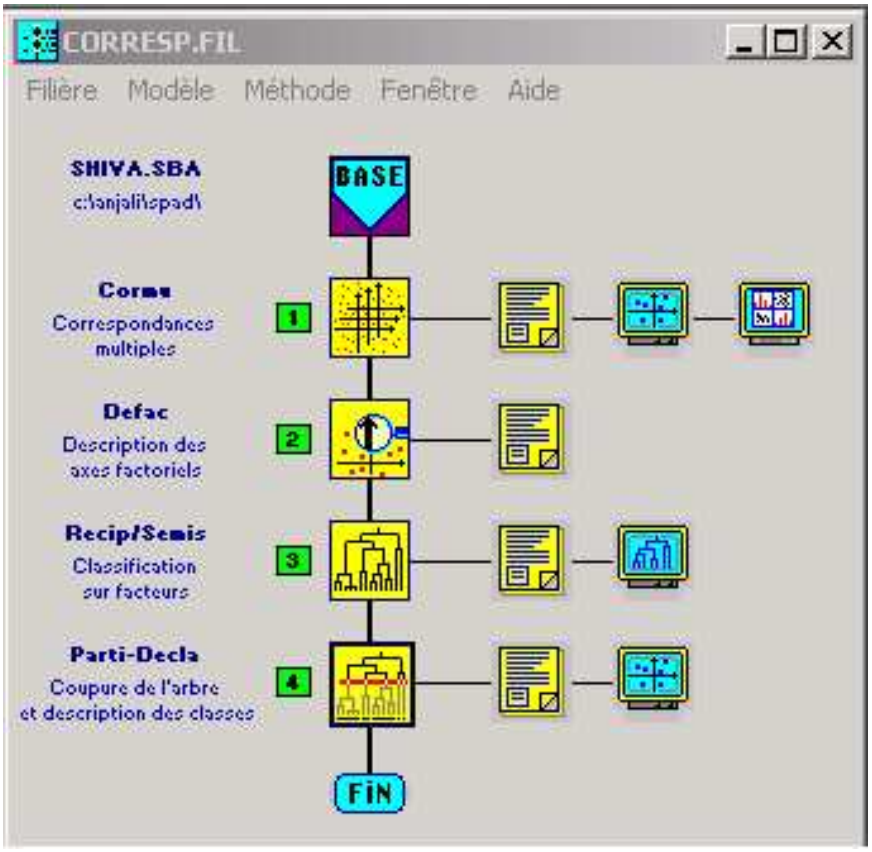

Figure 2: Four Methods in SPAD

Table 7 presents the 7 clusters and the identities of the paths, arcs and input flows present in each cluster. The clusters were subjected to canonical correlation analysis to measure the strength of association between the network variables. The canonical R value for each cluster was greater than 0.56. It can be seen in Table 7 that all the clusters contain paths and arc states while input flows are present only in clusters 2,4,5,6 and 7.

A threshold value equal to 0.1 was chosen for the correlation coefficient between the input and the output variables. The elements of each cluster were checked for threshold value and those meeting the criteria were retained.

Table 8 presents the canonical correlation analysis results for the fastest paths obtained from the 7 clusters. It can be seen in Table 8 that the fastest paths are more correlated with arc states than the input flows. The reason being the weak correlation coefficient between the input flows and the paths of rank 1 which was found to be less than 0.1 in all the 7

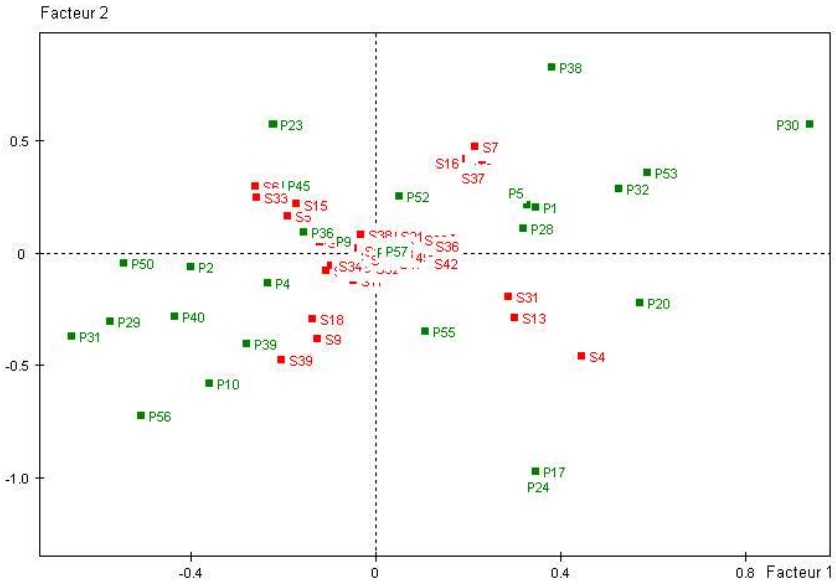

Figure 3: Multiple Correspondence Analysis

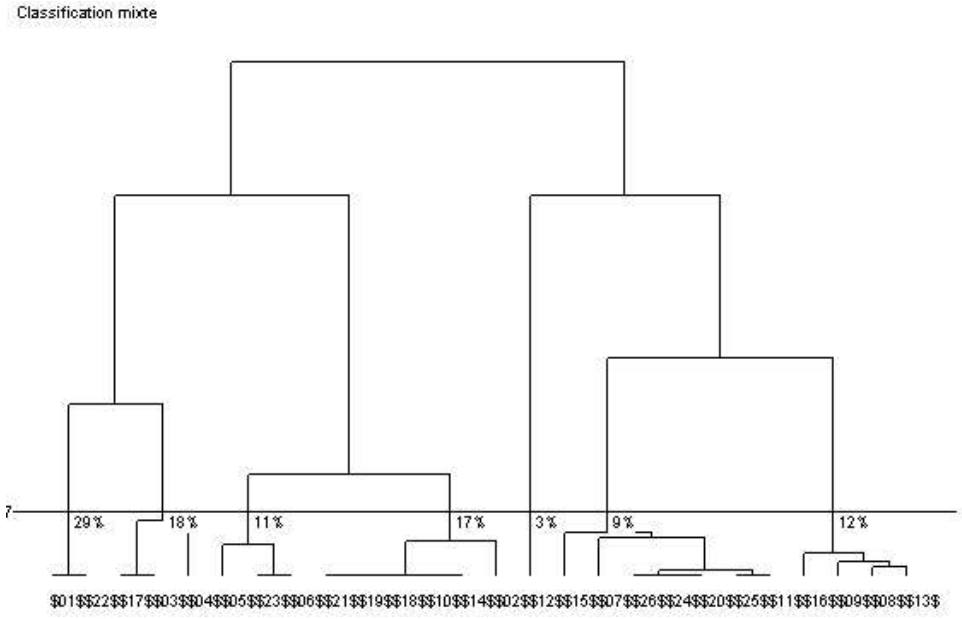

Figure 4: Ward's Hierarchical Tree/Dendogram

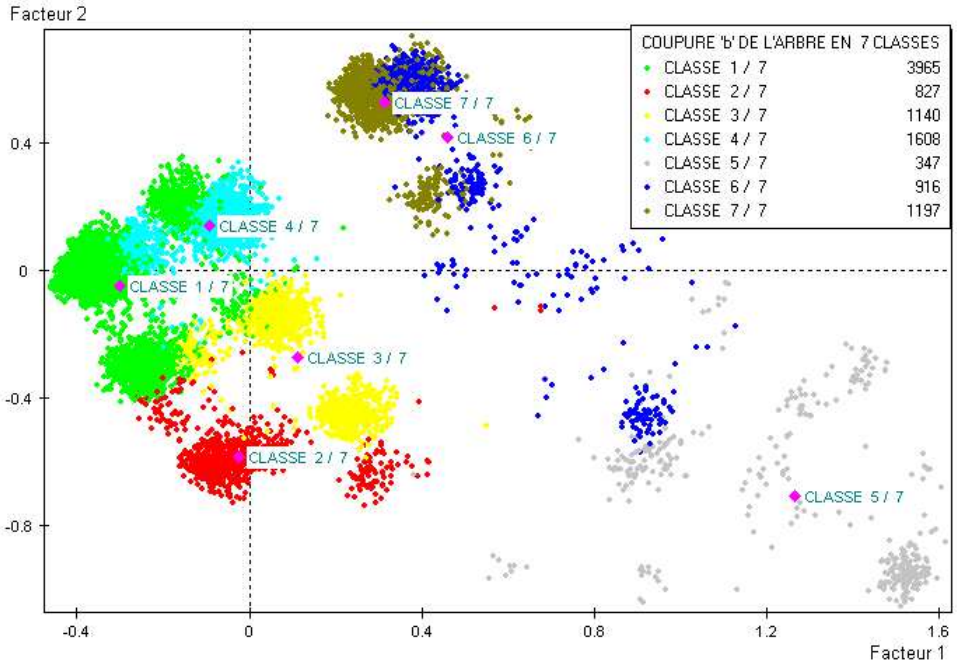

Figure 5: Cluster Analysis

clusters. Thus, we can say that in this example the time required by a car to join an input node to an output node depends on the state of the arcs but not on the input flows at the instant the car cross the input node. It is not true for any network since it depends on the situation of the input nodes.

The information depicted in Table 8 can be used to make predictions about the fastest paths on the network. For instance, using table 8 we can say that if at the entrance of the network we observe that arcs (1,3), (2,3), (3,8), (4,8) and (5,8) have state ranking 3 i.e. their state lies between 0.9*maximum state and maximum state, then the fastest path to take between 1 and 10 is 1-3-7-10. It can also be seen in table 8 that one path can be the fastest path under two different states of an arc. For instance, consider the path 2-3-8-10 for origin-destination pair 2-10. This path is the fastest path when arcs (3,8), (4,8) and (5,8) have rank 1 and arc (5,9) has rank 3. However for arc (4,9), we observe two ranks 1 and 3. It can be seen that positive correlation (+.11) exists for rank 3 and negative correlation(-.13) exists for rank 1. This means that the path 2-3-8-10 is chosen as the fastest path whenever rank 3 for arc(4,9) is observed and if rank 1 is observed then the path 2-3-8-10 should be avoided.

For the combination of other arc states not presented in table 8, the correlations between the fastest paths and the arc loads was found to be less than 0.1 indicating weak influence of

Table 7: Seven Clusters

Cluster	*Frequency*	*Elements*
1	.2868	P31,P2,P4,P18,P23,P50,P29,P40,P45,P36 S33,S34,P39,S28,S25,P9,S6,P38,S15,S5,S10,P13 S1,S14,S38,P12,S40
2	.1848	P17,P24,P31,P29,P2,P4,P40,P44 P51,P39,S4,S39,S25,S13,S34,S9,P9,P18 S28,S31,S10,S1,S32,P35,P13,I1,S8,S19,S17,P12
3	.1092	P17,P24,P28,P5,P1,P32,P39,P44,P40 P51,S31,S36,S27,S9,S30,S39,P36,S4,S18,P6 S3,S13,S29,S26,S42,S12
4	.1732	P5,P1,P28,P50,P32,P23,P18,P39,P40,S12 P45,S36,S30,S27,P36,P9,S6,S33,S26,S5,S15,S3 S29,S14,P13,S21,S9,S37,S32,S35,S8,P12,I2,P57
5	.0347	P34,P8,P3,P43,P24,P30,P51,S4,P17 P41,S13,P35,P33,P32,P4,S28,S25,S34,P44,S31 S42,S49,S32,S39,I1
6	.0917	P33,P49,P41,P1,P5,P38,P37,P28 P35,S37,S7,S27,S16,S36,S30,S31,P18,P43,P23 S12,S3,P57,P9,S13,S42,I3
7	.1196	P37,P49,P30,P41,P32,P18,P23,S7,P2 P4,P45,S16,S37,S34,S25,P36,S28,P9,P57,S15 S33,S1,S21,S52,I1

other arc ranks on the fastest paths of the network. Under these conditions we will consider the shortest paths to be also the fastest paths of the network.

Analysis of table 8 leads us to the following rules for determining the fastest paths between various origin-destination pairs of the network under any given combination of input flows and arc states. In the rules, the rank of arc (i, j) is denoted by $r(i, j)$ and the boolean operators "equal to" and "not equal to" are denoted by "=" and "!=" respectively. Each rule comprises of several combinations of sets of certain arc states. These arc sets are arranged in the rules in the increasing order of their risk for deciding the fastest path. For instance, consider the rule for origin node 2 and destination node 10.

IF {r(1,4) = 1 AND r(2,4) = 1} **THEN** the shortest path is (2,4,8,10).

Using this rule we can say that if both the arcs (1,4) and (2,4) are of rank 1, then the risk associated with choosing path 2-4-8-10 as the fastest path is less compared to the case when only one of the arcs has state rank 1. In other words, the probability of 2-4-8-10 being the fastest path is higher when both the arcs (1,4) and (2,4) have rank 1 than the case when

Table 8: Fastest Paths vs Arc States

O-D Pair	*Fastest Path*	*Arc States*
1-10	1-3-7-10(1)	(1,3)(3)(.11),(2,3)(3)(.103),(3,8)(3)(.295), (4,8)(3)(.27), (5,8)(3)(.31)
	1-3-8-10(1)	(1,3)(1)(.13),(2,3)(1)(.13),(3,8)(1)(.37), (4,8)(1)(.34), (5,8)(1)(.37)
	1-3-8-10(1)	(4,8)(1)(.34),(5,8)(1)(.37),(3,8)(1)(.37)
	1-3-8-10(1)	(5,8)(1)(.37),(3,8)(1)(.37),(4,8)(1)(.34), (1,3)(1)(.13)
1-11	1-3-8-11(1)	Indifferent to arc states
2-10	2-3-7-10(1)	(3,8)(3)(.21),(4,8)(3)(.25),(4,9)(1)(.13), (5,8)(3)(.302)
	2-3-7-10(1)	(3,8)(3)(.21),(4,8)(3)(.25),(4,9)(3)(-.13), (5,8)(3)(.302)
	2-3-8-10(1)	(3,8)(1)(.19),(4,8)(1)(.19), (4,9)(3)(.11), (5,8)(1)(.22)
	2-3-8-10(1)	(3,8)(1)(.19),(4,8)(1)(.19), (4,9)(1)(-.13), (5,8)(1)(.22),(5,9)(3)(.23)
	2-4-8-10(1)	(1,4)(1)(.2107),(2,4)(1)(.1526)
	2-5-8-10(1)	(1,5)(1)(.28),(2,5)(1)(.25),(5,9)(1)(.19)
2-11	2-3-8-11(1)	(1,5)(3)(.203),(2,5)(3)(.174),(5,9)(3)(.23)
	2-3-8-11(1)	(1,5)(3)(.203),(5,9)(1)(-.23)
	2-4-8-11(1)	(1,4)(1)(.24),(2,4)(1)(.16)
	2-4-8-11(1)	(2,4)(1)(.16)
	2-5-8-11(1)	(5,9)(1)(.24),(1,5)(1)(.31),(2,5)(1)(.27)
	2-5-8-11(1)	(1,5)(1)(.31),(2,5)(1)(.27),(5,9)(1)(.24), (4,9)(1)(.13)

only one of the arcs has state rank 1. The various rules used to make decisions about the fastest paths among the origins and the destinations of the network depicted in figure 1 are mentioned below. In this set of conditions, the two first conditions must apply. The greater the number of conditions that apply, the less the risk attached to this decision.

ORIGIN 1 - DESTINATION 10

IF {r(5,8) = 3 AND r(3,8) = 3 AND r(4,8) = 3 AND r(1,3) = 3 AND r(2,3) = 3} **THEN** the fastest path is (1,3,7,10).

IF {r(5,8) = 1 AND r(3,8) = 1 AND r(4,8) = 1 AND r(1,3) = 1 AND r(2,3) = 1} **THEN** the fastest path is (1,3,8,10).

IF {r(5,8) = 2} **THEN** the fastest path is (1,3,8,10).

ORIGIN 1 - DESTINATION 11

Choose always the shortest path (1,3,8,11) as the fastest path.

ORIGIN 2 - DESTINATION 10

IF {r(5,8) = 3 AND r(4,8) = 3 AND r(3,8) = 3 AND r(4,9) != 3} **THEN** the fastest path is (2,3,7,10).

IF {r(5,9) = 3 AND r(5,8) = 1 AND r(4,8) = 1 AND r(4,9) != 1 AND r(3,8) = 1 } **THEN** the fastest path is (2,3,8,10).

IF {r(1,4) = 1 AND r(2,4) = 1} **THEN** the fastest path is (2,4,8,10).

IF {r(1,5) = 1 AND r(2,5) = 1 AND r(5,9) = 1 } **THEN** the fastest path is (2,5,8,10).

IF { r(5,8) != 3 AND r(5,9) != 3 AND r(1,4) != 1 AND r(1,5) != 1 } **THEN** the fastest path is (2,5,8,10).

ORIGIN 2 - DESTINATION 11

IF {r(5,9) != 1 AND r(1,5) = 3 AND r(2,5) = 3 } **THEN** the fastest path is (2,3,8,11).

IF {r(1,4) = 1 AND r(2,4) = 1} **THEN** the fastest path is (2,4,8,11).

IF {r(1,5) = 1 AND r(2,5) = 1 AND r(5,9) = 1 AND r(4,9) = 1 } **THEN** the fastest path is (2,5,8,11).

IF {r(5,9) = 1 AND r(1,4) != 1 AND r(1,5) != 1 } **THEN** Choose (2,5,9,11) as the fastest path.

6 Validation

The results obtained from the statistical analysis were validated by comparison with simulation results. The network considered for the simulation study is depicted in Figure 1 and the network parameters used are presented in Table 5. Table 9 contains the input flow and initial state data for the simulation network. Table 10 and 11 present the results obtained from the simulation and the statistical analysis.

It can be seen in table 10 that the simulation result for the fastest path between origin-destination pair 1 and 10 is 1-3-8-10, 1 and 11 is 1-3-8-11, 2 and 10 is 2-5-8-10 and 2 and 11 is 2-5-8-11. Using the rules derived from table 8, we observe that path 1-3-8-10 is the fastest path between origin 1 and destination 10 when the arcs (1,3),(2,3),(3,8),(4,8) and (5,8) have state ranks 1 or only arc (5,8) has rank 2, path 1-3-8-11 is always the fastest path between origin 1 and destination 11 irrespective of the arc ranks, path 2-5-8-10 is the

Table 9: Initial States

Input Flow at Node 1	*Input Flow at Node 2*	*Arcs with their initial states*
70	60	(1,3)(150), (1,4)(203),(1,5)(80),(2,3)(190), (2,4)(152),(2,5)(60),(2,6)(174),(3,7)(130), (3,8)(170),(4,8)(55),(4,9)(121),(5,8)(144), (5,9)(135),(6,9)(60),(7,10)(112),(8,10)(96), (8,11)(50),(9,11)(121)

Table 10: Simulation Results

Input	*Output*	*Fastest path*
1	10	1-3-8-10
1	11	1-3-8-11
2	10	2-5-8-10
2	11	2-5-8-11

Table 11: Results provided by the rules

Input Node	*Output Node*	*Rule*	*Fastest path*
1	10	**IF** {r(5,8) = 2} **THEN** the fastest path is (1,3,8,10).	1-3-8-10
1	11	Choose always the shortest path (1,3,8,11) as the fastest path.	1-3-8-11
2	10	**IF** {r(1,5) = 1 AND r(2,5) = 1 AND r(5,9) = 1 } **THEN** the fastest path is (2,5,8,10).	2-5-8-10
2	11	**IF** {r(1,5) = 1 AND r(2,5) = 1} AND r(5,9) = 1 AND r(4,9) = 1 } **THEN** the fastest path is (2,5,8,11).	2-5-8-11

fastest path between origin 2 and destination 10 when the arcs (1,5),(2,5) and (5,9) have rank 1. The path 2-5-8-11 is the fastest path between origin 2 and destination 11 when the arcs (1,5),(2,5),(5,9) and (4,9) have state rank 1.

Let us now validate the statistical results of table 11 with the simulation results of table 10. On verifying the statistical results of table 11 with the input flow and arc data presented in table 9, we find that arc (5,8) has rank 2 which justifies the selection of 1-3-8-10 as the fastest path between 1 and 10. It can be seen in table 11 that the fastest path between 1 and 11 is 1-3-8-11 which conforms with the table 10 results and remains uninfluenced by the arc's state ranks. The statistical results for fastest paths between 2 and 10 and 2 and 11 are 2-5-8-10 and 2-5-8-11 which can be attributed to the rank 1 of arcs (1,5) and (2,5) and this result is in agreement with the simulation results and the arc states presented in table 9. Therefore, we can say that the results obtained from the simulation experiment follow the findings obtained from the hybrid clustering statistical approach.

7 Conclusions

This paper presents a statistical approach for approximating fastest paths under stepwise constant input flows and initial states of the arcs on urban networks. Hybrid clustering and canonical correlation analysis have been used to find arc states and input flows that govern the fastest paths on the network. During the study it was found that once a car arrives at the entrance of the network then the input flows do not play a significant role and it is mainly the arc states that regulate the fastest paths on the network. There are certain arcs called critical arcs whose ranks decide the fastest paths on the network. These critical arcs can be used to forecast the paths to follow while arriving at the entrances of the network. Before entering the network, the ranks of the critical arcs inside the network are checked for expected ranks. If agreement is found, then the next path to follow is the path obtained from statistical analysis otherwise the shortest path is chosen as the fastest path.

Real networks are considerably huge in size and the present approach can be applied to networks of relatively small dimension. The next step of our study is to develop an algorithm for approximating fastest paths on real networks. This would be done by decomposing the real network into small sub-networks and computing the fastest path for the sub-networks using the statistical approach discussed in the paper. The fastest path between any origin destination pair of the real network will be obtained by joining the average fastest paths of the sub-networks.

References

[1] Ambroise, C., Badran F., Thiria S., and Sèze G. (2000), *"Hierarchical Clustering of Self-Organizing Maps for Cloud Classification"*, Neurocomputing, 30 (1-4), pp. 47-52,.

[2] CISIA (1997), *"SPAD Reference Manuals"*, Centre International de Statistique et d'Informatique Appliquées, France 1997.

[3] Cutting, D., Karger, D., Pedersen, J., and Tukey, J. W. (1992), *"Scatter/Gather: a cluster based approach to browsing large document collections"*, In Proc. 15th Annual Int. ACM SIGIR Conf., Copenhagen.

[4] Diday, E., Govaert, G., Lechevallier, Y. and Sidi, J. (1978), *"Clustering in pattern recognition"*, in 4th International Joint Conference on Pattern Recognition, Kyoto, Japan.

[5] Diday E. and Simon J.C. (1976), *"Clustering Analysis"*,. In : Fu, K. S. (Eds.): Digital Pattern Recognition. Springer-Verlag, Heidelberg, 47-94.

[6] Dougherty M.S. (1995), *" A review of neural networks applied to transport"*, Transportation Research, 3 C(4), pp 247-260.

[7] Everitt B.S. (1974), *"Cluster Analysis"*, Heinemann Educational Books.

[8] Faghri A., Hua J. (1992), *"Evaluation of artificial neural network applications in transportation engineering"*, Transportation Research Record 1358, National Research Council., Washington D.C.

[9] Jain A. K. , Murty M. N. and Flynn P. J. (1999), *"Data Clustering: A Review"*, ACM Computing Surveys, Vol. 31, No. 3, September.

[10] Kisgyörgy L., Rillett L. R. (2002), *"Travel Time prediction by advanced neural network"*, Periodica Polytechnica Ser Civil Engineering, Vol 46, No. 1, pp 15-32.

[11] Kohonen T. (1988), *"Self organization and associative memory"*, 2nd edition, Springer-Verlag, Berlin, Germany.

[12] Meila, M. and Heckerman, D.(1998), *"An experimental comparison of several clustering and initialization methods"*, In Proc. Uncertainty in Artificial Intelligence, 386-395.

[13] Murtagh F. (1995), *"Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering"*, Pattern Recognition Letters, 16, 399-408.

[14] Park D. and Rillett. L. R. (1998), *"Forecasting multiple period freeway link travel times using modular neural networks"*, Transportation Research Record,1617, National Research Council., Washington D.C.

[15] Smith R., Chou J. and Romeijn E.(1998), *"Approximating Shortest Paths in Large Scale Networks with Application to Intelligent Transportation Systems"*, INFORMS Journal on Computing, 10 , no. 2, 163–179.

[16] Ward J. H. (1963), *"Hierarchical grouping to optimize an objective function"*, Journal of the American Statistical Association, 58, 236-244.

[17] Zhan Benjamin F.(1997), *"Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures"*, Journal of Geographic Information and Decision Analysis, vol.1, no.1, pp. 70-82.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399