Decentralized algorithms for multiple path routing in urban transportation networks

Ludovica Adacher Marta Flamini Gaia Nicosia

Dipartimento di Informatica ed Automazione, University of Roma Tre,
Via della Vasca Navale 79, I-00146 Roma, Italy.
E-mail: adacher,flamini,nicosia@dia.uniroma3.it

Abstract

In the last decades, the increase of traffic and the limited capacity of urban networks, led to the development of algorithms for traffic management and route guidance. GPS technology can be used for fleet monitoring in urban or suburban areas and may provide useful information concerning the movement of all vehicles. Current route guidance systems are simple from an algorithmic point of view, since they compute shortest paths to the destination, but they have to deal with very large networks. For this reason, a decentralized approach, in which each vehicle independently calculates its own route, is desirable. Usually, the main drawback of this approach is the possibility that too many vehicles choose the same route, thus causing oversaturation phenomena. Hence, to allow path diversification, we propose a decentralized algorithm in which each vehicle computes its own satisfactory route on the basis of (i) its specific settings and (ii) traffic information provided by a reference station based on other vehicles forecasted routes.
Keywords: Shortest path, Intelligent Transportation Systems, Route Guidance.

1 Introduction

In the last decades, the increase of traffic and the limited capacity of urban networks, led to the development of algorithms for traffic management and route guidance. Studies show that an individual "blind" choice of routes leads to travel times that are between 6% and 19% longer than necessary [5]. Hence, the focus is on developing Intelligent Transportation Systems that are capable of better managing existing capacity and encouraging more efficient vehicle routing over time and space.

Many vehicles are equipped with route guidance systems (from now on, RG systems). They guide the driver from the origin to the destination by visual and acoustic indicators and allow to effectively exploit the network. In order to compute their routes, RG systems need digital maps, the current position obtained by Global Position Systems (GPS), and possibly up-to-date traffic data.

Unfortunately, many simulations predict that the benefits that can be obtained by using RG systems can be lost once the number of equipped vehicles exceeds a certain threshold. In fact, if the embedded algorithms minimize individual routes, the RG systems may cause congestion by suggesting the same path to too many drivers. Thus, RG systems must take into account the overall road usage to improve traffic management, avoiding oversaturation phenomena [1]. This can be realized by providing the RG systems with multiple path routing embedded algorithms in order to split vehicles over several paths ([13], [12], [11]). Different approaches have been proposed to deal with multiple path routing. Rilett and Van Aerde [9] suggest adding individual random error terms to the road travel times by a central controller, in order to lead each vehicle to choose different paths. Lee [6] computes k shortest paths every ten minutes and then distributes vehicles over them every two minutes, considering the current travel times on these paths.

Two other approaches consist in computing *the system optimum* and the *user equilibrium*. When the interest is in minimizing the total travel time, the system optimum is computed. Unfortunately, this policy may be penalizing for some vehicles since it may suggest unacceptable long paths, in order to route most vehicles on shorter paths. In fact, the main drawback of this approach is that some vehicles may be unsatisfied and discouraged in using the RG system. Möhring et al. [4] propose a constrained system optimum approach, where each driver is routed along a path that is not too far from its *normal length*, being the normal length of a path an appropriate measure in terms of time or distance.

The user equilibrium approach minimizes the individual journey time, routing vehicles along paths, such that no vehicle can run a quicker path through the network by unilaterally changing its choice [2].

The relation between the two approaches was investigated by Roughgarden and Tardos [10], who show that the user equilibrium approach often proposes solutions far from the minimization of the total optimum travel time of the system.

Here, we consider a hierarchical structure with two different levels: a high level, where a reference station collects all information related to the traffic on the network and a local level, represented by a set of vehicles connected to the reference station. Each connected vehicle is provided with an Intelligent Traveler Information System (ITIS), a next generation information device, capable of providing route guidance and/or traffic advice both pre-trip and while en-route [1]. An ITIS can store and process information and take into account users preferences.

We adopt a decentralized approach able to take decision based on partial information when dealing with large amount of data. We propose a multiple path routing algorithm, in which each vehicle computes its own route on the basis of (i) its specific settings reflecting user's preferences and (ii) traffic information provided by the reference station. Our aim is to propose a solution that constitutes a good tradeoff between single users satisfaction and global utilization of the network.

Regarding users satisfaction, reaching a destination by a fixed time horizon through a low risk route might be more important than driving on the shortest path. For this reason we introduce the concept of *robust path*, i.e. a route that offers alternative satisfactory paths.

The paper is organized as follows. In Section 2, we define the problem and its features and in Section 3 we describe the proposed decentralized approach and the multiple path routing algorithm based on the setting of two parameters for each user. The concept of robust path and an algorithm for computing it are presented in Section 4. Finally, in Section 5, where we briefly report on preliminary computational results and in Section 6 some conclusions are drawn.

2 Problem Definition

We are given a urban network G and a set of vehicles U, equipped with an ITIS, that have to drive in the network G. The urban traffic network can be represented by a directed graph

$G = (N, A)$, with two attributes on each arc $a \in A$: (i) the capacity $c_a > 0$, in terms of vehicles per time unit; (ii) a function t_a that denotes the arc travel time depending on the traffic on this arc x_a. The most commonly used function for arc travel times is the one used by the U.S. Bureau of Public Roads:

$$t_a(x_a) = t_a^0 \left(1 + \gamma \left(\frac{x_a}{c_a}\right)^\beta\right)$$

where $t_a^0 > 0$ is the travel time of link a in the uncongested network and $\gamma > 0$, $\beta > 0$ and $c_a > 0$ are parameters to be set like suggested in [8].

The generic vehicle $u \in U$ is associated with an origin o_u and a destination d_u, where $o_u, d_u \in N$. We want to allow each vehicle to find its own *satisfactory path* to the destination so that oversaturation phenomena are avoided. A path is satisfactory if it is close enough to the shortest path. Obviously, the definition of "closeness" is not formal and strictly depends on the user's preferences. As we will see in the next section, to satisfy customers, each vehicle computes indeed a shortest path on an individually perturbed network.

3 A decentralized solution approach

Multiple path routing algorithms should split vehicles over several paths according to each driver preferences, otherwise one may be discouraged in using the RG system. Here, we propose a decentralized approach in which each vehicle u computes its shortest path from the origin o_u to the destination d_u on the basis of global and individual parameters. The global parameters are given by the reference station and are equivalent for all vehicles that have the same origin and destination. These are the subnetwork G_u and the travel time t_a^{min} for each link a: G_u contains the nodes and links of G involved in all candidate paths from o_u to d_u; the value t_a^{min} represents a lower bound on the travel time of link a, based on the current traffic flow in the network.

The individual parameters perturb the data of the network involved in the computation of a shortest path and regard:

- how to estimate an upper bound on the travel time of each link;
- how to balance the information regarding upper bounds and globally estimated travel times of each link.

We next describe how the two individual parameters are calculated and used for the shortest path computation.

3.1 Potential flow distribution methods

The potential flow represents all users that have not calculated and communicated their individual route yet, but have already requested information on the traffic network, specifying their origin o and destination d. These vehicles will soon contribute to the traffic flow of the network and therefore an estimate of their effect is desirable. Specifically, we consider the potential flow effects, by computing an upper bound on the travel time of each link of the network that may be involved in the candidate (o-d) paths. This upper bound is also used to individually alter links travel times as described in Section 3.2.

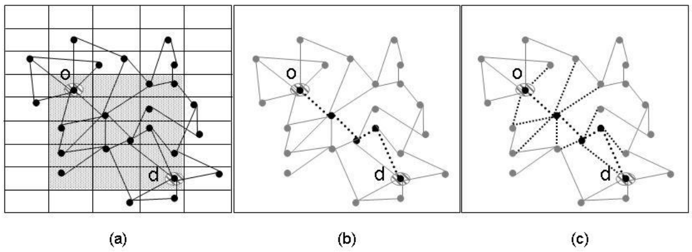

Figure 1: Potential flow computation.

Each user can choose among three different methods to perturb the network, characterized by different distributions of the estimated potential flow. The first individual parameter can be set by selecting one of the following potential flow distribution methods:

- *Potential flow in the sectors subnetwork.* We divide the subnetwork G_u in different grid sectors. The sectors are sized so that a fixed number of nodes belongs to the same sector. Each vehicle u contributes to the potential flow in the subset of sectors S_u in the rectangular area containing its origin and its destination, as represented in Figure 1(a). Hence, the travel time of each link in G is opportunely increased by considering the potential flow in every sector.

- *Potential flow on the shortest path.* Each vehicle u contributes to increase the travel time of the links of the shortest path from its origin to its destination, computed on the basis of the lower bounds on the travel times (see Figure 1(b)).

- *Potential flow on the neighborhood of the shortest path.* Each vehicle u contributes to the potential flow on the links insisting on the nodes of the shortest path from its origin to its destination, computed using lower bounds on the travel times (see Figure 1(c)).

Hence, the setting of the first parameter leads to different distributions of the potential flow on the network. Each distribution corresponds to different methods for evaluating the upper bound on the travel time of each link a, t_a^{max}.

3.2 Travel Time Computation

Here, we address how to take into account both the reference station information regarding the lower bound on the travel time of each link, and the related upper bound individually computed.

Precisely, each link a of the network is characterized by two values t_a^{min} and t_a^{max}, that respectively represent a lower bound and upper bound on the travel time of a. We propose for each vehicle u the use of a convex combination of the two bounds, with an individual parameter $\alpha_u \in [0, 1]$. So, in order to compute its most satisfactory path, each vehicle u, assumes that the travel time on link a is:

$$t_a = \alpha_u t_a^{min} + (1 - \alpha_u) t_a^{max}.$$

The parameter α_u represents the user's preferences in considering an optimistic (value of α_u close to 1) or pessimistic (value of α_u close to 0) estimate of the travel time. A very simple

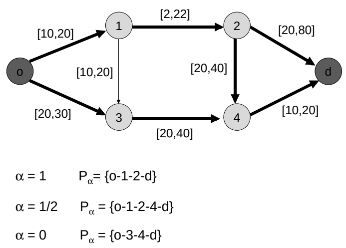

Figure 2: Example of path diversification.

example is depicted in Figure 2, where for each link a, t_a^{min} and t_a^{max} are reported, and the choice of different values for α_u leads to paths diversification.

In conclusion, as we will see in the next section, setting the individual parameters allows each vehicle to find its own *satisfactory path* to the destination so that oversaturation phenomena are avoided.

4 Robust Routes

Each user, besides choosing his individual parameters, may request a robust route, i.e. a route that offers alternative satisfactory paths for any link in case of an unexpected event. An alternative path is satisfactory if it allows the user to reach the destination in a fixed amount of time (e.g. in order to catch a flight).

In the following we formally define a $(o-d)$ robust path by relating it to the length of a shortest $(o-d)$ path P^*, in an individually perturbed network.

Definition 4.1 *An $(o-d)$ path P is a* k-robust *path, with $k \geq 1$, if for each link $(i,j) \in P$ there exist an alternative $(o-d)$ path P_{ij}, such that:*

1. *P_{ij} coincides with P from o up to i and $(i,j) \notin P_{ij}$ and*
2. *its length $\ell(P_{ij}) \leq k \cdot \ell(P^*)$.*

A user requires the route corresponding to the shortest k-robust path for a chosen k.

Obviously, a k-robust path may not exist for all $k \geq 1$ values. In the following we propose an algorithm, based on A^* algorithm, which determines a shortest k-robust path if it exists. The application of A^* is based on the possibility of giving an *estimate* e on the value of an optimal solution. It is known that while, in general, A^* is a heuristic algorithm, if the estimate e is a lower bound then it finds an optimal solution ([7]) (for a minimization problem).

Let us remind that A^* algorithm works as follows. Suppose we are given a network G, a node o and we want to find a shortest path from o to a node d representing some kind of *final* state for our search. Each arc (i,j) of G has length $t(i,j)$ and we denote by $d(i,j)$ the length of a shortest path from a node i to a node j.

The application of A^* is based on the possibility, for each node j of G, of giving an *estimate* $f(j)$ on the length of the best path through node i connecting o to d. Hereafter we describe the simplest version of A^* that considers consistent estimates ([3] and [7]). At each step, the algorithm chooses the most "promising" node (i.e. the one for which $f(j)$ is the smallest) among all the nodes which have not been yet *expanded* and expand it finding all its *successors*.

Of course, the first node we expand is node o. The process goes on, expanding at each step the node with the smallest value of f, until d becomes the most promising node.

Roughly speaking, our algorithm first determines the length of the shortest path $\ell(P^*)$ in order to compare it to the candidate alternative routes. Then it proceeds similarly to A^* ([3]), by expanding at each step the most promising node j (Step 4 in the algorithm described hereafter), after checking whether it may belong to a k-robust path (Step 5).

Here is a formal description of Algorithm kR. In the algorithm description we use the following notation.

- $t(i, j)$ denotes the length of arc (i, j).
- $w(i, j)$ denotes the length of arc (i, j) if it can belong to a $k-$robust $(o - d)$ path. This value is ∞ if the arc (i, j) can not belong to a $k-$robust $(o - d)$ path.
- $d^t(j)$ represents the minimum distance from o to j along a k-robust $(o - d)$ path, computed considering the $t-$weights.
- $d^w(j)$ represents the minimum distance from o to j along a k-robust $(o - d)$ path, computed considering the $w-$weights.
- $O^{(h)}$ is the set of expanded nodes at iteration h.
- $D^{(h)}$ denotes the set of unexpanded nodes, at iteration h..
- $Succ(O)^{(h)}$ denotes the set of nodes in $D^{(h)}$ which have a predecessor in the set $O^{(h)}$.
- $RobP(i, d, (i, j))$ is a boolean function that returns $TRUE$ only if there exist a $k-$robust $(o - d)$ path involving node i and not including arc (i, j).
- $pred(j)$ indicates the node preceding node j in a $k-$robust $(o - d)$ path.

Algorithm kR

Input: a network $G(N, A)$ weighted on arcs with weights $t(i, j)$, evaluated with individual parameters, an origin $o \in N$, a destination $d \in N$, an integer k;

Inizialization: $h = 0$; $\forall j \in N, d^t(j) \leftarrow \infty$; $d^t(o) = d^w(o) \leftarrow 0$; $pred(o) \leftarrow o$ $O(h) \leftarrow \{o\}$, $D(h) \leftarrow N \setminus \{o\}$, $\forall (i, j) \in A$ $w(i, j) \leftarrow t(i, j)$;

MainProcedure$^{(h)}$:

1. Let $Succ(O)^{(h)} = \{j \in D^{(h)} : (i, j) \in A \text{ and } i \in O^{(h)}\}$;
2. $\forall\ j \in Succ(O^{(h)})$, $pred(j) \leftarrow \arg\min_{i \in O^{(h)}} d^w(i) + w(i, j)$ and $d^w(j) \leftarrow d^w(pred(j)) + w(pred(j), j)$;
3. **if** $(\forall j \in Succ(O^{(h)}),\ d^w(j) = \infty)$
   then STOP, a robust $(o - d)$ path does not exist;
4. let $j' = \arg\min_{j \in Succ(O^{(h)})}\{d^w(j)\}$;
   $t(pred(j'), j') \leftarrow \infty$;

5. **if** $(RobP(pred(j'), d, (pred(j'), j')) = TRUE$
then
$O^{(h)} \leftarrow O^{(h)} \cup \{j'\}$;
$t(pred(j'), j') \leftarrow w(pred(j'), j')$;
$h \leftarrow h + 1$;
if $(j' = d)$
then RETURN the robust $(o - d)$ path identified by the *pred* vector
else repeat MainProcedure$^{(h)}$;
else
$t(pred(j'), j') \leftarrow w(pred(j'), j')$;
$w(pred(j'), j') \leftarrow \infty$;
$pred(j') \leftarrow NULL$;
repeat MainProcedure$^{(h)}$

5 Computational Results

In this section we shortly describe our computational experience. Tests were performed on randomly generated instances considering different network topologies (characterized by density of nodes, connectivity, and distribution of the capacity on the links) with different flow values. More precisely, we have considered graphs with a number of nodes varying in $[331, 526]$ and arcs in the range $[2520, 7904]$. There are three link types depending on the capacity: small, medium and large. The connectivity of the graph decreases from the central area to the outer part.

The flow value on each arc a is randomly generated in interval $[1, c_a/2]$. In Table 1 the characteristics of the instances and the related results are reported. All tests consider one origin-destination pair $(o - d)$. Columns $1 - 2$ refer to the number of nodes and arc of the considered graph, while column 3 indicates the potential flow considered. The value PF1 corresponds to a potential flow value of 100 between o and d, while PF2 includes PF1 plus a flow value of 70 between three origin-destination pairs close to the $(o - d)$ shortest path in the uncongested network. Finally, PF3 includes PF2 plus a flow value of 70 between two randomly picked origin-destination pairs. Columns $4 - 7$ report the relevant results. In particular, the computation time expressed in seconds is in column 4; in column 5 the number of different paths obtained by combining 5 different α_u values ($\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$) with the three potential flow evaluation methods is reported; columns $6 - 7$ indicate the minimum and maximum percentage *difference*[1] between each pair of computed different $(o - d)$ paths.

The preliminary results show that vehicles are indeed routed on different paths, even though path diversification is limited with respect to all tested possible combinations of the two parameters. In fact the number of different paths varies between 3 and 6 and the total number of parameter setting combinations is 15. This is not necessarily a negative result, since an excessive path diversification might route some users on unacceptable long paths.

A deeper analysis of the results shows that given a fixed alpha value the number of different paths found is 2 or 3 out of 3 (corresponding to the 3 parameter values regarding potential flow distribution method). On the other hand, given a potential flow distribution method, the

[1]Given two $(o-d)$ paths P_1 and P_2 a measure of their difference is computed by determining the number of different arcs in P_1 and P_2, i.e. the cardinality of the symmetric difference between the arcs of the two paths, divided by the total number of arcs.

n	m	potential flow type	t	different paths	min % difference	max % difference
331	3370	PF1	1	5	18	90
331	3370	PF2	1	4	10	95
331	3370	PF3	1	3	10	100
318	3079	PF1	1	3	100	100
318	3079	PF2	1	4	20	100
318	3079	PF3	1	4	67	100
307	2087	PF1	2	3	90	90
307	2087	PF2	2	6	20	100
307	2087	PF3	2	6	67	80
281	2520	PF1	3	4	15	85
281	2520	PF2	3	4	37	50
281	2520	PF3	3	4	20	62
355	4000	PF1	3	4	50	67
355	4000	PF2	3	6	36	66
355	4000	PF3	3	6	33	83
505	7512	PF1	3	4	80	90
505	7512	PF2	3	3	95	95
505	7512	PF3	3	4	32	100
526	7904	PF1	3	4	35	83
526	7904	PF2	3	4	45	75
526	7904	PF3	3	4	33	45
498	7348	PF1	3	5	40	56
498	7348	PF2	3	6	50	52
498	7348	PF3	3	5	46	50
518	7836	PF1	3	3	50	100
518	7836	PF2	3	3	71	100
518	7836	PF3	3	4	38	100
474	6521	PF1	3	3	20	100
474	6521	PF2	3	4	95	100
474	6521	PF3	3	4	46	95

Table 1: Computational Results

number of different paths found is at most 3 out of 5. Therefore the potential flow distribution method seems to be more effective in finding multiple paths.

Finally, observe that the results strictly depend on the specific instance. Hence, more tests are required in order to clearly point out the effectiveness of the proposed multiple path routing strategy, in terms of relation with network topology and/or potential flow value and distribution.

6 Conclusions

In this work we proposed a decentralized multiple path routing algorithm for the route-guidance problem. We described a simple approach able to find a good tradeoff between single users satisfaction and global utilization of the network. We are at a preliminary stage of our study, however the first computational results are encouraging. Further experiments will deal with a more accurate study of the effectiveness of parameters settings with respect to network topology and flow distribution and with the additional option of requesting a satisfactory and robust path.

References

[1] Adler J. L., Blue V. J., 1998. Toward the design of intelligent traveller information systems. *Transportation Research C, 6:157-172.*

[2] Fresz T. L., 1985. Transportation network equilibrium design and aggregation: key development and research opportunities. *Transportation Research A., 19:413-427.*

[3] P. E.Hart, N. J. Nilsson, B. Raphael, 1968. A formal basis for he heuristic determination of minimum cost paths, IEEE Trans. Syst. and Cybern. SSC-4, 100-108.

[4] Jahn O., Mhring R. H., Schulz A. S., Stier Moses N. E., 2002. System-Optimal Routing of Traffic Flows with User Constraints in Networks with Congestion, Technical Report 754/2002 Technische Universitt Berlin.

[5] Jahn O., Mhring R. H., Schulz A. S., 1999. Optimal Routing of Traffic flows with length restrictions in networks with congestion. Technical Report 658/1999 Technische Universitt Berlin.

[6] Lee C. K., 1994. A multiple-path routing strategy for route guidance systems. *Transportation research C, 2:185-195.*

[7] Nilsson N. J., 1971. Problem solving Methods in Artificial Intelligence, McGraw-Hill, New York, 1971.

[8] Y. Sheff., 1985. Urban Transportation Networks. Prentice-Hall, New Jersey.

[9] Rilett L.R., van Aerde M. W., 1991. Modeling distributed real-time guidance strategies in a traffic network that exhibits the Braess paradox. In proceedings of IEEE Vehicle Navigation and Information Systems Conference.

[10] Roughgarden T., Tardos E., 2000. How bad is selfish routing?. In *Proceedings of the 41st annual IEEE, symposium on foundation of computer science FOX'00.*

[11] I. Kaysi, M.E. Ben-Akiva, and A. de Palma. Design aspects of advanced traveler information systems. In *N.H. Gartner and G. Improta, editors. Urban Traffic Networks. Dynamic Flow Mod- elling and Control. Springer, Berlin, 1995., pages 59-81.*

[12] J.J. Henry, C. Charbonnier, and J.L. Farges. Route guidance, individual. In *M. Papageorgiou, editor. Concise Encyclopedia of Traffic and Transportation Systems. Pergamon Press, Oxford, 1991.*, pages 417-422.

[13] G. Beccaria and A. Bolelli, 1992. Modelling and assessment of dynamic route guidance: the MARGOT project. In *Proceedings of the IEEE Vehicle Navigation and Information Systems Conference, Oslo, 117-126.*