Traffic Routes for Emergency Services

Marek Małowidzki, Michał Mazur, Tomasz Dalecki, Przemysław Bereziński
Military Communication Institute
Zegrze, Poland
{m.malowidzki,m.mazur,t.dalecki,p.bereziński}@wil.waw.pl

Abstract—One of the key functions related to traffic optimization and control in Insigma is computing routes for emergency services. In Poland, due to its special status, a privileged car behaves in a substantially different way than a plain car. As a result, “privileged” routes should also considerably differ. In the paper, we propose a solution that combines road data and a behavior model, able to determine such routes. A number of presented case studies confirms our solution works and gives reasonable results—routes that are faster and more flexible than routes for unprivileged vehicles.

Keywords: road traffic, routing, emergency services, privileged vehicles, Open Street Map (OSM).

I. INTRODUCTION

The Insigma project is aiming at the development of an intelligent information system for global monitoring, detection and identification of threats. The system collects data from various kinds of sensors, cameras, and users, and processes the data to identify threats and notify appropriate public services. One of Insigma’s tasks is road traffic optimization and control, which includes traffic lights, information boards, and route planning.

The paper concerns one of the key functions of the Insigma’s traffic subsystem (see Figure 1.): computing routes for emergency services (the police, the fire brigade, the ambulance service, etc.). This function stems from the specific application of Insigma and its relation to public security. A “privileged” route may be used by the police to reach the place of a crime or select a hospital an ambulance will be able to reach in the shortest possible time.

It is assumed that such routes will be significantly different from routes for plain vehicles, as they should take into account the special status of privileged vehicles. In the paper, we discuss the design and implementation of this functionality in our routing service [15].

The paper is organized as follows: First, review related work. Then, we overview the traffic subsystem in Insigma and the routing service architecture and functions. We also discuss design assumptions and the implementation of privileged routes. Next, we evaluate our solution using a set of case studies. We also comment on limitations. Finally, we propose the areas of future work. We end the paper with conclusions.

II. RELATED WORK

Our research discovered multiple publications related to “routes for emergency services,” unfortunately, very few of them considered the unique nature of a privileged vehicle. A potential cause could be different rules of the road (than in Poland, see section V), equal for emergency services and plain road users.

Mali et al. [10] propose a priority matrix based on the AHP (Analytical Hierarchical Processing) method. Priorities take account of various features of roads (length, width, traffic volume, etc.). An enhanced variant of Dijkstra’s algorithm allows to find optimal routes. It is not clear how exactly the matrices are constructed (as a matrix is required for every road segment, in practice, some automation would be required). In our approach, presented further in the paper, a standard routing algorithm is employed, while the work related to handling privileged routes (or, another specific route type) is delegated to a dedicated adapter. Also, our routing service uses dynamic traffic data [17] and traffic prediction, as opposed to static matrices prepared in advance.

Westgate et al. [14] propose a method of travel time estimation for ambulances using Bayesian data augmentation. The estimation is made for individual roads and takes into account possible data errors. We believe that such a method, if correlated with the traffic situation (at the time the ambulance speed data have been collected) and the road facilities, could be a basis for a more sophisticated speed model of a privileged vehicle, or, at least, it could serve as a reference for model validation. The problem stems with large amount of dynamic data that need to be taken into account and the complexity of the resulting theory.

Wang and Liu [13] study the application of Internet of Things (IoT) for computing routes for ambulances. RFID and wireless sensors are used to collect data about traffic jams. Then, jammed ways are removed from the graph, which allows to find routes that bypass them. We think that this approach is simpler than ours, as we take into consideration additional features (drive against the stream, roadsides, etc.) that may in fact enable a relatively smooth drive through jammed areas.

There are also a number of interesting sources that relate to emergency services, although finding routes is not the primary goal. Buchenschicht et al. [12] discuss an emergency warning system that both warns other drivers and cooperates with traffic lights system (to assure a green light at crossroads). A similar project, involving simulations in SUMO, is described in [11].
III. THE INSIGMA’S TRAFFIC SUBSYSTEM

The Insigma’s ultimate (and somewhat ambitious) vision is presented in Figure 1. Insigma’s goals with respect to road traffic include traffic control, traffic prediction, route planning, and related functions. Despite the fact that ongoing work includes real-world equipment (advanced cameras located at crossroads, collecting detailed data about observed traffic license plate recognition software [4], etc.), the only way to verify control algorithms is simulation. (Drivers would not be pleased finding themselves to be beta testers of our ideas.) Thus, we are working [16] on the SUMO road traffic simulator [5] in order to integrate it with the control layer.

The control layer consists of a number of components. In this paper, we focus on the routing service. There are a number of such services commercially available and successful on the market (with Google Maps [9] as a premier example), however, specific functions that the service was to support as well as planned integration with higher-level traffic control algorithms in practice required implementing a new one from scratch.

In the following section, we discuss the routing service’s architecture and key ideas.

IV. THE ROUTING SERVICE

The routing service’s internal architecture is shown in Figure 2. The architecture has been discussed in detail in our previous work [15] but, for the completeness of the discussion, we briefly summarize it here.

The main elements (components) are as follows:

- Input/output, a component responsible for handling messages for clients, providing optional QoS and security functions;
- Database, containing the static map and dynamic data (traffic statistics: drive and turn times, speeds, etc.). At present, the static map contains the Open Street Map (OSM, [6]) data, although the database format has been significantly modified for our purposes (refer to section VII for details);
- Graph Builder, responsible for transforming map data into a graph (a set of nodes and edges) that may be used for route computations;
- Adapter(s), computing graph weights. Usually, each route type (Fast, Short, Optimal, etc.) requires a separate adapter. Their implementation may be trivial (e.g., for a Short route) or fairly complex, as it is in case of a Privileged Route adapter, described in section VI. Note that we assume that weights are functions of time (the start time of the drive at a given edge) [15], which allows to take into account dynamic (current and predicted) traffic data [17].
- Algorithm(s), performing route optimizations. Algorithms are separated from road data by adapters; they only see the graph with edge weights computed by adapters. We have tested a number of algorithms, including Ant Colony Optimization [1] and an adapted version of SAMCRA [2], but found that Dijkstra-based algorithms (an optimized version using a priority queue or the A* algorithm [3]), perform best.
- Finally, the Dispatcher, managing the above-mentioned elements, and providing additional functions (e.g., alternative routes) and debugging capabilities.

The software is implemented in the Microsoft .NET 4 framework environment. The internal architecture is organized around a number of interfaces. Most crucial elements (.NET classes implementing well-known interfaces) that affect the service behavior (the graph builder, adapters, algorithms) are dynamically loaded according to the server’s configuration.
We have also developed a client, implemented in JavaScript/OpenLayers [8] environment. The client’s capabilities allow to make use of most of the routing service’s functions. All the figures showing computed routes on a map, presented further in the paper, have been prepared using our client.

V. EMERGENCY SERVICES AND THE RULES OF THE ROAD

In Poland, a privileged vehicle (with a siren going; often referred to as a PV further in the paper) does not have to obey the rules of the road, as it always has the right of way. This means such a vehicle may:

- drive faster than allowed;
- ignore traffic lights;
- take prohibited turns on a crossroad;
- use parts of the road normally excluded from traffic (road sides, no traffic areas);
- turn off the road and use sidewalk or take a “shortcut” across a square or a park;
- or, finally, go against the stream.

Of course, such a behavior is generally dangerous and should be applied with care (e.g., the authors of the paper have never seen the drive against the stream), nonetheless, the routing service should be aware of (and apply in practice) the special status of a privileged vehicle.

VI. PRIVILEGED VEHICLE BEHAVIOR MODEL

“Privileged” routes are computed by a dedicated adapter. A privileged route will be returned by the service only if a client’s request requires a Fast and privileged route, and a vehicle profile specifies a privileged vehicle. The adapter employs a behavior model, which in turns builds upon a speed model. Both the behavior and speed models use detailed information about a road segment, summarized in the form of a friendliness factor. The implementation is based upon a number of constants that are loaded from a configuration file. These constants may be adjusted to model.

Friendliness factor. The factor takes values from 0 to 1 and specifies how much “friendly” a road segment (and a corresponding graph edge) is for a PV. The value of 0 means a narrow road with obstacles on both sides; in such a case, a PV can only move as fast as other vehicles. The value of 1 denotes a separate lane for emergency services (extremely rare in Poland), which enables collision-safe drive at a high speed. Intermediate values affect the estimated top speed value a PV could develop.

The friendliness factor takes into account the following data about the road and its sides: separate lanes, road sides and road signs, bus and tram lanes (and availability hours for bus lanes), road width (the number of lanes), etc. These data are not present in OSM, so we have developed additional tools to supplement the database (see section VII).

Speed model. For plain cars, we measure drive and turn times (or, not having such data, use some heuristics). However, a PV never stops at a crossroad. It only has to slow down when entering a crossroad, especially on a red light or when it is going to take a forbidden turn. In order to include the slowdown in our model, we assume a PV develops some (relatively small) speed at a crossroad, then linearly accelerates to its top value (computed on the basis of a friendliness factor value and a speed limit), drives at the top speed, then linearly slows down at the next crossroad (for a typical car, braking is faster than acceleration, but we ignore the difference). In case a road segment is short, the top speed is not reached.

Note that while this idea is (statistically) reasonable during high traffic hours, it is generally incorrect otherwise (e.g., late at night). Thus, a more complex model should take the time and road conditions into account as well.

Behavior model. The behavior model combines the above ideas and additional information (accessibility – whether the drive is normally allowed or prohibited) to compute the drive time along a road segment. The implementation is based on heuristics (that stems from some observed behavior and some “sane” assumptions). We do not describe it in details here; instead, we list the demands it fulfills:

- a PV goes faster that a plain car;
- its speed grows with friendliness factor value;
- even during a “total” traffic jam, a PV goes forward unless friendliness is 0;
- drive against the stream also depends on friendliness and is slow; there is additional penalty involved to reduce this dangerous behavior;
- the top speed inside a no traffic zone is limited; the speed model remains valid.

VII. DATABASE AND GRAPH BUILDER

Our data model is based on OSM but introduces some essential differences. The road model is based on way segments, which connect two crossroads. At present, a crossroad is merely a collection of turns. A way segment contains two lists of lanes (each for a single direction of traffic). Way segments are connected by turns.

For unidirectional way segments, a lane in a reverse direction is labeled as forbidden. Similarly, feasible in practice but administratively prohibited turns are marked as forbidden. This “forbidden” status is respected for plain cars; for a PV, however, the forbidden road elements are included in the graph (compare Figure 6. and Figure 7.).

Due to differences between data models we cannot use OSM software [7] to prepare map data. Instead, we have developed a dedicated tool, InismaDatabaseTool. The tool enables us to supply the road data with additional information, required for privileged routes (refer to Figure 3, below and the discussion about the friendliness factor in section VI).

VIII. CASE STUDY: THE SPEED MODEL

In this and subsequent sections we demonstrate a number of case studies. We show the best routes, although a driver may ask the service to compute alternatives. At present, there are no options to affect privileged routes in a
detailed manner (enable or disable some features, e.g., the drive against the stream) but it would be relatively easy to implement, as we already have a flexible interface for supplying route options.

We show an example in Figure 4. The computed route selects major roads and is the same for both vehicle types. However, additional facilities we define for the roads (which affect the friendliness factor: three lanes, a road side, a separate lane) yield different estimated drive times (Table I). As expected, estimated time drops with (almost) each subsequent table row, as each subsequent feature maps to an increased friendliness factor value.

TABLE I. ROUTE COMPARISON

<table>
<thead>
<tr>
<th>Feature</th>
<th>Time [s]</th>
<th>Length [km]</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (plain car case)</td>
<td>461</td>
<td>5.2</td>
</tr>
<tr>
<td>Wide road (three lanes)</td>
<td>434</td>
<td>5.2</td>
</tr>
<tr>
<td>Road side with “no parking” road signs</td>
<td>370</td>
<td>5.2</td>
</tr>
<tr>
<td>Separate lane for emergency services</td>
<td>370</td>
<td>5.2</td>
</tr>
</tbody>
</table>

IX. CASE STUDY: TRAFFIC JAM

This case study is a variant of the above one. This time, the “obvious” best route, composed of major roads, is jammed (low observed drive speeds are set in the dynamic map). Thus, for a plain car, the routing service recommends a detour (Figure 5.). However, despite the traffic jam, the main roads exhibit a big friendliness value due to road sides (and no parking road signs along them); as a result, a PV is directed to go down the jammed roads. The routing service estimates the PV route to take 358 s and be 4.2 km long (the route uses the main road, as in Figure 4.; the route is just shorter as the destination is closer), as opposed to 478 s and 5.7 km for a plain vehicle route.
X. CASE STUDY: NO TRAFFIC ZONES

The large part of the center of Cracow (mainly, the historic Old Town), is excluded from traffic (except for local residents and some town and logistic services). However, a PV should be able to use such areas when justified. According to our model, a PV speed in such areas is limited, as we expect people walking freely and not expecting any danger.

Indeed, as we show in Figure 6. and Figure 7., a PV takes an advantage and uses a no traffic zone as a shortcut, reducing the drive time from 204 s to 121 s, and the route length from 1.6 km to 1.2 km. The difference could be even bigger if there were traffic jams simulated on the public roads. Also note that the graph (presented using blue lines) in the PV case contains visibly more edges.

XI. CASE STUDY: REVERSE DIRECTION

The next case study verifies whether a privileged vehicle could go against the stream. Recall that, according to our model, a PV goes very slowly in a reverse direction. Moreover, we apply an additional penalty (currently, 10 times the drive time against the stream), which effectively “discourages” the routing service from including such segments in a route. Nonetheless, sometimes a gain is so significant that it should outweigh the penalty anyway.

An example of a drive against the stream is presented in Figure 8. (The red color denotes segments of a drive in a reverse direction.) The privileged route (Figure 8.) is 0.9 km long, as compared to 7.7 km and 430 s for a plain route (Figure 9.).

![Figure 6. Drive through a no-traffic zone (the green color)](image1)

![Figure 7. Corresponding route for a plain vehicle](image2)

![Figure 8. Drive against the stream](image3)

![Figure 9. A corresponding route for a plain vehicle](image4)
XII. LIMITATIONS

In the real world, a PV driver is able to take a shortcut across an area that has not been designed for traffic (e.g., a lawn, a park, etc.). Theoretically, our service could include such shortcuts but they would have to be defined in advance in the database. Note, however, that this task could be simplified (and automated to some degree by specialized software) by tracking PVs and identifying ways that are not present in the static map, then supplementing the map with these ways marked as “PV shortcuts.”

XIII. FUTURE WORK

We can see four main areas for a future work. The first one is related with our model validation: Having empirical (real world) data about privileged vehicles behavior, we could tune our model to be closer to the observed behavior.

The second area is related with simulation experiments (Figure 1.; also refer to [16] for a description of our work in this field) involving privileged vehicles. One could combine a dedicated route, computed as described in our paper, with traffic lights control in order to establish a non-blocking route (a green light on every crossroad) for a privileged vehicle, which would further shorten its drive time to destination. (This would of course invalidate our assumption of a PV slowing down at each crossroad, so our model should be altered.) A similar idea is explored in [11], [12].

The scope of Insigma encompasses another research issue: routes for special vehicles. A special vehicle is assumed to not only have unusual dimensions or height but carry unusual (and potentially dangerous) load as well. Thus, it is not sufficient for such routes to be feasible; they also should be safe and put as little disturbance to normal road traffic as possible.

Future work could also involve support for some special services, such as the one provided by the R² Foundation in Cracow. The foundation [18] is a social initiative aiming at delivering basic ambulance service. Volunteer rescuers employ bikes or motorbikes to reach the wounded person as quickly as possible and give first aid (some of motorbikes are even equipped with defibrillators). Note that supporting single-track vehicles would require a modified speed model (such a vehicle could always go through no matter how much a way is jammed) and a different approach to route computation (as it is very easy for such a vehicle to take off the road shortcuts).

XIV. CONCLUSIONS

The problem of computing privileged routes is complex and involves a lot of issues. It is significantly different from computing plain routes. Plain routes use “normal” graphs constructed from map data and employ dynamic values (speed values, turn times) observed in practice by sensors [17]. Thus, the main problem is computing these values and, perhaps, predicting traffic in future, as it may affect lengthy routes. In case of privileged vehicles that have special status (as in Poland), additional, detailed data about roads and their sides are required, and an extended graph should be formed. Moreover, a PV’s behavior must be modeled. We believe that this research area is mostly unexplored.

We have proposed and implemented a model of behavior of a privileged vehicle. While our model is heuristic, our case studies confirm it is sound and key behavior elements, specific for such a vehicle type, are sustained. Thus, our routing service is able to compute routes for emergency services that are significantly different (i.e., faster) than corresponding routes for plain vehicles.

ACKNOWLEDGMENT

The work has been co-financed by the European Regional Development Fund under the Innovative Economy Operational Program, INSIGMA project no. POIG.01.01.02-00-062/09.

REFERENCES