

# Generation of Realistic Mobility for VANETs Using Genetic Algorithms

Marcin Sereczynski\*, Grégoire Danoy†, Masoud Tabatabaei\*, Pascal Bouvry† and Yoann Pigné‡

\* Interdisciplinary Centre for Security, Reliability and Trust  
University of Luxembourg  
Luxembourg

Email: {marcin.sereczynski, masoud.tabatabaei}@uni.lu

† CSC Research Unit, University of Luxembourg  
Luxembourg, Luxembourg

Email: {gregoire.danoy, pascal.bouvry}@uni.lu

‡ LITIS - University of Havre  
Le Havre, France

Email: yoann.pigne@univ-lehavre.fr

**Abstract**—The first step in the evaluation of vehicular ad hoc networks (VANETs) applications is based on simulations. The quality of those simulations not only depends on the accuracy of the network model but also on the degree of reality of the underlying mobility model. VehILux—a recently proposed vehicular mobility model, allows generating realistic mobility traces using traffic volume count data. It is based on the concept of probabilistic attraction points. However, this model does not address the question of how to select the best values of the probabilities associated with the points. Moreover, these values depend on the problem instance (i.e. geographical region). In this article we demonstrate how genetic algorithms (GAs) can be used to discover these probabilities. Our approach combined together with VehILux and a traffic simulator allows to generate realistic vehicular mobility traces for any region, for which traffic volume counts are available. The process of the discovery of the probabilities is represented as an optimisation problem. Three GAs—generational GA, steady-state GA, and cellular GA—are compared. Computational experiments demonstrate that using basic evolutionary heuristics for optimising VehILux parameters on a given problem instance permits to improve the model realism. However, in some cases, the results significantly deviate from real traffic count data. This is due to the route generation method of the VehILux model, which does not take into account specific behaviour of drivers in rush hours.

**Index Terms**—Vehicular ad hoc networks; genetic algorithms; intelligent transportation systems; realistic vehicular mobility models; traffic simulation; mobility traces;

## I. INTRODUCTION

The application of wireless technology to moving vehicles enables the creation of one particular class of mobile ad hoc networks called *vehicular ad hoc networks* (VANETs). The use of such networks allows to create *intelligent transportation systems* (ITS), which hold huge benefits for road safety, traveller comfort and traffic efficiency. The concept of ITS encompasses three main groups of applications [1]: (i) *safety-related*, (ii) *infotainment and advanced driver assistance services* and (iii) *resource efficiency (traffic, environment)*. Developing applications and protocols requires experimental testbeds and simulation tools for their performance evalua-

tion. Testbeds allow to obtain performance measures under real-world conditions such as realistic signal propagation and vehicles' mobility patterns. However, due to several constraints such as costs and reproducibility of real testbeds, simulation is one of the most often used methods for performance evaluation [2]. It requires both the development of realistic (i) models for the signals' physical characteristics (see e.g. [3], [4]) and (ii) mobility of vehicles at a macroscopic and a microscopic level. This work focuses on the second aspect—*mobility models*. Such models represent the movement of vehicles by providing information on how their location, velocity and acceleration change over time. Mobility models are used to generate *mobility traces*, which can be used for the simulation of vehicle-to-vehicle or vehicle-to-infrastructure communication and the evaluation of related applications. It has been shown in the literature that the quality of the evaluation of VANETs applications strongly depends on the applied mobility models [5]. The recent trend in the evaluation process is to use vehicular traces as an input to a network simulators (e.g. ns-3 [6]). Traces can in general be classified as either *real-world* or *artificial* (synthetic) [7]. The former are typically obtained from GPS-based tracking of road transportation (e.g. busses, taxis). The latter are obtained using microscopic and macroscopic road traffic simulation. Using real-world traces would be the most desirable, however, they exist for very few cases only. Consequently, methods for generating realistic artificial traces are needed. An attempt to generate traces using real-world traffic information and traffic simulation is proposed in [8]: a model called *VehILux* uses two inputs describing real-world—a *geographical map* and a set of *traffic volume counts* corresponding to the area covered by the map. A traffic volume count is a count of traffic along a particular road. It provides precise information about the number of vehicles and their type [9]. In order to generate traces out of localised traffic count data, the model relies on probabilistic geographical attraction points used to select the destination of each vehicle. However, the VehILux model does not address

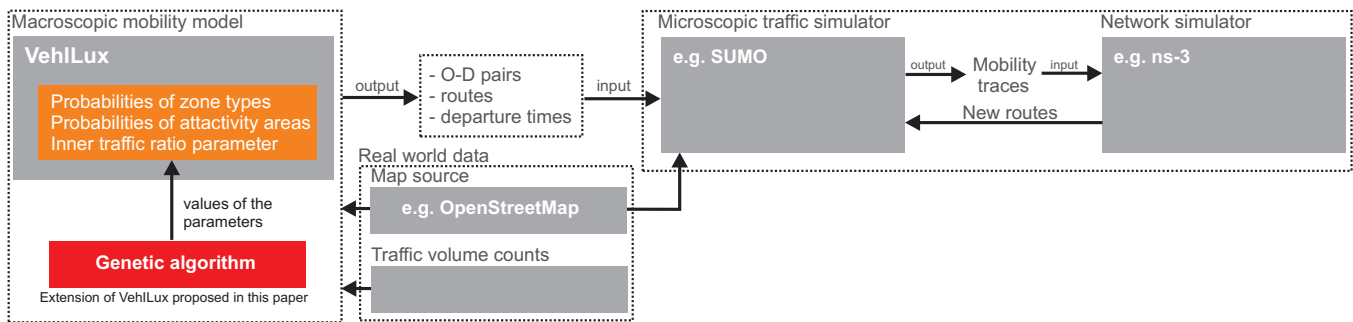


Fig. 1. Generation of realistic vehicular traces and bidirectional coupling between traffic and network simulators. VehILux model does not address the issue of how to select the values of the probabilities of selection of zones, attractivity areas and inner traffic ratio.

the question of how to select the values of these probabilities. These values depend on the actual problem instance (typically specified by given geographical region and corresponding traffic volume counts). Therefore, for every problem instance, they need to be discovered. In this paper we demonstrate how this task can be addressed using *genetic algorithms* (GAs). Given the particular set of data volume counts and the corresponding map of the area, an optimisation problem is defined. Its objective is to find the parameters of VehILux fitting the best to the problem instance. Using computational experiments and real traffic volume counts from Luxembourg we demonstrate how the proposed approach based on GAs can generate parameter values of VehILux leading to realistic vehicular traces.

This paper is structured as follows. The next section reviews the relevant developments in realistic mobility modelling. Section III defines the problem tackled in this work. The main contribution of this paper—the evolutionary approach for discovery of parameters values in the VehILux model—is then introduced in Section IV. Section V describes the experimental design and contains simulation results and analysis. Finally, section VI provides our conclusion and perspectives.

## II. RELATED WORK

The use of traffic and network simulators capturing realistic environment allows to evaluate VANET-based applications. Movement traces produced by a mobility simulator are used as an input to network simulator. Such an approach allows to analyse the influence of road traffic on network traffic. However, if the aim is to analyse the influence of a VANET-based application (e.g. cooperative traffic information system based on VANETs [10]) on the road traffic, then bidirectional coupling between the two simulators is required. Examples of such online interaction between traffic and networks simulators can be found in [11], [12], [13].

Vehicular traffic analysis can in general be classified into microscopic and macroscopic [14]. Macroscopic approach considers gross quantities of interest (e.g. traffic density, traffic flow, and velocity of vehicles) viewing traffic according to fluid dynamics. Microscopic analysis considers the movement of each vehicle in the road traffic, thus models traffic behaviour in a more precise way.

Three classes of mobility models can be distinguished: *trace-based*, *survey-based* and *traffic simulator-based* [15]. In the models belonging to the first class, mobility patterns are derived from real-world mobility traces. Such traces typically correspond to commercial vehicles like public busses (see e.g. [16]) or taxis (see e.g. [17]). A collection of datasets of mobility traces from various origins can be found in [18]. There are two problems with trace-based approaches. Firstly, the availability of the traces is limited. Secondly, traces are typically gathered for one particular type of transportation, therefore, observable patterns are limited (e.g. in [16] traces are gathered from 35 buses). In the models belonging to the second class (survey-based), mobility patterns are derived from traffic statistics. Examples include [19], [20]. As these models provide macroscopic mobility information only, their application for the evaluation of VANET-based applications is limited. The final class of mobility models relies on microscopic traffic simulators. It determines the movement of each vehicle in the road traffic. It can realistically simulate interactions between vehicles and road infrastructure. In order to obtain realistic mobility at macroscopic and microscopic levels, trace or survey-based information needs to be used in conjunction with microscopic traffic simulation. The *VehILux* mobility model introduced in [8] is an example of such an approach (see Fig. 1). It uses two sources of data as inputs describing real-world—a geographical map and a set of traffic volume counts corresponding to the area covered by the map. The advantage of the model is that it uses real-world data, which consider all road traffic and is widely available (e.g. [21], [22], [23]). The output of VehILux consists of three elements, independently defined for each time slot (typically one hour slots are used): (i) the estimation of the number of vehicles circulating in the region, (ii) a prediction of the origin-destination pairs for vehicles, and (iii) a set of a routes for all pairs. Such information in conjunction with microscopic traffic simulation allows to generate realistic mobility traces for any area for which traffic volume counts exist. Generation of origins and destinations as well as the routes is based on the concepts of *repulsion* and *attraction* points. The former constitute typical origin points on a map, while the latter represent popular final destinations. VehILux uses a probabilistic model for the

selection of attraction points—each of them is assigned with a probability of being chosen as a destination. However, the model not address the question of how to select the values of these probabilities and the value of the inner traffic ratio.

### III. PROBLEM DEFINITION

This section first explains relevant parts of the VehILux model and then it defines the related optimisation problem.

#### A. VehILux model

The VehILux model generates a set of origin-destination pairs and a route for each pair. As an input it uses a detailed *map* of the concerned region and hourly *traffic volume counts* for a set of locations in the region. A single traffic volume count represents the number of vehicles that passed through the count in a given time interval (hereafter referred to as a *time slot*). An example of a traffic volume count for a single location is shown in Fig. 2a.

The available traffic volume count locations are divided into two sets. The first set consists of *entering points*, while the second set is composed of *control points*. The former are used as trip origins, while the purpose of the latter is to validate the quality of the generated traffic flows.

In the example shown in Fig. 2b the entering points are selected from the locations positioned at the edges of the map, as this problem instance is concerned with commuting hours. Count locations from the second set—control points—are positioned inside the area covered by the map.

Entering points and corresponding volume counts are used as to initiate the flows of vehicles. For instance, in the first time slot, around 1200 vehicles will start a trip at count location number 1410. The traffic generated from these points is referred to as *outer traffic*. Naturally, some portion of traffic is also generated from the inside of the map. This traffic is referred to as *inner traffic*. It is proportional to the outer traffic, i.e. its volume is derived from outer traffic using a parameter called *inner traffic ratio*. The origins of the trips that start as inner traffic are chosen from residential zones (the concept of zones is explained in the next section). As residential zones might exist in several locations, the probability of selecting a specific location (as the origin) is positively correlated with its surface.

The process of selecting destinations and routes for each vehicle is described in the next section.

#### B. Route assignment: destination distribution model

Route assignment concerns the selection of a destination for each origin and the selection of a route between the origin-destination pair. The selection of a destination is based on the notion of *zone types* and *attractivity areas*. The following zone types differing in the offered service are defined—*commercial*, *residential* and *industrial*. Each of them is assigned with a probability of being selected as a destination type. However, zones of the same type can exist in several locations spread over the map. Moreover, one location can be more popular than the other, hence, it should be associated with a higher

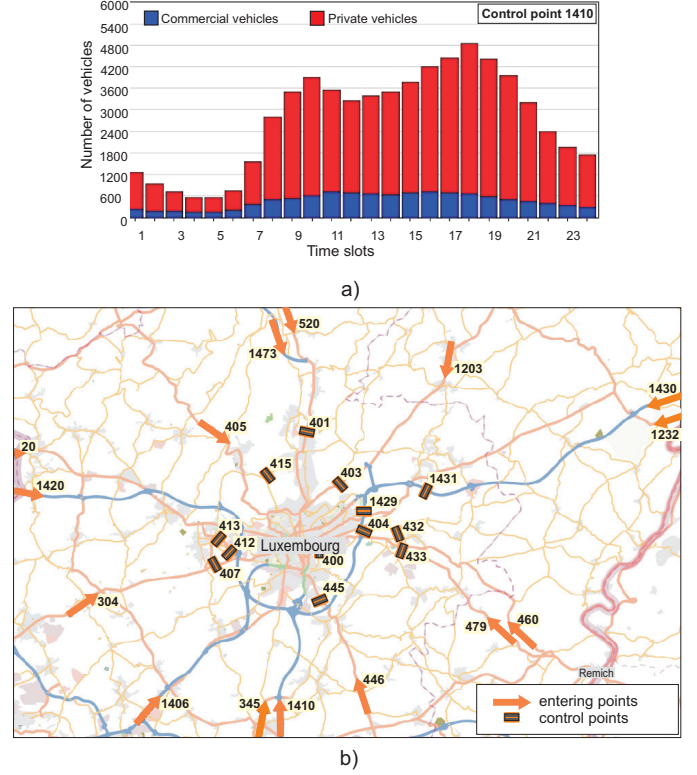


Fig. 2. Example of a traffic volume count for entering point number 1410. The count values correspond to 60 minutes time slots (a). Simulation area and locations of the counts at *entering points* and *control points*. The numbers associated to each location are assigned according to [21] (b).

selection probability. This property is modelled by the concept of attractivity areas. Such areas are defined independently for each zone type. Every zone location is associated with at least a default attractivity area. Additional attractivity areas might also be defined. Therefore, any zone location belongs either to a default attractivity area or to a specific area. The example shown in Fig. 3a illustrates three default attractivity areas corresponding to each zone type— $A_{CD}$  for the commercial zone,  $A_{ID}$  for the industrial zone, and  $A_{RD}$  for the residential zone. In addition, the map defines two specific attractivity areas for the commercial zone ( $A_{C1}$  and  $A_{C2}$ ) and one attractivity area for the industrial zone  $A_{I1}$ . Each attractivity area is assigned with a probability of being selected as a destination.

The selection of a destination is made using a three-step probabilistic model. The goal of the first step is to select a zone type. In the second step, the attractivity area for the zone selected in step 1 is chosen. Selection in both steps is made according to the probabilities assigned to zones as attractivities respectively. The third step is applied, if within the selected attractivity area, more than one zone of selected type exists. In such a case, the final selection of destination zone is positively correlated with the surface of the zone. Let us consider the example shown in Fig. 3a. Three zones with their attractivity areas are illustrated in Fig. 3a. Let us assume, that in step 1

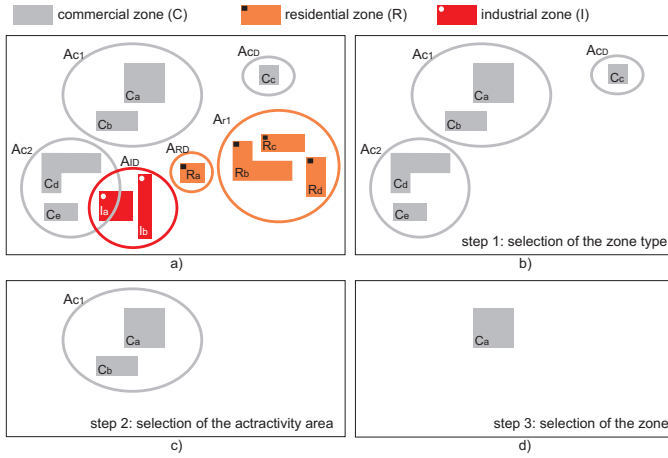


Fig. 3. Procedure for selection of a destination zone. Polygons represent the zones and circles represent related attractivity areas (a), step 1—selection of a zone type (b), step 2—selection of the attractivity area (c), step 3—selection of the zone.

the commercial zone type is selected (Fig. 3b). As zones of this type exist in three attractivity areas ( $A_{C1}$ ,  $A_{C2}$  and  $A_{CD}$ ), one of them needs to be selected in step 2. Let us assume that  $A_{C1}$  area is selected (Fig. 3c). In the final step, zone  $C_a$  is selected among the two commercial zones within attractivity area  $A_{C1}$ .

Finally, the route between origin-destination pair is generated using Dijkstra shortest path algorithm. Speed limits and traffic lights on roads are used to adjust the weight of each road in the generation process.

### C. Problem definition

VehILux model requires the following information: zone types, location of zones belonging to each type, and location of attractivity areas. This information can be partially extracted from a map, however it also requires an analytical process. Assuming that this information is available, the next step in VehILux is to discover the values of (i) the selection probabilities of each zone type, (ii) the selection probabilities of attractivity areas associated with each zone type, (iii) the inner traffic ratio, and (iv) a shifting ratio (described in Sec. IV-B). These values depend on the actual problem instance, which is defined by the geographical region and period of the day. Therefore, they should be discovered independently for each problem instance. This work proposes to use EAs to discover the parameters values mentioned above automatically. The objective is to find values of the parameters minimising the mean error. The error is defined as a difference between the number of vehicles counted in every control point using traffic flows generated by the model, and the actual traffic count volumes.

## IV. PROBLEMS SOLVING USING GENETIC ALGORITHMS

This section starts with a brief introduction of optimisation tools used in this work. Next, it moves to the problem domain tackled in this work, by defining the fitness function, solution encoding and genetic operators.

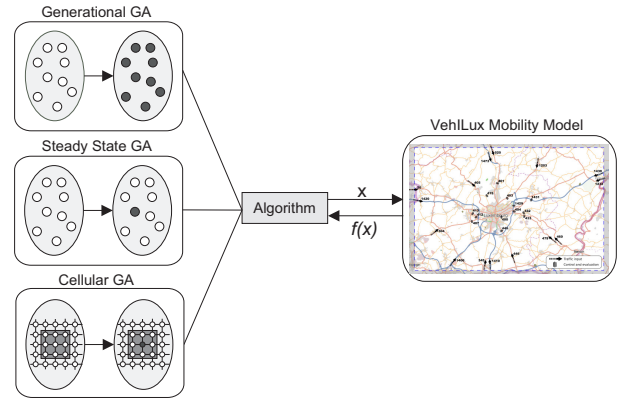


Fig. 4. Experimental study components.

### A. Optimisation tools

The overview of our approach is shown in Fig. 4. We compare three genetic algorithms with different population structures (panmictic and cellular). Two panmictic GAs are used—the generational GA (genGA) and the steady-state GA (ssGA). These algorithms do not consider any structure in the population, hence, any individual can mate with any other individual in the population during the breeding loop. Both genGA and ssGA perform in a similar way. Two individuals, i.e. parents, are chosen in the population using a given criterion (selection process). They are then recombined, the obtained offsprings are mutated and they are finally evaluated and inserted back into the population following a given criterion.

The genGA and the ssGA differ in the way the new individuals (offsprings) are inserted in the population. For the genGA, the obtained offsprings are inserted in an auxiliary population, that will replace the current population once full (it is a  $(\mu, \lambda)$ -GA, with  $\mu = \lambda$ ). The population is therefore updated in a synchronous way. In the ssGA, the new individuals are directly inserted into the current population (it is a  $(\mu+1)$ -GA). The population in the ssGAs is updated asynchronously with the newly generated individuals.

Cellular genetic algorithms (cGAs) [24] are a kind of GA with a structured (decentralised) population. Individuals are spread in a two dimensional toroidal mesh, and can interact with with a defined neighborhood. Fig. 5 presents the disposition of the individuals in the population of a cGA, the neighborhood of the center individual (shadowed), and of other individuals far from the center, in the four corners (dashed line).

The pseudo-code included in Algorithm 1 describes a canonical cGA. The population is typically structured in a regular grid of  $d$  dimensions ( $d = 1, 2, 3$ ), and a neighborhood is defined. Each individual is iteratively considered by the algorithm (line 3) and its neighbors are found (line 4). Parents are selected in the neighborhood (line 5) with some criterion. Recombination and mutation operators are applied in lines 6 and 7 with probabilities  $P_c$  and  $P_m$ , respectively. The fitness value is then computed for the new offspring individual (or individuals) (line 8) and it (or one of them) is inserted instead

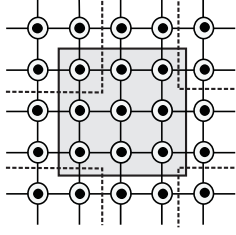


Fig. 5. cGA with  $5 \times 5$  population and C9 neighborhood.

of the current individual in the population (line 9) following a given replacement policy. This loop is repeated until a termination condition is met (line 2).

---

**Algorithm 1** Pseudocode for a canonical cGA

---

```

1: proc Evolve(cga) //Algorithm parameters in ‘cga’
2: while ! StopCondition() do
3:   for individual  $\leftarrow 1$  to cga.popSize do
4:     n_list  $\leftarrow$  Get_Neighborhood(cga.position(individual));
5:     parents  $\leftarrow$  Selection(n_list);
6:     offspring  $\leftarrow$  Recombination(cga.Pc,parents);
7:     offspring  $\leftarrow$  Mutation(cga.Pm,offspring);
8:     Evaluation(offspring);
9:     Add(position(individual),offspring,cga);
10:  end for
11: end while
12: end proc Steps_Up;

```

---

### B. Fitness function

In order to assign a fitness value to the candidate solutions of the different algorithms, we use the evaluation function  $F$ . The solved problem consists of traffic flows (sets of origin-destination pairs with corresponding routes). The quality of these flows is evaluated by comparing traffic volume counts at control points obtained from these estimated flows with real traffic volume counts. The fitness function is the following:

$$F = \sum_{c=1}^C \sum_{t=1}^T |r_c(t) - c_c(t)|, \quad (1)$$

where  $r_c(t)$  is the real traffic volume count at control point  $c$  in time slot  $t$ ,  $c_c(t)$  is the number vehicles at control point  $c$  derived from the generated traffic flows in time slot  $t$ ,  $C$  is the number of control points and  $T$  is the number of time slots. The objective is therefore to minimise the sum of the absolute difference between the real traffic volume counts and the estimated ones for all the control points for the simulated period.

Traffic flows are generated independently for each one hour time slot  $t$  and control point  $c$ , i.e.  $c_c(t)$ , as real traffic count data are typically collected on per hour basis. However, there is no precise departure time related to each origin-destination pair, except the 60 minutes slots (e.g. vehicle  $i$  departs from

point  $a$  to point  $b$  between 9 and 10AM). Such a function is proposed in [25]. It assumes that a vehicle passes through a control point at the same hour as it initiates the trip. This is not always true as the vehicle may pass through the control point in the next hour. To include this “delay” effect, we introduce a *shifting ratio* parameter, which defines the ratio of vehicles whose trips are scheduled to start in time slot  $t$ , but will pass through the control point in the slot  $t + 1$ . The estimated number of vehicles that pass through control point  $c$  within time slot  $t$  ( $c(t)$ ) is calculated as follows:

$$c_c(t) = p_c(t) \times (1 - \alpha) + p_c(t - 1) \times \alpha, \quad (2)$$

where  $p_c(t)$  is the number of all vehicles generated in time slot  $t$  that pass through control point  $c$ ,  $p_c(t - 1)$  is the number of vehicles generated in time slot  $t - 1$  that pass through control point  $c$  in time slot  $t$ , and  $\alpha$  is the shifting ratio.

The fitness  $F$  of candidate solutions is evaluated by independently running the VehiLux model on the considered problem instance. It first proceeds by creating the origins at each entering point for each time slot. The number of vehicles, whose trips starts at a given entering point is then defined by traffic count values for the given point. The candidate solution specifies the values of the parameters that enable to generate origins corresponding to inner traffic and destinations for inner and outer traffic. For each origin-destination pair, a route is generated. Then for each time slot, the total number of vehicles passing through each control point is calculated. Finally, the candidate solution is assigned its corresponding fitness value calculated with  $F$ .

### C. Solution encoding

Finding a suitable encoding of the solutions in the problem domain to a chromosome is one of the most significant problems in EAs. In this work we use an integer representation, where each gene represents one parameter of the VehiLux mobility model. The following parameters are defined: (i) the selection probabilities of each zone type, (ii) the selection probabilities of attractivity areas associated with each zone type, (iii) the inner traffic ratio, and (iv) the shifting ratio. The length of the chromosome depends on the actual problem instance, i.e. the numbers of zone types, zones and attractivity areas. Genes are divided into groups. The number of groups is equal to  $n + 3$ , where  $n$  is the number of zone types defined. The first group contains the probabilities for each defined zone type. The next  $n$  groups contain attractivity areas for each of the zone type. Finally, the last two groups only include one gene each, which represent the inner traffic and shifting ratios respectively. An additional constraint is imposed on each group, i.e. the sum of probabilities within the group must be equal to 100. For the fitness evaluation, all of the values for each gene are scaled so that they all fall in the range from 0 to 1.

An example—problem instance of Luxembourg—is shown in Fig. 6. Since there are three zone types defined, the chromosome is composed of 6 groups of genes. These genes encode the values corresponding to the following parameters:

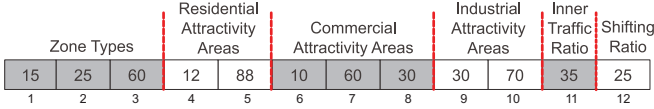


Fig. 6. Solution encoding.

- Genes 1–3: The probabilities assigned to the three zone types, i.e. industrial, commercial and residential respectively.
- Genes 4–5: The probabilities assigned to residential attractivity areas. Gene 4 encodes the default probability and gene 5 provides the probability for the single residential attractivity area defined in the problem instance.
- Genes 6–8: The probabilities assigned to commercial attractivity areas. Gene 6 encodes the default probability, while genes 7–8 represent the probabilities for the two commercial attractivity areas defined in the problem instance.
- Genes 9–10: The probabilities assigned to industrial attractivity areas. Gene 9 encodes the default probability, while gene 10 represent the probability for the single industrial attractivity area defined in the problem instance.
- Gene 11: The inner traffic ratio is bounded between 30 and 70 percent. These values are based on the sensitivity analysis conducted on the VehILux mobility model in [25].
- Gene 12: The shifting ratio is bounded between 20 and 80 percent. These values are based on our preliminary experimental results.

#### D. Crossover and mutation operators

The crossover operator is a genetic operator that combines two individuals with a fitness dependent probability to produce a offsprings. Experiments have been conducted using the single point crossover function (SPX) with crossover points restricted to five possible locations, represented as lines in Fig. 6. This ensures that the constraint on the sum of gene groups is respected.

The mutation operator introduces some randomness in the search to prevent premature convergence to local optima. We use a modified uniform mutation operator, which replaces the value of the chosen gene with a uniform random value selected between the upper and lower bounds of this gene. If the mutated gene belongs to one group (i.e. zone types or attractivity areas), and the new sum within a group does not equal to 100, the value of the other genes of the group is uniformly adapted to fulfill the constraint (sum equal to 100). Fig. 7 provides an example, in which the gene with value 88 is mutated and replaced by 42. As it belongs to the residential attractivity group, the gene with value 12 is also changed to 58.

## V. COMPUTATIONAL EXPERIMENTS

This section presents the results obtained on the VehILux mobility model optimisation problem using the three algorithms presented in Section IV-A. The problem instance used

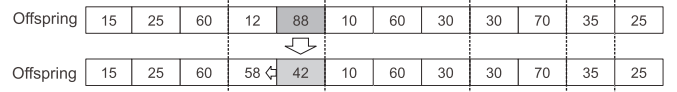


Fig. 7. Uniform mutation operator.

TABLE I  
PARAMETERS OF ZONES IN THE PROBLEM INSTANCE (LUXEMBOURG).

Zone type	#zones	$\Sigma$ surface ( $km^2$ )	Mean surface ( $m^2$ )
Commercial	772	8.82	11425
Industrial	47	14.20	302215
Residential	304	82.58	271645

in this work corresponds to Luxembourg. It was created using data obtained from [21]. This section first describes the specification of the parameters used in the experiments for both evolutionary algorithms and simulation. Then it describes the solution encoding for our problem instance. Finally, experimental results are given.

#### A. Parameters specification

In Table II, we show the parameters used for all the proposed algorithms and the simulations. The genGA and ssGA algorithms had a single population of 100 individuals and the cGA had a  $10 \times 10$  population. For all algorithms, the termination condition was limited to 6000 fitness function evaluations, as the computationally heavy fitness evaluation required on average 24 hours per run. The recombination operator (the single points crossover—SPX) was used with the probability  $p_c = 0.9$ . The uniform mutation operator described in Section IV-D, was applied with the probability  $p_m = 0.05$ . The binary tournament was used for the selection of the two parents. A specific parameter of the cellular model is the neighborhood. We used C9 (9 closest individuals measured in Manhattan distance—see Fig.5). Finally, an elitist strategy was used in genGA and cGA.

In the simulations a detailed map of Luxembourg taken from OpenStreetMap [26] was used. An area of 1700 square kilometers ( $47 \times 36$  kilometers) was chosen as illustrated in Fig. 2. Traffic volume counts were obtained from [21]. Among 27 traffic count locations, 15 were selected as the entering points, and 12 as the control points (see Fig. 2). For each control point, values corresponding to each hour (ranging from midnight until noon) were taken. During this period, most of the travel involves commuting, therefore, the entering points are selected from the locations positioned at the edges of the map. Three zone types are defined: *residential*, *commercial* and *industrial*. The information about the number and position of these zones was extracted from the OpenStreetMap. Precise information about these zones is given in Table I.

Solution encoding corresponding to the problem instance of Luxembourg is described in Sec IV-C.

#### B. Results

In this section, we present and analyse the results obtained in the experiments with the three GAs. In Table III, we

TABLE II  
PARAMETERS USED FOR THE STUDIED GAS

Genetic Algorithms	Population size	100 (genGA, ssGA) 10 × 10 (cGA)
	Termination Condition	6000 function evaluations
	Number of independent runs	30
	Selection	Binary tournament (BT)
	Neighborhood	C9 in cGA
	Crossover operator	SPX, $p_c=0.9$
	Mutation operator	Uniform, $p_m = 0.05$
	Elitism	1 individual (not for ssGA)
Simulations	Simulation area	1700 km <sup>2</sup> (47 × 36 km)
	Number of entering points	15
	Number of control points	12
	Simulated time period	12AM–12PM (12 time slots)

TABLE III  
EXPERIMENTAL RESULTS (FITNESS VALUES) OBTAINED WITH THE  
GENGA, SAGA AND CGA.

Algorithm	Best Result	Avg. Result
genGA	16209	17416.76 ±511.42
ssGA	<b>15717</b>	<b>17368.07</b> ±704.79
cGA	16135	17474.45 ±686.69

show for each algorithm the best fitness, the average fitness and the corresponding standard deviation obtained after 30 independent runs of the algorithms. It can be seen that the best performing algorithm for this problem is the ssGA. It provides both the overall best and best average results. The cGA provides the second best result but the worst average value. Fig. 8 presents the average fitness values for genGA, ssGA and cGA at each generation. It confirms that the panmictic ssGA performs best, with a faster convergence speed than the other two algorithms. The genGA and cGA have comparable convergence results.

Since the objective of this work is to optimise the accuracy of the VehILux mobility model, we study the properties of the best solution, i.e. the best set of VehILux parameter values, found by the evolutionary approach. Based on the experimental results, we analyse the overall best solution provided by the ssGA, which parameter values are presented in Table IV. To this end, the comparison of vehicle volume counts resulting from traffic flows generated by the evolutionary model and the real data for all 12 control points are shown in Fig. 9.

It can be observed that the traffic generated by the optimised model is close to the real one for 8 out of 12 counting loops. On the contrary, significant differences (difference greater than 25%) can be observed in control points number 404, 403, 420 and 432. In these locations, the traffic generated by VehILux is always lower than the real one. There may be several reasons for these differences. For instance, some roads might be more attractive than others, due to environmental features that are not included in the map. The greatest difference is observed in control point number 403, located on a national road entering the city. Indeed, the nearby motorway is selected more often

TABLE IV  
OVERALL BEST SOLUTION FOUND

Parameter	Value
Probability of selecting zone of residential type	0.03
Probability of selecting zone of commercial type	0.96
Probability of selecting zone of industrial	0.01
Default residential area attractivity	0.07
Defined residential area attractivity	0.93
Default commercial area attractivity	0.62
1 <sup>st</sup> defined commercial area attractivity	0.08
2 <sup>nd</sup> defined commercial area attractivity	0.30
Default industrial area attractivity	0.16
Defined industrial area attractivity	0.84
Inside flow ratio	0.75
Shifting ratio	0.36

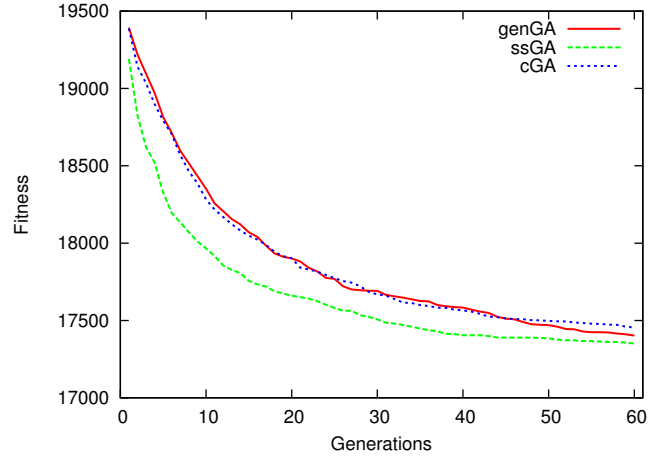


Fig. 8. The average fitness values for genGA, ssGA and cGA at every generation.

as a route to the centre of the city, due to its higher speed limit. However, in reality traffic congestion will occur during commuting hours and a significant portion of the traffic would use national roads. This parameter is currently not taken into account in the VehILux model, as routes are discovered using Dijkstra shortest path algorithm.

## VI. CONCLUSION

Simulation of inter-vehicle communication and related applications requires mobility traces. The recent trend in the evaluation process is to generate the traces using microscopic road traffic simulators and then use the traces as an input to network simulators. The more realistic these traces are, the more accurate the performance evaluation of the applications is. As the availability of real-world traces is limited, methods for generating realistic traces are needed. The recently proposed VehILux mobility model allows, in conjunction with a microscopic road traffic simulator, generating such traces for any region, for which traffic volume counts exist. However, the model does not address the question of how to select the values of the probabilities associated with attraction points. In this paper, we demonstrate how these probabilities can

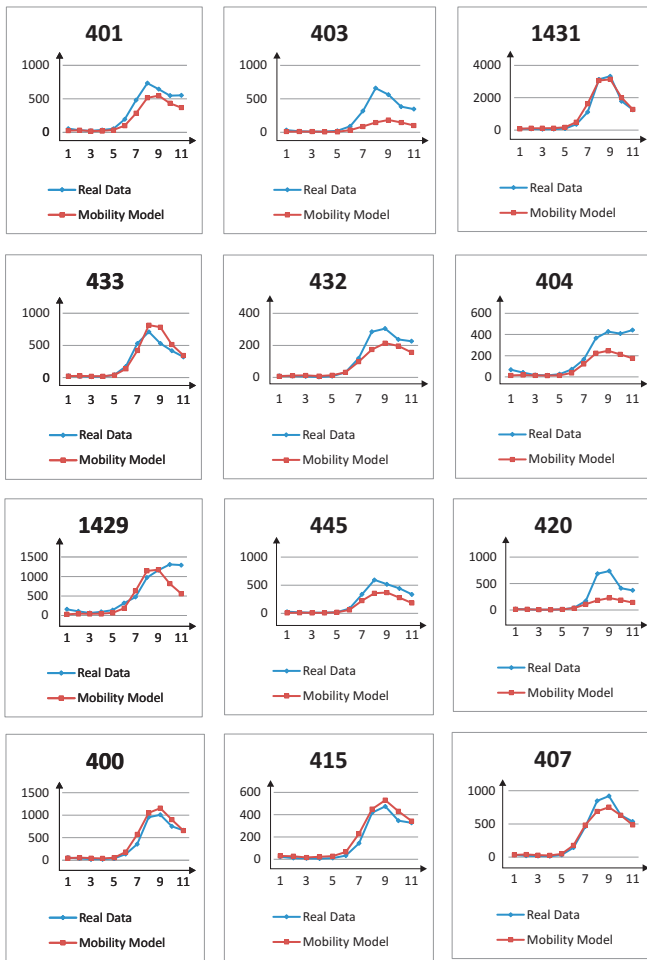


Fig. 9. Comparison between traffic volume counts obtained from vehicle flows generated by VehILux and the real traffic volume counts for the 12 control points. The x-axis represents time, y-axis represents traffic volume counts.

be discovered using GAs. A problem instance corresponding to Luxembourg is used to demonstrate our approach. Three GAs—a generational GA, a steady-state GA, and a cellular GA—are compared. The results are promising, however, they demonstrate a weakness of the VehILux model, namely, its route generation method. Currently, the model uses Dijkstra algorithm, which only considers posted speed limits as the major factor determining route selection. When simulating rush hours, such an approach leads to results which can not be validated by real data. Our future work is to improve the route generation model by adding “attractivity” weights to the road’s speed limits. The amount of these extra weights should depend on parameters such as time of the day or the actual traffic conditions on the road.

## REFERENCES

- [1] R. P-Zeletin, I. Radusch, and M. A. Rigani, *Vehicular-2-X Communication: State-of-the-Art and Research in Mobile Vehicular Ad hoc Networks*. Springer, 2010.
- [2] E. Giordano, R. Frank, G. Pau, and M. Gerla, “CORNER: a realistic urban propagation model for VANET,” in *Proc. of 7th international conference on Wireless on-demand network systems and services (WONS 2010)*, 2010, pp. 57–60.
- [3] —, “CORNER: A radio propagation model for VANETs in urban scenarios,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1280–1294, 2011.
- [4] M. Boban, T. Vinhoza, M. Ferreira, J. Barros, and O. Tonguz, “Impact of vehicles as obstacles in vehicular ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 15–28, 2011.
- [5] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.
- [6] ns-3 network simulator, [online] <http://www.nsnam.org/>.
- [7] C. Sommer and F. Dressler, “Progressing toward realistic mobility models in VANET simulations,” *IEEE Communications Magazine*, vol. 46, no. 11, pp. 132–137, 2008.
- [8] Y. Pigné, G. Danoy, and P. Bouvry, “A vehicular mobility model based on real traffic counting data,” in *Proc. 3rd International Workshop on Communication Technologies for Vehicles (Nets4Cars 2011)*, vol. 6596. Springer, LNCS, 2011, pp. 131–142.
- [9] J.-P. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*. New York: Routledge, 2009.
- [10] M. Seredynski and P. Bouvry, “A survey of vehicular-based cooperative traffic information systems,” in *Proc. 14th International IEEE Conference on Intelligent Transportation Systems (ITSC 2011)*. IEEE, 2011, pp. 163–168.
- [11] C. Sommer, R. German, and F. Dressler, “Bidirectionally coupled network and road traffic simulation for improved ivc analysis,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.
- [12] Y. Pigné, G. Danoy, and P. Bouvry, “A platform for realistic online vehicular network management,” in *Proc. IEEE International Workshop on Management of Emerging Networks and Services*. IEEE, 2010, pp. 615–619.
- [13] F. Dressler, C. Sommer, D. Eckhoff, and O. Tonguz, “Toward realistic simulation of intervehicle communication,” *IEEE Vehicular Technology Magazine*, vol. 6, no. 3, pp. 43–51, 2011.
- [14] M. Fiore, J. Harri, F. Filali, and C. Bonnet, “Vehicular mobility simulation for VANETs,” in *Proc. 40th Annual Simulation Symposium (ANSS 2007)*. IEEE, 2007, pp. 301–309.
- [15] J. Harri, F. Filali, and C. Bonnet, “Mobility models for vehicular ad hoc networks: a survey and taxonomy,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 19–41, 2009.
- [16] Umass dome testbed, <http://prisms.cs.umass.edu/dome/>.
- [17] The cabspotting project, <http://cabspotting.org/>.
- [18] CRAWDAD trace repository, <http://crawdad.cs.dartmouth.edu/data.php>.
- [19] J. Kim, V. Sridhara, and S. Bohacek, “Realistic mobility simulation of urban mesh networks,” *Ad Hoc Networks*, vol. 7, no. 2, pp. 411–430, 2009.
- [20] Q. Zheng, X. Hong, and J. Liu, “An agenda based mobility model,” in *Proc. 39th Annual Simulation Symposium (ANSS 2006)*, 2006.
- [21] Traffic volume counts in Luxembourg, Luxembourg ministry of transportation, [online] <http://www.pch.public.lu/trafic/comptage/index.html>.
- [22] Traffic volume counts in Massachusetts, USA, Massachusetts department of transportation, [online] <http://www.mhd.state.ma.us/default.asp?pgid=content/traffic01&sid=about>.
- [23] Traffic volume counts in New South Wales, Australia, [online] [http://www.rta.nsw.gov.au/trafficinformation/downloads/aadtdata\\_d11.html](http://www.rta.nsw.gov.au/trafficinformation/downloads/aadtdata_d11.html).
- [24] E. Alba and B. Dorronsoro, *Cellular Genetic Algorithms*, ser. Operations Research/Computer Science Interfaces. Springer-Verlag Heidelberg, 2008.
- [25] Y. Pigné, G. Danoy, and P. Bouvry, “Sensitivity analysis for a realistic vehicular mobility model,” in *Proc. first ACM international symposium on Design and analysis of intelligent vehicular networks and applications (DIVANet 2011)*. ACM, 2011, pp. 31–38.
- [26] OpenStreetMap project, [online] <http://www.openstreetmap.org/>.