

# Route selection for mobile sensor nodes on public transport networks

Olga Saukh · David Hasenfratz · Lothar Thiele

Received: 2 June 2012 / Accepted: 6 December 2012 / Published online: 29 December 2012  
© Springer-Verlag Berlin Heidelberg 2012

**Abstract** The sensing range of a sensor is spatially limited. Thus, achieving a good coverage of a large area of interest requires installation of a huge number of sensors which is cost and labor intensive. For example, monitoring air pollution in a city needs a high density of measurement stations installed throughout the area of interest. As alternative, we install a smaller number of mobile sensing nodes on top of public transport vehicles that regularly traverse the city. In this paper, we consider the problem of selecting a subnetwork of a city's public transport network to achieve a good coverage of the area of interest. In general case, public transport vehicles are not assigned to fix lines but rather to depots where they are parked overnight. We introduce an algorithm that selects the installation locations, i.e., number of vehicles within each host depot, such that sensing coverage is maximized. Since we are working with low-cost sensors, which exhibit failures and drift over time, vehicles selected for sensor installation have to be in each other's vicinity from time to time to allow comparing sensor readings. We refer to such meeting points as checkpoints. Our algorithm optimizes sensing coverage while providing a sufficient number of checkpoint locations. We evaluate our algorithm based on the tram network of Zurich and show how an accurate selection of vehicles for installing measurement stations affects the overall system quality. We show that our algorithm outperforms random search, simulated annealing, and the greedy approach.

**Keywords** Mobile sensors · Route selection · Area coverage · Sensor placement · Sensor checkpointing

## 1 Introduction

Today's big cities suffer from high concentrations of traffic and industrial facilities that heavily impact ecological sustainability and quality of living in the area. Monitoring air pollution, limiting the amount of transit traffic, and emission reduction has been addressed at many levels, but mostly through legislative decisions and standards. Due to high cost, weight, and size of traditional air pollution measurement instruments, there are still no precise air pollution maps with a *high spatial* resolution.

In the last several years low-cost gas sensors have become available on the market. We work on combining this technology with wireless sensor networks for air pollution monitoring applications. In particular, we install sensor nodes on top of several public transport vehicles to be able to achieve better coverage than in the case of statically deployed stations. Public transport networks, referred to as *timetable networks* (Pyrga et al. 2008) in the research community, form an attractive backbone for performing periodic measurements due to a (i) large number of spatially spread predefined routes, (ii) fixed timetables, and (iii) usually good reliability.

A measurement is usually valid in its temporal and spatial vicinity and thereby provides a certain amount of coverage in the region depending on the distance in space and time from the measurement location. A set of measurements resulting in a good coverage of the whole area of interest gives precise information about the measured phenomenon and its temporal development, leading to pollution maps with a high resolution.

---

O. Saukh · D. Hasenfratz (✉) · L. Thiele  
Computer Engineering and Networks Laboratory,  
ETH Zurich, Zurich, Switzerland  
e-mail: hasenfratz@tik.ee.ethz.ch

O. Saukh  
e-mail: saukh@tik.ee.ethz.ch

L. Thiele  
e-mail: thiele@tik.ee.ethz.ch

The first problem we face is how to choose a subnetwork of the public transport network such that the city is covered well given the route plan, timetables, and mapping of the vehicles to their host depots. The latter is required because usually transport vehicles are not assigned to fix lines but rather to depots (which are assigned to sets of lines) where they are maintained and parked overnight. Several years ago each tram served just one specific line. The policy was changed to improve flexibility. Today each depot serves a subset of all lines with its vehicles. Hence, we can not choose specific lines for sensor installation as assumed in Saukh et al. (2012), but only specify the number of vehicles to be equipped with sensing stations within each depot.

Since our sensor nodes are mainly equipped with low-cost gas sensors, we demand that the final subset of selected vehicles allows comparing measured sensor values across different sensor nodes, i.e., we require that different sensor nodes periodically take measurements at the same time and location. In this paper, the problem is referred to as *sensor checkpointing*.

Sensor checkpointing is an important property of a distributed sensing system and contributes to the system's fault tolerance. In particular, it allows recognizing sensor failures and provides necessary support for sensor calibration (Hasenfratz et al. 2012b). Informally, a pair of measurements performed by two separate sensors makes a *checkpoint* if these measurements are taken in each other's temporal and spatial vicinity. This naturally implies the simultaneous presence of the corresponding mobile vehicles at the same place.

When choosing a timetable subnetwork to provide both maximum coverage of the region and fulfill sensor checkpointing requirements, the solution space is huge even for moderate-sized cities for the following reasons: (i) public transport networks usually operate with several hundreds to a few thousands of mobile vehicles; (ii) tracks have different lengths; and (iii) vehicles within one depot can take different routes with different probabilities.

We investigate the problem of pre-deployment route selection based on the air pollution monitoring scenario in the city of Zurich, Switzerland, as part of the OpenSense project (Aberer et al. 2010). The long-term goal of OpenSense is (i) to raise community interest in air pollution and (ii) to encourage public involvement in the measurement campaign using enhanced cell phones or pocket sensors (Dutta et al. 2009). To establish an initial coverage of the city, we deploy sensors on top of several public transport vehicles, such as buses and trams. By doing so, we hope to foster community interest and its active involvement in data gathering (Hasenfratz et al. 2012a).

The main contributions of this paper are:

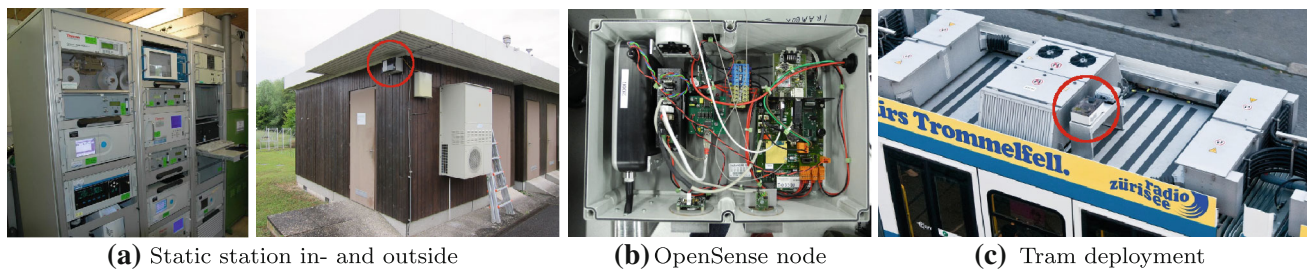
- Compared to our previous work (Saukh et al. 2012), we release the assumptions that each mobile vehicle follows a predefined track and a fixed timetable. We assume that each vehicle is free to take any route served by its host depot. Additionally, for each selected vehicle, a specific line can be favored to some extent, which increases the probability of taking the line, but with no guarantees.
- We propose a probabilistic model of the problem including the calculation of the expected coverage and the probability that the selected subset of vehicles form a connected subnetwork.
- Similar to Saukh et al. (2012), we solve the problem with an evolutionary algorithm and evaluate its performance using data of the Zurich tram network. We show that the evolutionary algorithm always achieves at least as good or better coverage than simulated annealing, random search, and greedy algorithms.

In the next section we describe the OpenSense scenario that leverages the solution to the route selection problem. In Sect. 3 we define the problem statement and introduce the terminology used throughout this work. Section 4 describes how to compute the area coverage and sensor checkpointing under uncertainties and presents an evolutionary algorithm for solving the problem. In Sect. 5 we evaluate our approach under different parameter settings. We review related work in Sect. 6 and conclude in Sect. 7

## 2 Sensing the air we breathe

Nowadays, air pollution is monitored by a network of static measurement stations operated by official authorities. The currently used measurement stations are equipped with traditional analytical instruments as depicted in Fig. 1a. These instruments are very precise but big in size, extremely expensive, and require frequent and laborious maintenance. The extensive cost of acquiring and operating such stations severely limits the number of installations (Carullo et al. 2007; Yamazoe and Miura 1995). For example, many big cities in Switzerland have only few such stations. Thus, the spatial resolution of the published pollution maps is very low. This is especially problematic for urban areas where the problem of air quality is highly important.

The concentration of air pollutants in urban areas is highly location-dependent. Motor vehicles, traffic junctions, urban canyons, and industrial facilities all have considerable impact on the local air pollution (Hasenfratz et al. 2012b). In recent years, several research groups started measuring the chemical pollutants in the atmosphere with low-cost solid-state gas sensors. Most of these



**Fig. 1** A conventional static measurement station and a mobile OpenSense prototype sensor node deployed on a tram

sensors are based on measuring an electrochemical reaction when exposed to a specific gas. These solid-state gas sensors are inexpensive, small, and suitable for mobile measurements. The integration of these sensors into mobile sensor nodes with wireless communication technologies was exploited by several recent works for mobile measurements (Choi et al. 2009; Honicky et al. 2008; Völgyesi et al. 2008), for example, by installing the sensors on top of public transport vehicles such as buses (Boscolo and Mangiavacchi 1998; Gil-Castiñeira et al. 2008). As part of the OpenSense project, we install sensor nodes on top of ten trams in Zurich (see Fig. 1b, c). The introduced sensor mobility allows to obtain a higher spatial measurement resolution and increase the covered area without the need of hundreds or thousands of sensors.

Our OpenSense nodes (see Fig. 1b) are based on the *CoreStation* platform (Buchli et al. 2011) that is a Gumstix (embedded computer) with a 600 MHz processor running the Angstrom Linux operating system. The station supports GPRS/UMTS and WLAN for communication and data transfer. A GPS receiver supplies the station with precise geospatial information. The weight of the OpenSense node is approximately 4.5 kg and has an energy consumption of around 40 W. The node is supplied with power from the tram. The station is equipped with  $O_3$ , CO, and particulate matter sensors.<sup>1</sup> The  $O_3$  sensor, a metal oxide semiconductor gas sensor, performs measurements by heating up a thin semiconducting metal oxide layer to several 100 °C. When  $O_3$  is present, the electric conductivity of the semiconductor is altered. The CO sensor is an electrochemical gas sensor that measures the concentration of a target gas by oxidizing and reducing the target gas at an electrode. The particulate matter sensor measures the particle number concentration and average particle diameter by evaluating the electrical charge of the particles (Fierz et al. 2011).

Low-cost gas sensors usually have limited accuracy and poor selectivity, which can be improved with periodic sensor recalibration using reference measurements from

high quality monitoring stations (Hasenfratz et al. 2012b; De Vito et al. 2008; Kamionka et al. 2006). Recalibration requires the presence of a sensing node at the location of a reference station from time to time (i.e., checkpointing) or, alternatively, the existence of a multi-hop path to the reference station over several different sensing stations (i.e., a sequence of checkpoints). In the next section we introduce the terminology and formalize the problem we are solving in this paper.

### 3 Models and problem statement

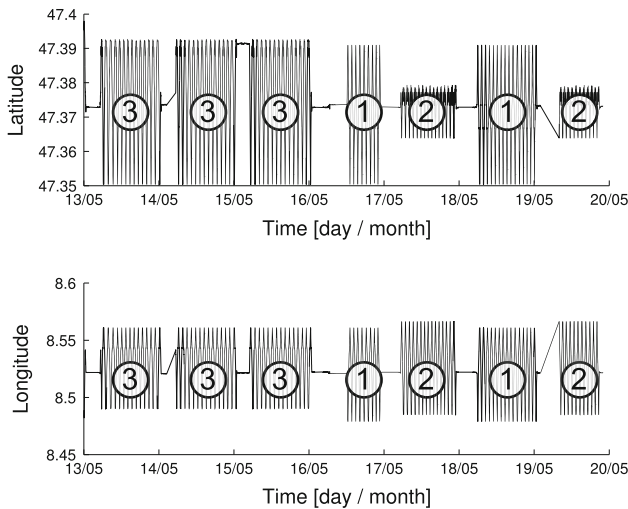
This section introduces the required terminology and the models that are used throughout this work to formally state and solve the problem of pre-deployment route selection with sensor checkpointing constraints.

#### 3.1 Timetable network

Let  $\Omega \subset \mathbb{R}^3$  represent an area and a time period of interest. A timetable network  $\mathcal{N} = (H, S, E)$  consists of a set of mobile vehicles  $H$ , a set of stations  $S$ , and a set of elementary connections  $E$  between immediate neighboring stations. A *timetable subnetwork*  $\mathcal{L} \subset \mathcal{N}$  is a timetable network induced by a subset of vehicles  $H_{\mathcal{L}} \subset H$  of  $\mathcal{N}$ . The *size of a timetable network* is defined as the number of vehicles  $|H_{\mathcal{L}}|$  the network comprises.

In our previous work (Saukh et al. 2012), we assumed that a mobile vehicle always follows its track and its timetable. In Zurich this was true until a few years ago. Recently, the public transport company suspended this restriction in order gain more flexibility when assigning vehicles to lines and scheduling vehicle replacements if necessary. Nowadays, public transport vehicles usually adhere to their host depots, where they are maintained and parked over night, rather than to individual lines. We show in Fig. 2 the recorded position of a tram over a period of 1 week. In this time frame, it served three different lines as it can be seen on the latitude and longitude patterns. The public transport company of Zurich (VBZ) provided us with the line assignment graph to different depots as

<sup>1</sup> MiCS-OZ-47 from <http://www.e2v.com>, CO-AF from <http://www.alphasense.com>, DiSCmini from <http://www.matter-aerosol.ch>.



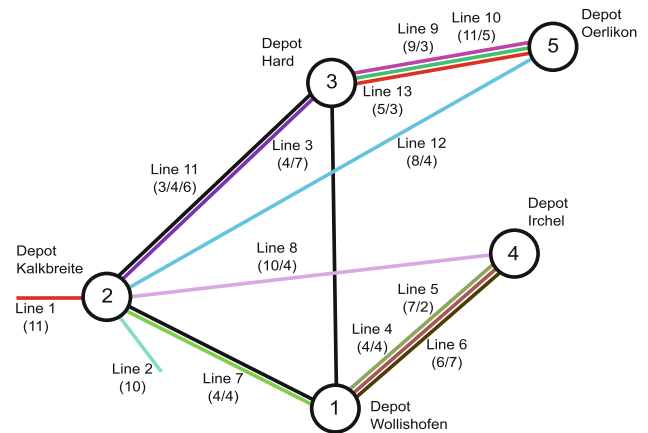
**Fig. 2** Latitude and longitude position of one tram over the course of 1 week. The tram is based in depot Kalkbreite and serves over the illustrated time period line 3, 1, and 2

schematically shown in Fig. 3.<sup>2</sup> Each circle in the figure represents a depot with a set of outgoing edges and rays showing tram lines served by the depot. If several depots are linked by edges of the same color, then the set of trams serving the line is distributed between these depots. The numbers in brackets below each line number denote the number of vehicles serving the line in the corresponding depots in clock-wise order. For example, the set of 13 trams serving line 11 is distributed among depots 3, 1, and 2 having 3, 4, and 6 vehicles in each depot, respectively. Within one depot, the operators freely choose which line is served by a tram on a particular day. In Sect. 4 we will use the line assignment graph to calculate the probability that a random tram within its depot chooses a certain line number.

### 3.2 Area coverage

Consider a mobile vehicle  $h \in H$  with a sensor node installed on top of it. Measurements can be taken by  $h$  while it is moving and at the stops without restrictions. A *measurement* consists of a sensor reading  $\zeta \in \mathbb{R}^+$  along with a location and a timestamp  $z \in \Omega$ . If a measurement is taken by a vehicle  $h \in H$ , we use the notation  $z \in h$  to express that the location  $z$  of a measurement belongs to the space-time movement curve of  $h$ . A measured value  $\zeta$  is valid in a close vicinity of the measurement location, i.e., within a certain area and for a certain time in  $\Omega$ . Let  $d = \|z - x\|$  be the distance from a point  $x$  to the measurement location  $z$ . Let  $w : \mathbb{R}^+ \rightarrow [0, 1]$  denote the *validity* of a

<sup>2</sup> The lines on the figure are re-numbered to have a sequential ordering of lines without gaps.



**Fig. 3** The line assignment graph maps tram lines to their host depots. Each circle represents a depot with a set of outgoing edges and rays showing tram lines served by the depot

measurement at location  $z$  at a point in its vicinity. For a point  $x \in \Omega$ ,  $w(d)$  represents the level of *coverage* at  $x$  provided by a measurement at  $z$ . Naturally,  $w$  monotonically decreases with increasing distance from  $z$ . We define the validity of a measurement as a strictly monotonically decreasing function independent of the actual measurement position or measurement time:

1.  $w(0) = 1$ ;
2.  $w(d_1) \geq w(d_2) \Leftrightarrow d_1 < d_2, \forall d_1, d_2 \in \mathbb{R}^+$ .

The exact decay of a measurement’s validity over distance highly depends on the monitored phenomenon. In our evaluation in Sect. 5 we will introduce specific expressions for the generic definition presented above.

Let a *density requirement function*  $\rho : \Omega \rightarrow [0, 1]$  represent the measurement density demand in the area  $\Omega$ . We use  $\rho$  to express the fact that some areas might require greater coverage than others. For a timetable subnetwork  $\mathcal{L} \subseteq \mathcal{N}$ , the *coverage* of  $\Omega$  achieved by  $\mathcal{L}$  is given by

$$C(H_{\mathcal{L}}) = \int_{\Omega} \rho(x) \max_{\forall z \in h, \forall h \in H_{\mathcal{L}}} w(\|z - x\|) dx \tag{1}$$

In this paper we are interested in finding a timetable subnetwork  $\mathcal{L}$  that maximizes the area coverage. Since a mobile vehicle is assigned to its host depot and not to a specific line, it can take any line hosted by its depot with a certain probability. The timetable subnetwork of interest is thus defined by a set of lines probabilistically taken by a selected number of vehicles in each depot.

### 3.3 Sensor checkpointing

Let  $h_1$  and  $h_2$  be two mobile vehicles equipped with air quality measurement stations. Let two measurements at

locations  $z_1 \in h_1$  and  $z_2 \in h_2$  be performed by  $h_1$  and  $h_2$ , respectively. Two measurements form a checkpoint if the distance between them in space and in time is below a certain threshold  $\epsilon$ , i.e.,  $\|z_1 - z_2\| < \epsilon$ .

Checkpoints are essential in distributed sensing systems, since they allow implementing mechanisms to detect failures of low-cost sensing hardware, identify sensor errors, and provide necessary support for automatic sensor calibration (Hasenfratz et al. 2012b). Compared to the notion of a *transfer* (Pyrga et al. 2008), introduced in timetable networks, checkpoints are direction-independent and have no transfer time.

### 3.4 Problem statement

Knowing the checkpoints between each pair of mobile vehicles, enables us to construct a checkpoint graph  $\mathcal{G} = (H_{\mathcal{L}}, E_{\mathcal{L}})$  where the set of nodes corresponds to the set of mobile vehicles  $H_{\mathcal{L}}$  equipped with sensing stations. There is an edge  $e \in E_{\mathcal{L}}$  between any two vehicles  $h_i, h_j \in H_{\mathcal{L}}$  if between the measurements of  $h_i$  and  $h_j$  there is at least one checkpoint.

To enable sensor checkpointing we need to ensure that the checkpoint graph  $\mathcal{G}$  is connected or  $k$ -vertex-connected, meaning that any two sensors can be compared over at least  $k$  vertex-independent paths. We refer to this type of checkpointing as X-checkpointing (*cross checkpointing*). In this paper we consider only 1-vertex-connectivity, although requesting  $k$ -vertex-connectivity would improve the system's resistance to traffic artifacts such as delays.

*X-Checkpointing:* Given a timetable network  $\mathcal{N}$ . Select a timetable subnetwork  $\mathcal{L}$  of size  $K$  to ensure maximum coverage of the area of interest  $\Omega$  under the condition that the checkpoint graph  $\mathcal{G}$  is  $k$ -vertex-connected.

Sensor checkpoints enable us to compare measurements among distinct sensor nodes and allows to detect faulty sensors in a similar fashion as majority voting. The quality of checkpointing can be further improved if the sensors can regularly synchronize with a set of *reference stations*  $\mathcal{R}$ . Reference stations can be static or mobile and are usually capable of performing high-quality sensing. We assume that all reference stations reflect the ground truth and can be used to calibrate low-cost sensors from time to time over one or several hops. This problem is referred to as R-checkpointing (*reference checkpointing*).

*R-Checkpointing:* Given a timetable network  $\mathcal{N}$  and a set of reference nodes  $\mathcal{R}$ . Choose a subnetwork  $\mathcal{L}$  of size  $K$  to ensure maximum coverage of the area of interest  $\Omega$  under the condition that each node in the checkpoint graph  $\mathcal{G}$  is  $k$ -vertex-connected to the set of reference stations  $\mathcal{R}$ .

Many big cities have a sparse network of highly precise static stations that can be used as references. For example,

in Zurich, Switzerland, there is one station of the national air quality monitoring network NABEL<sup>3</sup> and four smaller stations of the cantonal measurement network OstLuft.<sup>4</sup> The availability of such infrastructures allows us to considerably improve the system's reliability by selecting a timetable subnetwork that has R-checkpointing property. In contrast, many smaller cities have no reference station installed. In this case, pairwise cross-tests among low-cost sensors (i.e., X-checkpoints) are essential for being able to identify sensor faults and for recalibration needs.

In the next section we describe our solution approach to the defined route selection problem with sensor checkpointing constraints.

## 4 Route selection

This section presents the route selection algorithm using the assumption that a mobile vehicle can take any line served by its host depot. Given uncertainties which line is taken, we show how to calculate the expected coverage and the probability of fulfilling checkpointing constraints. Finally, we give the details of the developed evolutionary algorithm that solves the probabilistic route selection problem.

### 4.1 Line preference

Let  $H$  be a set of mobile vehicles in a city serving a set of lines  $L = \{l_j\}$ . These vehicles are hosted by a set of  $D = \{d_i\}$  depots in a city, such that each depot  $d_i$  serves a subset of lines  $L_i \subset L$ . Note that  $\bigcup L_i = L$  and the sets of lines served by different depots might overlap, meaning that vehicles operating on the same line can still be hosted by different depots. Since depots might have different capacities  $|d_i|$ , the distribution of vehicles among depots is generally unbalanced.

The set of vehicles  $H$  is partitioned between the depots  $\bigcup H_i = H, \forall i \neq j, H_i \cap H_j = \emptyset$ . Each vehicle can take different lines on different days, but can never change its host depot  $d_i$ . The set of vehicles that are hosted by depot  $d_i$  and serve the same line  $l_j$  are denoted with  $|H_{ij}|$ . We assume that the number of vehicles  $|H_i|$  is tight, meaning  $\sum_{j=1}^{|L_i|} |H_{ij}| = |H_i|$ .

For every depot  $d_i$ , the probability that a randomly chosen vehicle takes route  $l_j$  is given by

$$p_{ij} = \frac{|H_{ij}|}{|H_i|}, \quad \text{where} \quad \sum_{j=1}^{|L_i|} p_{ij} = 1. \quad (2)$$

The vector  $\mathbf{v}_i$  represents the probabilities of taking a line with a vehicle of depot  $d_i$

<sup>3</sup> <http://www.bafu.admin.ch/luft/luftbelastung>.

<sup>4</sup> <http://www.ostluft.ch>.

$$v_i = \begin{pmatrix} p_{i1} \\ p_{i2} \\ \vdots \\ p_i |L_i| \end{pmatrix} \tag{3}$$

The purpose of installing a measurement node on top of a vehicle is to achieve a good coverage of the city. However, if the vehicles take different lines at random, the expected coverage might be low, since not all lines make an equally good contribution to the coverage. Assuming some effort of the transport company to assign vehicles to lines in such a way that the vehicles chosen for installation do mostly follow the designated tracks, we introduce a constant  $\beta$  as a preference factor (alike the constant used in the PageRank algorithm (Brin and Page 1998) to represent the teleporting to a random page). Similar to the PageRank algorithm, we calculate the probabilities  $v_i^f$  of taking a line with a vehicle of depot  $d_i$  that is favoring line  $l_f$

$$v_i^f = \beta s_f + (1 - \beta)v_i \tag{4}$$

where  $\beta s_f$  is a vector with component  $\beta$  in the position of the favored vehicle.<sup>5</sup> The term  $(1 - \beta)v_i$  represents the case where, with probability  $1 - \beta$ , the vehicle decides to follow the usual probability distribution. The term  $\beta s_f$  represents the case where the vehicle takes the favored line  $l_f$ .

### 4.2 Probabilistic coverage

Consider a set  $H_L$  of mobile vehicles selected for the installation of  $|H_L|$  measurement stations. Each selected vehicle  $h_{if} \in H_L$  is assigned to a host depot  $d_i$  and has a favorite line  $l_f$ . For each line  $l_j \in L$  we define the *line load*  $e_j$  as the expected number of vehicles serving the line as follows

$$e_j = s_j^T \sum_{\forall h_{if} \in H_L} v_i^f \tag{5}$$

Additionally, for each line  $l_j$ , we calculate the *line utilization*  $u_j$  as the probability that a vehicle in  $H_L$  is serving line  $l_j$

$$u_j = 1 - \prod_{\forall h_{if} \in H_L} (1 - s_j^T v_i^f) \tag{6}$$

In the following we introduce a few heuristics and simplifications to estimate the achieved coverage with a set of mobile vehicles  $H_L$ . We assume that the tracks of all lines are close to straight line segments. Consider a track  $l_j$  is a straight line segment of length  $\|l_j\|$  with  $c$  vehicles  $y_1, y_2, \dots, y_c$  operating on the line (see the upper line in

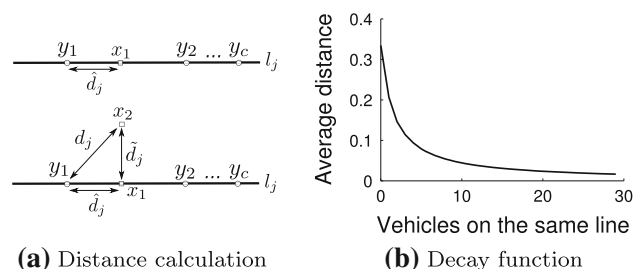
Fig. 4a). Although the timetables of single vehicles are unknown, we can estimate the average distance  $\hat{d}_j$  from a random point  $x_1$  on line  $l_j$  to the closest vehicle on the line as follows

$$\hat{d}_j = \frac{1}{\|l_j\|^c} \int_0^{\|l_j\|} \dots \int_0^{\|l_j\|} \int_0^{\|l_j\|} \min_{1 \leq k \leq c} \|y_k - x\| dy_1 \dots dy_c dx \tag{7}$$

According to this formula, if there is only one vehicle operating on  $l_j$ , the expected distance to this vehicle from a random point taken on  $l_j$  is  $\hat{d}_j = \frac{1}{3}\|l_j\|$ , for two vehicles it is reduced to  $\hat{d}_j = \frac{5}{24}\|l_j\|$  and so on. In Fig. 4b, we plot the values of  $\hat{d}_j$  for an increasing number of vehicles operating on the same line with line length  $\|l_j\| = 1$ . Since the line load  $e_j$  is in general fractional we linearly interpolate between  $\lfloor e_j \rfloor$  and  $\lceil e_j \rceil$  to estimate the distance to the closest vehicle on the line.

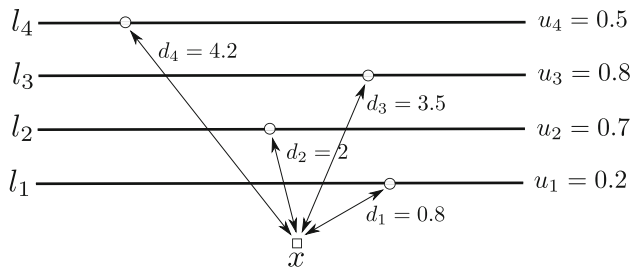
In the general case, we are interested in the distance from a random point  $x_2$  in the area of interest  $\Omega$  which does not necessarily lie on the track, as shown on the lower line in Fig. 4a. Let  $\tilde{d}_j$  be the minimal distance from  $x_2$  to the closest point  $x_1$  on the line  $l_j$ . We use the Pythagoras formula to calculate distance  $d_j = (\tilde{d}_j^2 + \tilde{d}_j^2)^{\frac{1}{2}}$  between  $x_2$  and the closest vehicle on line  $l_j$ .

We have to calculate the average expected distance not only to the vehicle on the closest line but also to other lines, since the utilization  $u_j$  of the closest line  $l_j$  is usually below 1. We assume that lines are sorted according to their distance  $d_j$  from a random point  $x \in \Omega$  as shown in Fig. 5 on the example of four lines. Simply calculating the average over all probabilistic distances  $d_j \cdot u_j$  does not give a good approximation, because the average probabilistic distance can increase if additional lines and mobile vehicles are considered, independent of their location. This problem occurs if  $\sum_{l_j \in L} u_j > 1$ . Consider the simple example of two lines  $l_1$  and  $l_2$  with utilizations  $u_1=1$  and  $u_2=0.5$  and distances  $d_1=5$  and  $d_2=100$ . Since with 100 % probability there is a vehicle on line 1, the probabilistic distance is 5. However, if we add up all probabilistic distances then we



**Fig. 4** Distance calculation to the nearest vehicle on a *line* and the decay in average distance to the closest vehicle on the *same line*

<sup>5</sup> We denote with  $s_f$  a selector vector whose  $f$ th element is 1 and all other elements are 0, i.e.,  $s_f^T = (0, 0, 1, 0, \dots, 0)$ .



**Fig. 5** Distance from a random point  $x$  in the area of interest  $\Omega$  to four tracks with a mobile vehicle on each track with a certain probability

receive a probabilistic distance of  $5 \times 1 + 100 \times 0.5 = 55$ . The same problem occurs if the utilizations  $u_j$  are normalized. This leads to an probabilistic distance of  $5 \times 2/3 + 100 \times 1/3 = 40$ . Therefore, we use the probabilistic function  $psgn$  to calculate the average probabilistic distance  $d(x)$  as

$$d(x) = \sum_{j=1}^{|L|} d_j \cdot psgn(u_j) \tag{8}$$

where  $psgn(u_j)$  is defined as

$$psgn(u_j) = \begin{cases} u_j, & \text{if } \sum_{k=1}^j u_k < 1; \\ 1 - \sum_{k=1}^{j-1} u_k, & \text{if } \sum_{k=1}^{j-1} u_k < 1 < \sum_{k=1}^j u_k; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

which does not consider additional probabilities if the sum of utilizations reaches 1. For the example in Fig. 5, we receive an average distance  $d(x) = 0.8 \cdot 0.2 + 2 \cdot 0.7 + 3.5 \cdot (1 - 0.2 - 0.7) = 1.91$ .

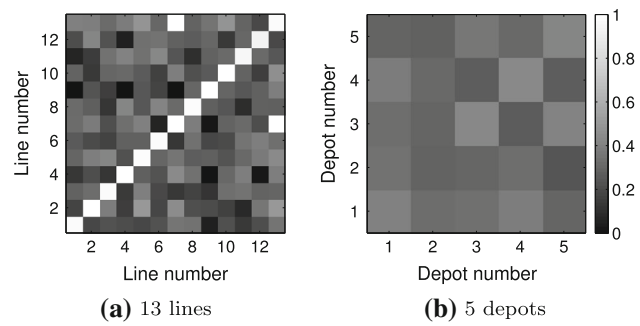
We make use of the above findings to estimate the validity  $w(d(x))$  of a measurement at distance  $d(x)$  from a random point  $x \in \Omega$  and calculate using Eq. 1 the probabilistic area coverage as

$$C(H_{\mathcal{L}}) = \int_{\Omega} \rho(x)w(d(x)) \, dx \tag{10}$$

Note that the estimation of the area coverage computed this way is submodular (Krause et al. 2009).

### 4.3 Probabilistic checkpoints

For each pair of lines  $l_i$  and  $l_j$  in the area of interest  $\Omega$ , we consider all possible starting locations of a public transport vehicle based on the lines' timetables. We calculate the probability  $P(l_i, l_j | 1, 1)$  that two vehicles operating on  $l_i$  and  $l_j$  (one on each) make a checkpoint. If two vehicles operate on the same line ( $i = j$ ), their checkpoint probability is always  $P(l_i, l_i | 1, 1) = 1$ . The checkpoint



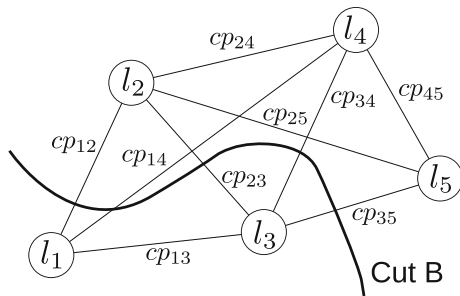
**Fig. 6** Probabilistic pairwise checkpoints of the 13 lines and 5 depots in Zurich

probability is 0 if the lines do not cross. Figure 6a shows as example the checkpoint probabilities of all 13 tram lines in Zurich. Similarly, we can calculate the checkpoint probability of all five depots in Zurich as illustrated in Fig. 6b. The probabilities are computed by selecting two vehicles on two lines at random and operating them according to their timetables during the time period of 2 h. After 50 trials we compute the average checkpoint probability between the two lines.

In general, multiple vehicles may serve the same line. In this case, they always make a checkpoint with a frequency depending on the round trip tour of a tram operating the line. Consider two crossing lines  $l_i$  and  $l_j$  with average line loads  $e_i$  and  $e_j$  respectively, as defined by Eq. 5. Vehicles on the same line make checkpoints with each other. We are interested to find the probability that the vehicles across different lines meet. There are  $e_i \cdot e_j$  possible rendezvous chances between vehicles on different lines, which leads to the following checkpoint probability between the lines

$$cp_{ij} = P(l_i, l_j | e_i, e_j) = 1 - (1 - P(l_i, l_j | 1, 1))^{e_i e_j} \tag{11}$$

We construct a checkpoint graph  $\mathcal{G}(L, E)$ , where  $L$  is the set of lines traversed by at least one vehicle in  $H_{\mathcal{L}}$  and  $E$  is the set of edges between those lines that have a non-zero checkpoint probability. In case of R-checkpointing we add for every reference station a vertex and edges the reference station and the lines that pass by it. Each edge  $(l_i, l_j) \in E$  has an assigned weight corresponding to the checkpoint probability  $cp_{ij}$  as shown in Fig. 7. In the following we want to calculate the probability that a checkpoint graph fulfills the checkpointing constraints. The problem is similar to the problem of estimating network reliability if each link has a certain failure probability. This problem is usually solved by finding minimum cuts in the graph (Karger 1995; Alon 1995). A *cut* is a partitioning of the vertices of a graph into two disjoint subsets. Consider a cut  $B$  in  $\mathcal{G}$  as shown in Fig. 7. The probability that vehicles operating on two lines  $l_i$  and  $l_j$  connected by an edge  $(l_i, l_j) \in B$  fail to make a checkpoint is  $1 - cp_{ij}$ . Thus, the



**Fig. 7** Checkpoint graph of 5 lines with annotated checkpoint probabilities and a possible cut

probability that checkpoints will not fail along all edges in cut  $B$  simultaneously is defined as

$$\phi(B) = 1 - \prod_{(i,j) \in B} (1 - cp_{ji}) \tag{12}$$

A checkpoint graph  $\mathcal{G}$  remains connected with a probability at least  $\theta$  if  $\theta \leq \min \phi(B)$ . If this is the case, we say that  $\mathcal{G}$  is  $\theta$ -connected and fulfills checkpointing constraints. Note that if we set  $\theta$  to 1, the only solution fulfilling checkpointing constraints consists of all vehicles scheduled on the same line.

#### 4.4 Selection algorithm

The formulated problem of pre-deployment route selection involves high computational complexity even for moderate-sized cities. The brute force approach would require to go through  $|D|^{|H_{\mathcal{L}}|} \cdot |L|$  combinations to optimally select the  $|H_{\mathcal{L}}|$  vehicles for installation which each belong to one of  $|D|$  depots and have one favorite line among all possible  $|L|$  lines. In Table 1 we present some network characteristics of several cities world wide. In case of the tram network Zurich, the smallest among the listed transport networks, there are more than  $5^{10} \times 13 \approx 100,000,000$  possible solutions for the installation of  $|H_{\mathcal{L}}| = 10$  sensor nodes on top of trams served by  $|D| = 5$  depots and assuming

**Table 1** Type of network and number of lines, vehicles, stops, and depots of public transport companies in different cities

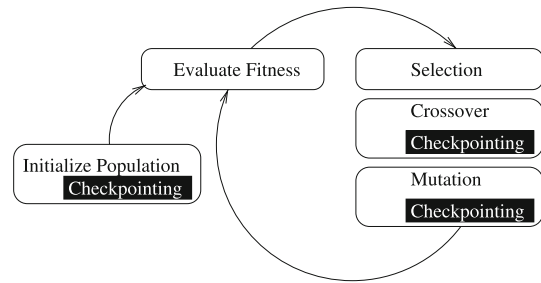
City	Netw.	Lines	Vehicles	Stops	Depots
Zurich <sup>a</sup>	Tram	13	260	187	5
Zurich <sup>a</sup>	Bus	283	732	2,543	n/a
Berlin <sup>b</sup>	Bus	149	1,338	2,634	n/a
Chicago <sup>c</sup>	Bus	140	1,781	11,493	n/a
New York <sup>d</sup>	Bus	324	5,908	15,226	n/a

<sup>a</sup> <http://www.zvv.ch>

<sup>b</sup> <http://www.bvg.de>

<sup>c</sup> <http://www.transitchicago.com>

<sup>d</sup> <http://www.mta.info>



**Fig. 8** Structure of the proposed evolutionary algorithm

$|L| = 13$  possibilities to choose a preference line within a depot. Neither X- nor R-checkpointing constraints reduce the size of the worst case solution space. Hence, performing exhaustive search is not feasible. Although our objective function  $C(H_{\mathcal{L}})$  is submodular (or convex), checkpointing constraints of the discussed form make the problem very hard to solve. Greedy-based approaches, which give approximation guarantees for many submodular function optimization problems, perform arbitrary poor in our case. In this paper, we choose to use an *evolutionary algorithm* (Back et al. 1997; Goldberg 1989) to approach the problem solution.

The working principle of evolutionary algorithms is schematically depicted in Fig. 8. The initial population (i.e., first generation) is composed of randomly generated candidate solutions. The fitness of all these solutions is evaluated before the best solutions are chosen for the next generation. Then, new solutions are generated by recombining two existing solutions (i.e., crossover) and randomly altering current solutions (i.e., mutation). In the remaining part of this section we describe the representation of a chromosome, the used fitness function, the selection scheme, and the variation operators crossover and mutation.

A *chromosome* encodes a candidate solution of our route selection problem and is of length  $|H_{\mathcal{L}}|$ . Each gene in the chromosome represents a vehicle chosen for installation and is defined by the vehicle’s (i) host depot and (ii) favorite line. Note, that this representation is rather different from the one used in Saukh et al. (2012), where each gene uniquely identified a tram, i.e., included a line number and a time when it departed from a specific station. Based on the new probabilistic model, the trams within one depot are indistinguishable. A chromosome satisfies the checkpointing constraint if the checkpoint graph  $\mathcal{G}$  is  $\theta$ -connected. We use direct representation and do not transform our chromosomes into a binary representation form.

We begin with an initial population composed of a random set of chromosomes. In case of X- and R-checkpointing we only consider chromosomes that fulfill the corresponding checkpointing constraint, i.e., we keep



generating random chromosomes until we have the desired number of feasible chromosomes.

The *fitness* of a subnetwork represents the coverage of the area of interest  $\Omega$  provided by the subnetwork. Calculation of the coverage as defined by Eq. 10 involves discretization of the region of interest by regular or Monte Carlo sampling and computation of the level of coverage at each of the sampled points. To speedup coverage computation, the distances  $d_j, \forall j = 1, \dots, |L|$  in Eq. 8 for all regularly sampled points are computed only once as part of the algorithm initialization.

To variate the given set of chromosomes, we first randomly assign the chromosomes to pairs. Then, a *crossover operator* is applied to each pair with probability 0.7. We use uniform crossover with a mixing rate of 0.5. In case of X- and R-checkpointing we only consider chromosomes that fulfill the corresponding connectivity constraint.

Each chromosome is *mutated* with probability 1 by selecting one of its genes at random, and replacing it by a randomly generated gene. We keep generating offspring in this way for a maximum of five iterations, until a chromosome that satisfies the given constraints is found. No offspring is generated if no feasible chromosome can be constructed.

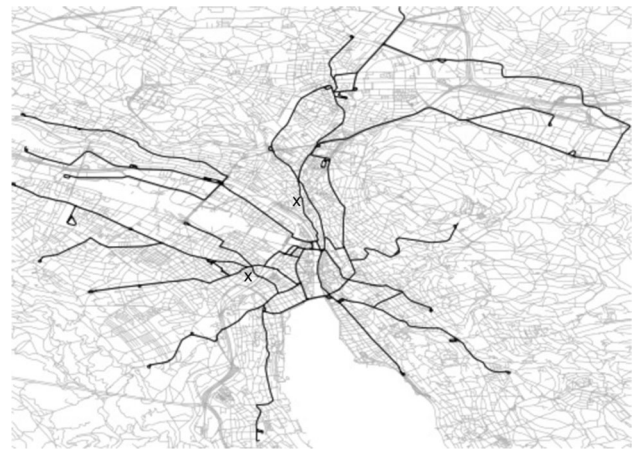
After calculating the fitness of the newly created offspring, the algorithm uses the restricted tournament (Harik 1995) *selection operator* to decide which chromosomes of the parents and offspring are going to survive into the next iteration. In restricted tournament, a variated chromosome can only substitute its more similar parent, and only if its fitness value is better than the parent's fitness. The approach is elitist since the best solution is always kept. Also, the selection operator always preserves checkpointing constraints.

## 5 Evaluation

Similar to our previous work (Saukh et al. 2012), we evaluate our approach based on the tram network of Zurich. Zurich is one of the target cities in OpenSense with the goal to deploy a network of mobile sensors on top of up to 10 trams.

### 5.1 Setup

We run the algorithm on the data of the Zurich tram network. We obtain track plans from OpenStreetMap<sup>6</sup> and timetables from the Zurich public transport (ZVV) website.<sup>7</sup> Based on the OpenStreetMap data we extract all



**Fig. 9** Tram network of Zurich. The positions of the two reference stations used in our evaluation is marked with  $x$

possible tracks of Zurich as depicted in Fig. 9. ZVV serves 13 tram lines (see Table 1) with the involvement of maximum 260 individual trams (whereas only a subset of trams is involved in daily operation).

We calculate pairwise checkpoint probabilities between trams operating on different lines based on the tram network timetable on a business day at 7 o'clock in the morning. We select a time slice of 2 h in order to include the round trip time of all operating trams. For simplicity, we do not consider any differences between business days and weekends or times of the day. The speed of a tram between two stations is linearly interpolated, although it might differ between two station pairs depending on their timetables. The timetable information is only used when calculating pairwise checkpoint probabilities between tram lines as described in Sect. 4.3.

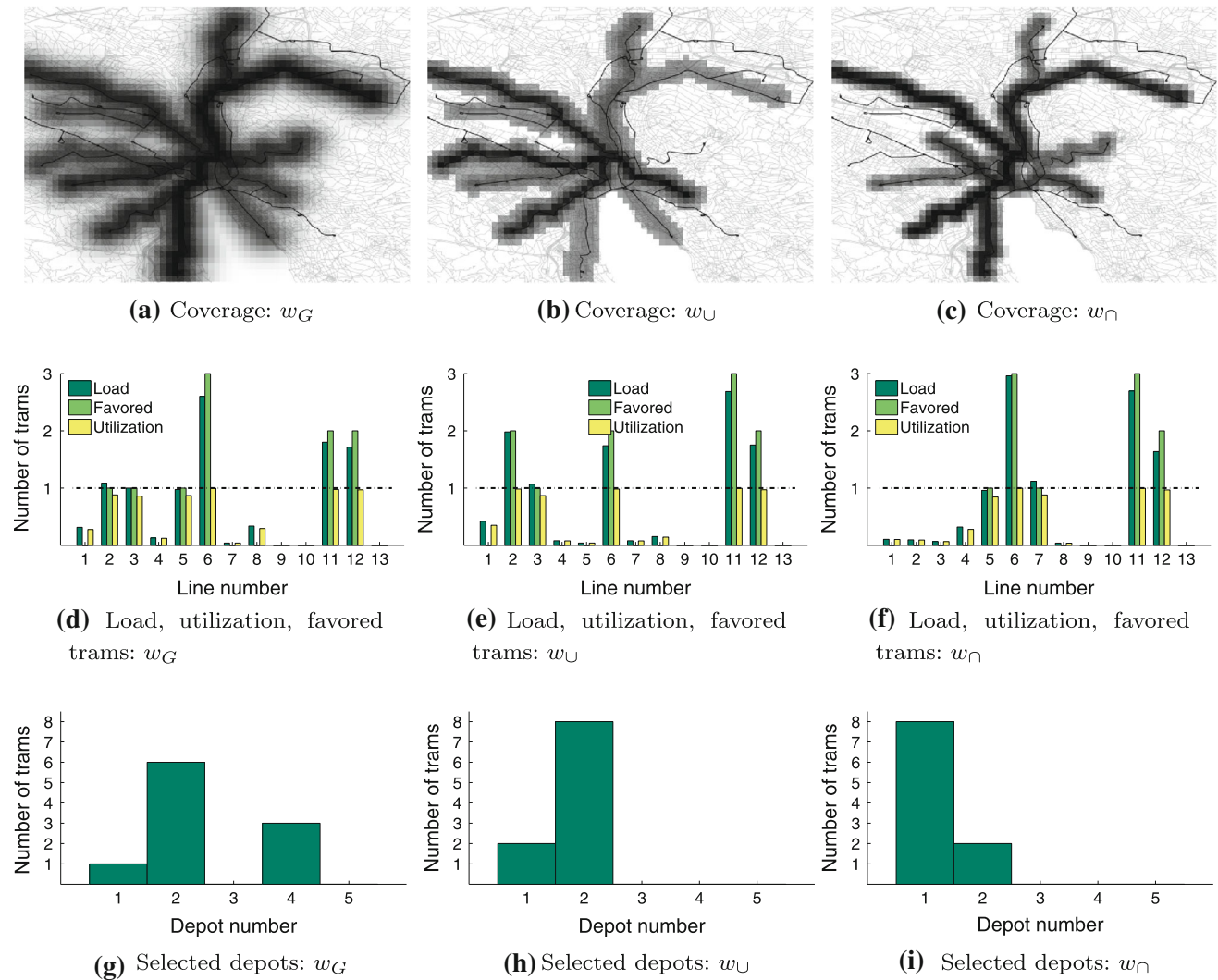
The population size is set to 10 chromosomes. Compared to the previously published results Saukh et al. (2012) which suggested to use 60 chromosomes to solve the route selection problem for the Zurich tram network, the property of the new probabilistic model that the individual trams are indistinguishable within one depot allows to reduce the population size, required to achieve good selection results. This also solves the scalability issue persisted at the early stage of this work. Note, that Zurich is quite a small city (also see Table 1) and that the population size must be increase when using the algorithms on bigger timetable networks.

The area of interest  $\Omega$  is Zurich city, as depicted in Fig. 9. To simplify the understanding of the results, we present solutions for the uniform density requirement  $\rho \equiv 1$ . We stop the evolutionary algorithm after 50 iterations and repeat the execution 50 times.

We consider that two trams make a checkpoint if they are closer to each other than  $\epsilon = 200$  m. This corresponds to non-zero measurement validity  $w$  in all points in space

<sup>6</sup> <http://www.openstreetmap.org>.

<sup>7</sup> <http://www.zvv.ch>.



**Fig. 10** Predicted area coverage, line load, and depot allocation for three distinct validity functions with line preference factor  $\beta = 0.8$ , checkpoint threshold  $\theta = 0.9$ , and X-checkpointing constraint

within radius  $\epsilon$ . We evaluate our selection algorithm with three different validity functions:

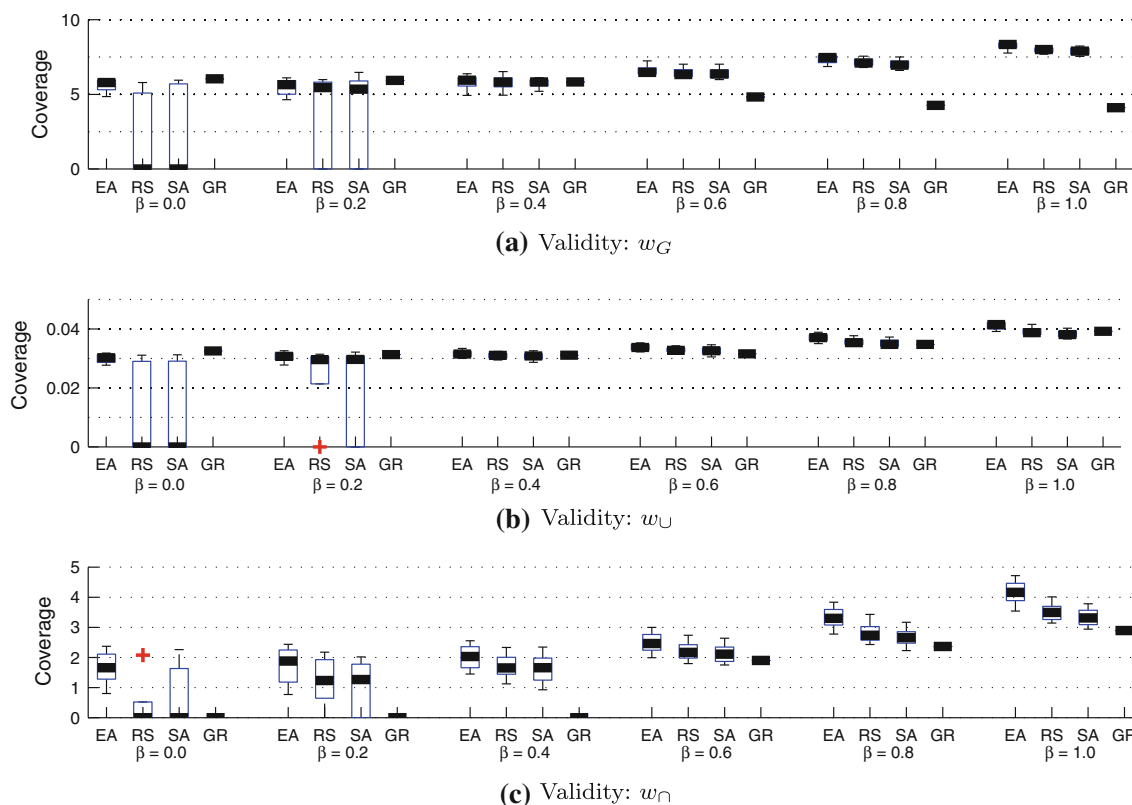
- Convex decay  $w_U(d) = \frac{1}{1+d}$ .
- Concave decay  $w_n(d) = \max(0, 1 - \frac{d^2}{\gamma^2})$ ,  $\gamma = 500$  m.
- Gaussian decay  $w_G(d) = e^{-\frac{d^2}{2\sigma^2}}$  with zero mean and  $\sigma = 500$  m.

The first two validity functions  $w_U$  and  $w_n$  differ significantly in their impact on the calculation of coverage due to their complementary convexity properties. The Gaussian  $w_G$  is a mixture of both. The validity function considerably impacts the maximum achievable coverage. Hence, it is important to choose the right validity function depending on the properties of the monitored phenomenon of interest. For example,  $w_U$  better suits for the CO and particulate matter phenomena that exhibit spike-like signals in space

and time, whereas  $w_n$  and  $w_G$  are more suitable for smooth phenomena such as  $O_3$  that is slowly changing over time (in the order of minutes) (Vardoulakis et al. 2003).

In the following paragraphs we describe the results obtained by our algorithm for several exemplary runs and further investigate the influence of parameter settings on the achieved coverage. In particular, we vary the values of our main algorithm parameters, the preference factor  $\beta$  and checkpoint threshold  $\theta$ , to reveal the impact on the achievable coverage. We compare the solutions of the evolutionary algorithm with random search, simulated annealing (Skićim and Golden 1983), and a greedy approach shortly described below.

*Simulated annealing* starts with a random feasible solution and uses the same mutation operator as the evolutionary algorithm to generate new candidate solutions. The algorithm always accepts a better solution and



**Fig. 11** Impact of the line preference factor  $\beta$  on the expected city coverage with checkpoint threshold  $\theta = 0.9$  and X-checkpointing constraint for four different algorithms: evolutionary algorithm (EA), random search (RS), simulated annealing (SA), and greedy (GR)

probabilistically a slightly worse solution than the current one depending on the number of iterations left. The algorithm makes  $10 \times 50 = 500$  steps to make the results comparable to our evolutionary algorithm.

*Random search* generates 500 random solutions and picks the one which satisfies the required checkpoint constraint and provides the highest coverage.

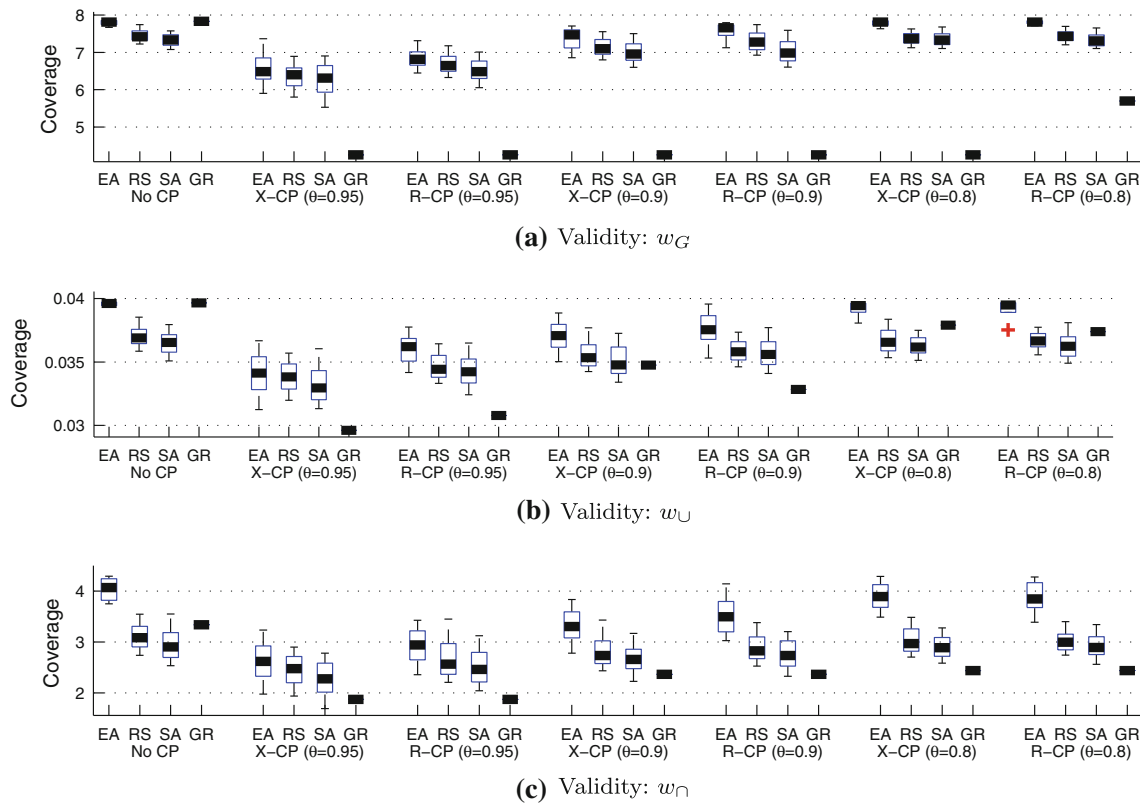
*Greedy* starts by selecting the depot and favorite line that provide the highest coverage. In each subsequent step the depot and favorite line is chosen such that the current partial solution satisfies the given constraints and maximizes coverage. It can happen that a partial solution makes it impossible to fulfill the checkpointing constraint in the next step independent which depot and favorite line is chosen. In this case Greedy is not able to calculate a valid solution and returns coverage zero.

### 5.2 Results

Figure 10 presents exemplary solutions found with our evolutionary algorithm for  $\beta = 0.8$  and  $\theta = 0.9$  under X-checkpointing constraints for all three introduced validity functions. The first row of plots Fig. 10a–c illustrates the amount of coverage in the area of interest. The achieved coverage is high in the vicinity of the tracks and

decreases with increasing distance, clearly visible in Fig. 10b with  $w_U$  as validity function. Additionally, we see that the overall achievable coverage highly depends on the chosen validity function. In the second row of plots with Fig. 10d–f we show load and utilization of each line and indicate the lines favored in the solution. The plots show that multiple trams should favor the same line. The reason for this is that some lines provide good coverage, but are very long. It is thus beneficial to favor a good line by several vehicles. In the plots, the load of selected lines is almost as high as the number of trams favoring the line due to a high preference factor  $\beta = 0.9$ . For lower values of  $\beta$  the randomness dominates, which also reduces the achievable coverage, as we will show later. It can be seen, that all three solutions include line numbers 6, 11, and 12 multiple times, and two solutions include line number 2 and 3. These lines are favored in our current installation of five measurement stations on top of five trams in the city of Zurich.<sup>8</sup> The last row of plots Fig. 10g–i shows the number of trams selected in each depot. Depot 1 and 2 seem to be highly attractive when optimizing city coverage. On the other hand, depots 3 and 5 are unused in all three

<sup>8</sup> Pollution data and real-time tram locations are online at <http://data.opensense.ethz.ch/position.html>.



**Fig. 12** Impact of the checkpoint threshold  $\theta$  on the achievable coverage with  $\beta = 0.8$  without checkpointing and with X-CP and R-CP constraints for the three validity functions and four different

algorithms: evolutionary algorithm (EA), random search (RS), simulated annealing (SA), and greedy (GR)

configurations. The main reason for this behavior is that both depots serve tram lines that also serve destinations outside the city and thus during these times do not contribute to the city coverage. Depot 1 and 2 seem to be attractive for their central locations and thus serving very diverse set of lines.

In Fig. 11 we show the impact of the preference parameter  $\beta$  on the expected city coverage. We evaluate the coverage according to Eq. 10 by generating a discretization of the area of interest  $\Omega$  with Monte Carlo sampling and computing the level of coverage at every sampled location. For all three validity functions, the city coverage increases with the rising of the preference factor  $\beta$ . The best coverage is achieved if each equipped tram always follows the assigned line number, i.e., for  $\beta = 1$ . In our case, we expect to have  $\beta$  around 0.7–0.8, which would lead to a reduction in coverage of approximately 20 %, compared to the best case. Notice that our evolutionary algorithm outperforms all three approaches used for comparison. Moreover, for the case when no specific line can be favored within a depot ( $\beta = 0$ ), both random search and simulated annealing show difficulties in finding a solution that satisfies the X-checkpointing constraint and thus result in zero coverage. Furthermore, for all three considered validity

functions, the higher the preference likelihood  $\beta$  is, the more control one has over the system with respect to checkpointing constraints and the higher coverage can be achieved.

Finally, we evaluate the influence of the checkpoint threshold  $\theta$  on the achievable coverage with and without checkpointing constraints. In Fig. 12 we show the comparison of our evolutionary algorithm with random search, simulated annealing, and greedy under X- and R-checkpointing constraints and without any constraints for the three validity functions. We vary the values of the checkpoint threshold  $\theta$  between 0.95 and 0.8. Note that the case  $\theta = 1$  is out of interest, since all feasible solutions would require  $\beta = 1$  and force the algorithm to favor the same line for all trams. We achieve the highest coverage without checkpointing constraints, independent of the used validity function. This is because checkpointing introduces an additional constraint (i.e., requires a connected checkpoint graph) that excludes some solutions, which potentially lead to a high coverage. No matter which algorithm is used, the impact of checkpointing on the achievable coverage depends on the threshold  $\theta$ . For threshold values below 0.8 the impact is negligible. This is due to the excellent connectivity of transport networks in general, and tram

network in the city of Zurich in particular, as it can be seen in Fig. 9. Our experiments show that for high values of  $\beta > 0.8$ , 10 randomly selected trams build a connected checkpoint graph with 70 % probability. With increasing threshold requirements the obtainable coverage declines.

Our evolutionary algorithm clearly finds better solutions than the other three approaches. The relative improvement heavily depends on the validity function. The greedy algorithm performs comparable if only coverage is optimized. In this case the algorithm provides approximation guarantees due to submodularity property of the coverage function, as shown in (Krause 2008). However, the performance of the algorithm considerably drops as checkpointing constraints are introduced. In our implementation, greedy checks in each steps that the checkpoint constraint is fulfilled, which clearly reduces the achieved coverage, since only the final solution is obligated to satisfy this constraint. Greedy is unable to find a valid solution if the constraint is only tested at the end.

## 6 Related work

The work related to our approach can be divided in three groups: solutions to the area coverage problem with static sensors, route selection and planning algorithms in the context of area coverage, and route finding in timetable networks.

The problem of area coverage with sensors is often considered without sensor checkpointing in static settings. Related work on the topic includes solutions to the coverage problem as is (So and Ye 2005), combined with event detection (Ganguli et al. 2005), and motion planning of mobile agents to achieve area coverage (Cortes et al. 2004; Hussein and Stipanovic 2006; Wang and Hussein 2010). There is little work on checkpoint design in this context.

Towards checkpoint design, in (Xing et al. 2008) the authors present an approach for saving energy in wireless sensor networks by introducing a mobile base station and designing a set of rendezvous points for data collection. Further related approaches on the design of movement patterns for base stations can be found in (Jea et al. 2005; Somasundara et al. 2006; Gu et al. 2006; Basagni et al. 2011; Turgut and Bölöni 2011). The following two properties distinguish our approach from the above solutions: (1) the underlying timetable network provides a *fixed* backbone and is a considerable limitation in terms of coverage; (2) all nodes in the network are mobile and thus time-dependent.

Similarly to our scenario, timetable networks consist of plenty of mobile nodes. Routing on timetable networks is currently a hot topic in the respecting community. In particular, the interesting problems are earliest arrival and

minimum number of transfers when planing a route from A to B (Pyrga et al. 2008). The main difficulty here is the computational overhead due to lack of hierarchical structure in timetable networks. Both problems are concerned with an efficient design of checkpoints. This closely reflects the problem we face when designing a connected timetable subnetwork. In contrast to route planning, we are rather interested in selecting a subnetwork with very short transfer times, and name these interchange points checkpoints.

We are not aware of any approach solving the coverage problem atop a timetable network with an additional checkpointing requirement.

In (Krause and Guestrin 2009; Krause et al. 2009), the authors show that the problem of selecting sensing locations in order to stay informed at minimum cost exhibits the property of *submodularity*. Thus, a greedy algorithm solves the problem optimally up to a constant factor.

## 7 Conclusions

In this paper we consider the problem of selecting a set of vehicles of a public transport network to maximize the coverage of a city. Since individual mobile vehicles are usually not assigned to specific lines and timetables, but rather to their host depots, each vehicle can take any route served by its host depot.

When selecting vehicles for the installation of sensor nodes, it is essential to have sensor checkpoints to be able to detect sensor malfunction or recalibration needs. Upon the availability of a set of reference stations, the quality of the selected subnetwork can be further improved by considering the closeness to these reference stations. We solve both problems with an evolutionary algorithm since computing the optimal solution is only feasible for very small transport networks. Additionally, we show how to consider route selection uncertainties when calculating the expected city coverage and checkpoint probability. Our algorithm outperforms all three related approaches: simulated annealing, random search and greedy algorithm.

We evaluate our approach under different parameter settings for the tram network of Zurich. Our results show that due to the introduced uncertainties the achievable city coverage decreases by up to 20 %. Further we show that due to the high connectivity of the public transport network, the probability that the selected set of trams form a connected network is over 70 %.

The obtained results to the route selection problem are used in the OpenSense project to select vehicles for the installation of our air quality sensing stations.

**Acknowledgments** We would like to thank the anonymous reviewers for their valuable comments and suggestions. Further, we

thank Christoph Walser for his technical support and VBZ for their support and collaboration. This work was funded by NanoTera.ch with Swiss Confederation financing.

## References

- Aberer K, Sathe S, Chakraborty D, Martinoli A, Barrenetxea G, Faltings B, Thiele L (2010) Opensense: open community driven sensing of environment. In: Proceedings of the 1st International Workshop on GeoStreaming (IWGS), ACM, pp 39–42
- Alon N (1995) A note on network reliability. In: Discrete probability and algorithms. Springer, Berlin, pp 72:11–14
- Back T, Fogel DB, Michalewicz Z (1997) Handbook of evolutionary computation, 1st edn. IOP Publishing Ltd, Bristol
- Basagni S, Carosi A, Petrioli C, Phillips CA (2011) Coordinated and controlled mobility of multiple sinks for maximizing the lifetime of wireless sensor networks. *Wirel Netw* 17:759–778
- Boscolo A, Mangiacavalli C (1998) Pollution dynamic monitoring system for city air quality control. In: Proceedings of the international conference on instrumentation and measurement technology (I2MTC), IEEE, vol 1, pp 202–205
- Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the 7th international conference on world wide web (WWW), Elsevier, Amsterdam, pp 107–117
- Buchli B, Yucecel M, Lim R, Gsell T, Beutel J (2011) Feature-rich platform for WSN design space exploration. In: Proceedings of the 10th international conference on information processing in sensor networks (IPSN), ACM/IEEE, pp 115–116
- Carullo A, Corbellini S, Grassini S (2007) A remotely controlled calibrator for chemical pollutant measuring-units. *IEEE Trans Instrum Measur* 56:1212–1218
- Choi S, Kim N, Cha H, Ha R (2009) Micro sensor node for air pollutant monitoring: hardware and software issues. *MDPI Sensors* 9:7970–7987
- Cortes J, Martinez S, Karatas T, Bullo F (2004) Coverage control for mobile sensing networks. *IEEE Trans Robot Autom* 20:243–255
- De Vito S, Massera E, Piga M, Martinotto L, Di Francia G (2008) On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sens Actuators B: Chem* 129:750–757
- Dutta P, Aoki P, Kumar N, Mainwaring A, Myers C, Willett W, Woodruff A (2009) Common sense: participatory urban sensing using a network of handheld air quality monitors. In: Proceedings of the 7th international conference on embedded networked sensor systems (SenSys), ACM, pp 349–350
- Fierz M, Houle C, Steigmeier P, Burtcher H (2011) Design, calibration, and field performance of a miniature diffusion size classifier. *Aerosol Sci Technol* 45:1–10
- Ganguli A, Susca S, Martinez S, Bullo F, Cortes J (2005) On collective motion in sensor networks: sample problems and distributed algorithms. In: Proceedings of the 44th international conference on decision and control (CDC), IEEE, pp 4239–4244
- Gil-Castiñeira F, González-Castaño F, Duro R, Lopez-Peña F (2008) Urban pollution monitoring through opportunistic mobile sensor networks based on public transport. In: Proceedings of the international conference on computational intelligence for measurement systems and applications (CIMSA), IEEE, pp 70–74
- Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading
- Gu Y, Bozdogan D, Ekici E (2006) Mobile element based differentiated message delivery in wireless sensor networks. In: Proceedings of the international symposium on world of wireless, mobile and multimedia networks (WOWMOM), IEEE, pp 83–92
- Harik G (1995) Finding multimodal solutions using restricted tournament selection. In: Proceedings of the 6th international conference on genetic algorithms (ICGA), Morgan Kaufmann, Los Altos, pp 24–31
- Hasenfratz D, Saukh O, Sturzenegger S, Thiele L (2012a) Participatory air pollution monitoring using smartphones. In: Proceedings of the 2nd international workshop on mobile sensing, ACM, New York
- Hasenfratz D, Saukh O, Thiele L (2012b) On-the-fly calibration of low-cost gas sensors. Proceedings of the 9th European Conference on Wireless Sensor Networks (EWSN) pp 228–244
- Honicky R, Brewer E, Paulos E, White R (2008) N-smarts: networked suite of mobile atmospheric real-time sensors. In: Proceedings of the 2nd international workshop on networked systems for developing regions (NSDR), ACM, 25–30
- Hussein I, Stipanovic D (2006) Effective coverage control using dynamic sensor networks. In: Proceedings of the 45th international conference decision and control (DAC), IEEE, pp 2747–2752
- Jea D, Somasundara A, Srivastava M (2005) Multiple controlled mobile elements (data mules) for data collection in sensor networks. In: Proceedings of the 1st international conference on distributed computing in sensor systems (DCOSS), Springer LNCS, pp 244–257
- Kamionka M, Breuil P, Pijolat C (2006) Calibration of a multivariate gas sensing device for atmospheric pollution measurement. *Sens Actuators B: Chem* 118:323–327
- Karger DR (1995) A randomized fully polynomial time approximation scheme for the all terminal network reliability problem. In: Proceedings of the 27th annual symposium on theory of computing (STOC), ACM, pp 11–17
- Krause A (2008) Optimizing sensing: theory and applications. PhD thesis, Carnegie Mellon University
- Krause A, Guestrin C (2009) Optimizing sensing: from water to the web. *IEEE Comput Archit Lett* 42:38–45
- Krause A, Rajagopal R, Gupta A, Guestrin C (2009) Simultaneous placement and scheduling of sensors. In: Proceedings of the 8th international conference on information processing in sensor networks (IPSN), ACM/IEEE, pp 181–192
- Pyrga E, Schulz F, Wagner D, Zaroliagis C (2008) Efficient models for timetable information in public transportation systems. *Exp Algorithm (JEA)* 12:2–4
- Saukh O, Hasenfratz D, Noori A, Ulrich T, Thiele L (2012) Route selection for mobile sensors with checkpointing constraints. In: Proceedings of the 8th international workshop on sensor networks and systems for pervasive computing (PerSeNS), IEEE, pp 2–4
- Skiścim C, Golden B (1983) Optimization by simulated annealing: A preliminary computational study for the tsp. In: Proceedings of the 15th conference on winter simulation, IEEE, pp 523–535
- So A, Ye Y (2005) On solving coverage problems in a wireless sensor network using voronoi diagrams. *Internet Netw Econ* 3828:584–593
- Somasundara A, Kansal A, Jea D, Estrin D, Srivastava M (2006) Controllably mobile infrastructure for low energy embedded networks. *IEEE Trans Mobile Comput* 5:958–973
- Turgut D, Bölöni L (2011) Heuristic approaches for transmission scheduling in sensor networks with multiple mobile sinks. *Comput J* 54:332–344
- Vardoulakis S, Fisher B, Pericleous K, Gonzalez-Flesca N (2003) Modelling air quality in street canyons: a review. *Atmos Environ* 37:155–182
- Völgyesi P, Nadas A, Koutsoukos X, Lédeczi Á (2008) Air quality monitoring with sensormap. In: Proceedings of the 8th international conference on information processing in sensor networks (IPSN), ACM/IEEE, pp 529–530

- Wang Y, Hussein I (2010) Awareness coverage control over large-scale domains with intermittent communications. *IEEE Trans Autom Control* 55:1850–1859
- Xing G, Wang T, Jia W, Li M (2008) Rendezvous design algorithms for wireless sensor networks with a mobile base station. In: *Proceedings of the 9th international symposium on mobile ad hoc networking and computing (MobiHoc)*, ACM, pp 231–240
- Yamazoe N, Miura N (1995) Development of gas sensors for environmental protection. *IEEE Trans Compon Packag Manuf Technol* 18:252–256