On the $k$ Nearest-Neighbor Path Distance from the Typical Intersection in the Manhattan Poisson Line Cox Process

Konstantinos Koufos, Harpreet S. Dhillon, Mehrdad Dianati and Carl P. Dettmann

Abstract—In this paper, we consider a Cox point process driven by the Manhattan Poisson line process. We calculate the exact cumulative distribution function (CDF) of the path distance (L1 norm) between a randomly selected intersection and the $k$-th nearest node of the Cox process. The CDF is expressed as a sum over the integer partition function $p(k)$, which allows us to numerically evaluate the CDF in a simple manner for practical values of $k$. These distance distributions can be used to study the $k$-coverage of broadcast signals transmitted from a road side unit (RSU) located at an intersection in intelligent transport systems (ITS). Also, they can be insightful for network dimensioning in vehicle-to-everything (V2X) systems, because they can yield the exact distribution of network load within a cell, provided that the RSU is placed at an intersection. Finally, they can find useful applications in other branches of science like spatial databases, emergency response planning, and districting.

Index Terms—Manhattan Poisson line Cox process, spatial databases, stochastic geometry, vehicular networks.

I. INTRODUCTION

The development of the road network is a key component of urban planning because it greatly affects commuting efficiency, districting, emergency response dispatching, and first-aid services, to name but a few. Since the recent advent of wireless connectivity for pedestrians, and the ongoing proliferation of connected vehicles through vehicle-to-everything (V2X) systems, the road network is also the setting for several location-based e-services [1]. Exemplar applications could be electric vehicles querying over the internet for the nearest charging stations, and/or pedestrians searching with their smartphones for the closest available taxis [2]–[5].

A. Modeling road networks

The simplest models for urban road networks utilize just a set of vertices and edges [6], [7]. The vertices may represent junctions, the start/end points of roadways, critical locations where the speed limit or the travel direction changes, etc. Naturally, two vertices are connected by an edge if there is a straight link between them, giving rise to the adjacency matrix of the road network. The adjacency matrix is not necessarily binary. The graph representation can be enhanced by assigning weights to the edges, which might be proportional to the (average or minimum) travel time, fuel cost, etc., along the road segment(s) that the edge represents. Algorithms exploring the graph have been also implemented, e.g., the best-first search to identify the nearest neighbors from a vertex and the Dijkstra’s algorithm to find the shortest paths, i.e., the sequence of edges of minimal aggregate cost between two non-adjacent vertices [8].

Another line of research in graph-based models, which is particularly useful for emergency response planning, assumes that the edge weights are proportional to the length of the associated streets, and models random events along the edges of the graph. If these events represent points of emergency, the graph distance distributions, investigated in [9], can reveal the intrinsic properties of the response system we need to build to combat all emergencies effectively. For instance, they can be used to infer the number of ambulances, medical personnel, etc. we have to deploy.

While certainly important, the graph-based approaches apply to specific road networks. Even though different cities can share common features and network graph properties [10], the graph-based models provide a limited level of abstraction. Besides, due to the high complexity of graph-based routines, it is often impossible to model the road network very precisely. Also, the graph representation cannot answer questions pertinent to network planning, e.g., what is the minimum required intensity of charging stations, so that two of them are within a driving distance of one kilometer from a randomly chosen road intersection, with probability at least 90%? This paper aims to bridge this gap. We argue that the mathematical tools of point and line processes, as well as the stochastic geometry, see [11] for an introduction in the field, widely and successfully utilized during the past 15 years in the performance evaluation of random wireless communication networks, can also be insightful for urban road planning.

Unlike the graph-based methodology, we follow the general stochastic geometry framework, which does not consider a particular topology for the road network. Only the intensity of streets is available (or can be estimated). In addition, we assume that: (i) The road layout has a relatively regular structure, and thus, a Manhattan Poisson line process (MPLP) is a realistic model for the urban road network, and (ii) the locations of points of interest (POI), e.g., vacant cabs at a specific time, events triggering police action, etc. follow a homogeneous Poisson point process (PPP) along each street. Under these assumptions, i.e., a Manhattan Poisson line Cox...
process (MPLCP) for the locations of POI, we will derive the path distance (L1 norm) distribution of the $k$ nearest POI from a randomly selected intersection.

The path distance distributions can provide a quick insight into the probabilistic relationship between the area of a Manhattan cell and the number of random events inside it. We have identified three potential applications, namely spatial database queries, districting, and urban vehicular ad hoc networks (VANETs), where the $k$ nearest-neighbors (kNN) path distance distributions would be of use. We elaborate on these areas next.

**B. Motivation and prior art**

In the kNN query, a spatial database returns the locations of the $k$ nearest objects, in terms of network distance, to the query point (or agent) [2]. Consider, for instance, a driver looking for the $k$ nearest hotels (static objects) in terms of travel time, or a pedestrian querying for the $k$ nearest vacant cabs (mobile objects). The agent reports its location, and the database solves the query using, e.g., a graph representation for the road network [3, Fig. 2].

Even with static objects, the graph dynamically changes due to varying traffic conditions, and the computational complexity can quickly explode, especially with frequent queries from mobile agents [4]. The server must continuously track and update the locations of the $k$ nearest objects. Because of that, neglecting the constraints imposed by the road network, and using just Minkowski distances to solve the kNN problem, especially for group queries, has not been abandoned [5]. The study in [2] has already pointed out that the Euclidean distance is a lower bound to the network distance, and thus, we could use it to prune the search space in kNN queries. However, pruning based on a lower bound is not always effective. Therefore the calculation of the exact path distance distributions, as we will do in this paper, will be very helpful.

Apart from spatial databases, the kNN distance distributions can also be used in the planning of dispatching policies for emergency response services and districting [12]. In balanced district design, the road network of a metropolitan area is partitioned into smaller units (territories) which contain about the same expected number of road accidents so that the workload is divided equally among police departments [12]. Given the size of the districts, the kNN distributions, we will develop in this paper, can be used to calculate the probability that a police department can cover the $k$ nearest emergencies with probabilistic target time guarantees.

In wireless communication research, line processes have already been used for the performance evaluation of vehicular networks [13]–[16]. This is motivated by the ongoing standardization activities of V2X communication, e.g., the transmission of cooperative awareness messages from the vehicles to the infrastructure [17], and the response of the latter with the collective perception message [18]. The one- and two-dimensional PPPs are valid models for urban VANETs, only in the high- and the low-reliability regime respectively [19]. As a result, the optimal transmission probability in VANETs modeled by Cox point processes is in general different than that calculated using the PPP [16]. To draw solid conclusions about the network performance at any operation threshold, the road intersections must be modeled explicitly. For motorway VANETs, on the other hand, the superposition of one-dimensional (1D) point processes is sufficient [20]–[22].

The studies in [13]–[15] have suggested using a Poisson line process to capture the random orientation of streets in urban environments, and stationary 1D PPPs to model the locations of vehicles per street. In the resulting Poisson line Cox process $\Phi_c$, which is a generalization of the MPLCP, the study in [13] has evaluated the distance distribution between an arbitrary point in the plane and the nearest point of $\Phi_c$. This is essentially the distribution of serving distance in cellular vehicular networks with nearest base station association, where the locations of base stations follow the two-dimensional PPP [14]. The study in [15] has investigated the performance of VANETs modeled by the Cox process $\Phi_c$. It has derived the coverage probability for the typical receiver and pointed out the conflicting effect of the intensities of the roads and vehicles on the coverage probability. It has also solved for the distance distribution between the typical vehicle and the nearest vehicle of $\Phi_c$. Finally, the study in [23] has used a MPLCP model for the locations of base stations and calculated their distance distribution to the origin.

Unfortunately, the above studies have measured the distances in the Euclidean (L2 norm) sense, even though the attenuation of wireless signals, especially in millimeter-wave frequencies, is better described by a street canyon model [24, Eq. (1)]. In this regard, the kNN Manhattan distance distributions will be useful in investigating the $k$-coverage of wireless signals, diffracted around buildings at road intersections as they propagate. We will use them to identify, e.g., how many vehicles within half a kilometer from an intersection can successfully receive broadcast safety messages with probability at least $q$%? Thus far, the kNN distributions have been identified for Poisson and binomial processes see [25]–[27], without considering the deployment constraints due to the road layout. For $k = 1$, more general convex geometries like the $n$-sided polygon have been also investigated [28].

**C. Contributions**

The probability distribution function (PDF) of the shortest path between a random intersection and a point of the MPLCP has been recently calculated in [29]. In this paper, we will generalize this result to $k \geq 1$ nearest points. We present various methods to compute the distance distributions, and finally, we cast the solution as a sum over the integer partitions of $k$. The computational complexity of the suggested numerical algorithm is low, and the cumulative distribution functions (CDFs) can be easily obtained for all practical values of $k$. Finally, it should be noted that the CDF of the distance between a random intersection and the $k$-th nearest point of the MPLCP can serve as a lower bound to the CDF of the distance between a random position of the road and its $k$-th nearest point of the MPLCP. For $k = 1$ both CDFs have been computed in [29].

The rest of the paper is organized as follows. In Section II, we formally introduce the MPLCP. In Section III,
we calculate the PDF of the total length $L_t$ of line segments inside a Manhattan square, centered at a randomly selected road intersection. In Section IV, we calculate the moment generating function (MGF) of the random variable (RV) $L_t$, and in Section V we present a numerical algorithm which can be used to compute the CDF of the distance between an intersection and the $k$-th nearest point of the MPLCP. In Section VI, we validate the suggested algorithm against simulations, and finally, in Section VII, we conclude this study.

II. SYSTEM MODEL AND NOTATION

A line process, in layman’s terms, is just a random collection of lines. If we limit our attention to undirected lines in the Euclidean plane, each line $l_i$ can be uniquely determined by the following parameters: the length $\rho_i \geq 0$ and the angle $\phi_i \in [-\pi, \pi]$, measured counter-clockwise, of the line segment being perpendicular to $l_i$ and passing though the origin [11, Chapter 8.2.2]. Therefore a line process can be associated with a point process, and vice versa, where the line $l_i$ is uniquely mapped to the point $x_i \in \mathbb{R}^2$ with polar coordinates $(\rho_i, \phi_i)$. The associated point process is often called the representation space of the line process.

Let us now consider the realizations of two independent 1D PPPs of equal intensity $\lambda$, along the $x$ and $y$ axes, and construct the resulting realization of lines. Obviously, all points on the $x$ axis give rise to vertical lines $\phi_i \in \{0, \pi\}$, and all points along the $y$ axis correspond to horizontal lines $\phi_i \in \{-\pi/2, \pi/2\}$. This is known as the Manhattan Poisson line process (MPLP) and consists only of vertical and horizontal lines. It is stationary and motion-invariant owing to the stationarity and motion-invariance of the PPP in the representation space. Its intensity, defined as the mean total length of lines per unit area, is equal to $2\lambda$ [11, Chapter 8.1].

Due to the fact that the contact distribution of the 1D PPP is exponential, the distances between neighboring intersections along a line of a MPLP follow the exponential distribution. Therefore the $k$-th nearest facility to a point process is the $k$-th nearest facility to a line process.

The CDF for the path distance of the $k$-th nearest facility to an intersection is denoted by $F_{R_k}(r) = P(R_k \leq r) = 1 - P(R_k > r)$. The complementary CDF, $P(R_k > r)$, is equal to the sum of the probabilities $P_j, j \in \{0, 1, 2, \ldots (k-1)\}$, hence, $F_{R_k}(r) = 1 - \sum_{j=0}^{k-1} P_j$, where $P_j$ is the probability that there are exactly $j$ facilities inside the square, see Fig. 1, which is the locus of points with equal Manhattan distance $r$ to the origin. The set of all points inside the square, including the sides, is denoted by $B(r) \equiv B$ for brevity.

The CDF of the RV $R_1$ has been derived in [29, Theorem 1]

$$
F_{R_1}(r) = 1 - P_0 - e^{-2\lambda \rho} e^{-4\lambda \rho (1 - a_0)}
$$

where $a_0 = \frac{1 - e^{-2\lambda \rho}}{2\lambda \rho}$ and $P_0 = e^{-4\lambda \rho} e^{-4\lambda \rho (1 - a_0)}$ is the probability that no facility lies in $B$.

In addition, we define the RV $N_{\rho} (\Phi \cap B)$ which counts the number of points of the Cox process, driven by the line process $\Phi$, within $B$. The total number of lines intersecting $B$, excluding the typical lines $L_x, L_y$ is denoted by the RV $N$. Furthermore, the RV $L_t \geq 0$ describes the random length of the

![Fig. 1. Example realization of a Cox point process driven by MPLP, modeling the spatial distribution of facilities in a city.](image)
i-th line $\ell_i$ intersecting $B$, and the RV $L_t \geq 4r$ describes the total length of line segments in $B$ including the contribution, $4r$, due to the typical lines. Finally, the realizations of the RVs $L_i$ and $L_t$ are both denoted by $l$.

### III. Calculating $P_k$ Using the Distribution of $L_t$

Given the realization $l$ of the RV $L_t$, the number of facilities in $B$ follows a Poisson distribution with parameter $\lambda_l$, Po($\lambda_l$) with $i \in \mathbb{N}$ being the integer, where the probability mass function of the Poisson distribution will be evaluated. As a result, based on the law of total expectation, the probability $P_k$ that there are $k$ facilities in $B$ can be obtained by averaging the probability mass function of the Poisson RV Po($\lambda_l$) over the PDF, $f_{L_t}(l)$, of the total length of line segments $L_t$ in $B$. Hence,

$$P_k = \int_{4r}^{\infty} e^{-\lambda_l} \frac{(\lambda_l)^k}{k!} f_{L_t}(l) \, dl,$$

where the lower integration limit equals $4r$, because $B$ always contains the segments due to the typical lines $L_x, L_y$ of $\Phi_L$.

In order to derive the PDF $f_{L_t}(l)$, we start with the random number $N$ of line segments intersecting $B$, which follows the Poisson distribution with parameter $4\lambda r$, $N \sim$ Po($n, 4\lambda r$). Recall that $\lambda$ is the density of intersection points along a line, and $2r$ is the length of the diagonal of $B$. Conditionally on $N \geq 1$, the abscissas (ordinates) of the line segments parallel to $L_x$ (or $L_y$) are distributed uniformly at random in $(-r, r)$. As a result, the distribution of the RV $L_t$ describing the length of the $i$-th line segment in $B$ is uniform too. In order to derive its CDF, we note that $L_t$ takes values in $(0, 2r)$ and thus, $\mathbb{P}(L_t \leq l) = \frac{l}{2r}$, $l \in (0, 2r)$. For instance, the vertical line with abscissa $z_1$ in Fig. 1 has length $l = 2(r - z_1)$ in $B$.

Conditionally on the realization $n \geq 1$ for the RV $N$, the total length of line segments in $B$, $\sum_{i=1}^{n} L_i$, is equal to the sum of $n$ independent and identically distributed (i.i.d.) uniform RVs in $(0, 2r)$. As a result, the sum of RVs $\sum_{i=1}^{n} L_i$ follows the Irwin-Hall distribution with PDF

$$\frac{n}{2r} \sum_{i=1}^{n} \frac{1}{(n-1)!} \sum_{k=0}^{\lfloor \frac{l}{2r} \rfloor} (-1)^k \binom{n}{k} \left( \frac{l}{2r} - k \right)^{n-1}, \quad (3)$$

where $n \geq 1$ and $l \geq 0$.

In order to compute the PDF of $L_t$, we need to average equation (3) over the Poisson distributed number $N$ for $n \geq 1$. The case $N = 0$, i.e., no intersections along $\{L_x \cup L_y\} \cap B$, which occurs with probability $e^{-4\lambda r}$, leads to $L_t = 4r$ and it is treated separately below.

$$f_{L_t}(l) = e^{-4\lambda r} \delta_{l, 4r} + \mathbb{E}_{n \geq 1} \left\{ \frac{1}{2r(n-1)!} \sum_{k=0}^{\lfloor \frac{l}{2r} \rfloor} (-1)^k \binom{n}{k} \left( \frac{l}{2r} - k \right)^{n-1} \right\}$$

$$= e^{-4\lambda r} \delta_{l, 4r} + \sum_{n=1}^{\infty} \frac{(4\lambda r)^n e^{-4\lambda r}}{n!} \frac{1}{2r(n-1)!} \times \sum_{k=0}^{\lfloor \frac{l}{2r} \rfloor} (-1)^k \binom{n}{k} \left( \frac{l}{2r} - k \right)^{n-1}, \quad l \geq 4r,$$

where $\delta_{x,y} = 1$ for $x = y$ and $\delta_{x,y} = 0$ otherwise, is the Kronecker delta function, and also note that equation (3) has been shifted to the right by $4r$.

The above expression can be simplified, to some extent, by interchanging the order of summations, and while doing so, carefully setting the lower limit of the sum with respect to $n$.

$$f_{L_t}(l) = e^{-4\lambda r} \delta_{l, 4r} + e^{-4\lambda r} \sum_{n=1}^{\infty} \frac{(4\lambda r)^n}{n!} \frac{1}{2r(n-1)!} \left( \frac{l}{2r} - k \right)^{n-1}$$

$$= e^{-4\lambda r} \delta_{l, 4r} + e^{-4\lambda r} \sum_{n=1}^{\infty} \frac{(4\lambda r)^n}{n!} \frac{1}{(n-1)!} \left( \frac{l}{2r} - k \right)^{n-1}$$

$$= e^{-4\lambda r} \delta_{l, 4r} + e^{-4\lambda r} \sum_{k=1}^{\lfloor \frac{l}{2r} \rfloor} \frac{(4\lambda r)^n}{n!} \frac{1}{(n-1)!} \left( \frac{l}{2r} - k \right)^{n-1}$$

$$= \frac{1}{4r} \sum_{k=1}^{\lfloor \frac{l}{2r} \rfloor} e^{-4\lambda r} \sum_{n=1}^{\infty} \frac{(4\lambda r)^n}{n!} \frac{1}{(n-1)!} \left( \frac{l}{2r} - k \right)^{n-1}$$

where $l \geq 4r$ and $aF_1(\alpha, z) = \sum_{k=0}^{\infty} \frac{1}{(\alpha+k) \Gamma(\alpha)} z^k$ is the regularized hypergeometric function; example validations of equation (5) are depicted in Fig. 2.

The expression in the last line of equation (5) is quite complicated to use it in that form in the integral in (2), hence, calling for another approach to evaluate the probabilities $P_k$.

### IV. Calculating $P_k$ Using MGFs

Since the PDF of the RV $L_t$ has a complicated form, we may instead work with its MGF, $M_{L_t}(t) = \mathbb{E}\{ e^{tL_t} \}, t \in \mathbb{R}$. Actually the MGF of $L_t$ can be computed using the properties of the compound Poisson distribution. Recall that the RV $L_t$ is equal to the sum of $N \sim$ Po($n, 4\lambda r$) i.i.d. uniform RVs $L_i \equiv L$ in $(0, 2r)$ plus the constant $4r$. Therefore,

$$\mathbb{E}\{ e^{tL_t} \} = \mathbb{E}_N \{ \mathbb{E}\{ e^{tL_i} | N \} \}$$

$$= \mathbb{E}_N \{ e^{4rt} M_L(t)^N \}$$

$$\approx e^{4rt} \mathbb{E}_N \left\{ e^{2rt - 1} \right\}^N \quad (6)$$

$$\approx \exp(4rt + 4\lambda r \left( \frac{e^{2rt} - 1}{2rt} - 1 \right))$$

where (a) follows from the MGF of a uniform RV, and (b) uses the probability generating function of a Poisson RV.

The limit of the first derivative of $M_{L_t}(t)$ with respect to $t$ at $t \to 0$ in equation (6) yields $(4r + 4\lambda r^2)$, which is the mean of the RV $L_t$. The second term in the parenthesis is, as expected, equal to the product of the mean length $r$ of a randomly selected line segment $L_t$ multiplied by the expected number $4\lambda r$ of line segments in $B$.

Conditionally on the realization of the length $L_t = l$, the number of facilities in $B$ is Poisson distributed with parameter $\lambda_l$. As a result, using the law of total expectation, the
MGF of the (discrete) RV of the number of facilities in $B$, $N_p(\Phi_L \cap B)$, can be read as

\[
M_{N_p(\Phi_L \cap B)}(t) = \mathbb{E}\{e^{N_p(\Phi_L \cap B) t}\} = \mathbb{E}_L\{\mathbb{E}\{e^{N_p(\Phi_L \cap B) t} \mid L_t\}\}\]

\[
= \mathbb{E}_L\{e^{N_p(t-1)}\} = M_{L_t}(\lambda_g (e^t-1)),
\]

where $(a)$ is due to the MGF of a Poisson RV.

After substituting the above argument, $\lambda_g (e^t-1)$, into the last equality in (6), we obtain the MGF of the RV $N_p(\Phi_L \cap B)$. $M_{N_p(\Phi_L \cap B)}(t) = \exp\left(4r\lambda_g (e^t-1)+4\lambda r \left(\frac{e^{2r\lambda_g (e^t-1)}-1}{2r\lambda_g (e^t-1)}-1\right)\right)$.

Furthermore, starting from the definition of the MGF of a discrete RV on the natural numbers, $M_{N_p(\Phi_L \cap B)}(t) = \sum_{k=0}^{\infty} P_k e^{kt}$ we get $P_k$

\[
= \frac{1}{ct} \frac{d}{dt} M_{N_p}(\log t) \bigg|_{t=0}
= \frac{1}{ct} \frac{d}{dt} \exp\left(4r\lambda_g (t-1)+4\lambda r \left(\frac{e^{2r\lambda_g (t-1)}-1}{2r\lambda_g (t-1)}-1\right)\right) \bigg|_{t=0}.
\]

After substituting $k=0$ in equation (7), we obtain $P_0 = \exp(-4r\lambda_g - 4\lambda r (1-a_0))$, as expected, see equation (1). For $k=1$ in (7), after some simplification, we have

\[
P_1 = P_0 \left(4r\lambda_g + 4r \left(a_0 - e^{-2r\lambda_g}\right)\right).
\]

The calculation of higher-order derivatives in (7) results in complicated expressions which are difficult to manipulate. For instance, we list below the expressions we get, after some simplification, for $P_2$ and $P_3$.

\[
P_2 = \frac{1}{2} P_0 \left(4r\lambda_g + 4r \left(a_0 - e^{-2r\lambda_g}\right)\right)^2 + 4\lambda r P_0 \times (a_0 - e^{-2r\lambda_g} - r\lambda_g e^{-2r\lambda_g}),
\]

\[
P_3 = \frac{1}{6} P_0 \left(4r\lambda_g + 4r \left(a_0 - e^{-2r\lambda_g}\right)\right)^3 + 4\lambda r P_0 \times (4r\lambda_g + 4r \left(a_0 - e^{-2r\lambda_g}\right) \times (a_0 - e^{-2r\lambda_g} - r\lambda_g e^{-2r\lambda_g}) + 4r \lambda P_0 \times (a_0 - e^{-2r\lambda_g} - r\lambda_g e^{-2r\lambda_g} - \frac{2}{3} \lambda^2 e^{-2r\lambda_g} - r\lambda_g e^{-2r\lambda_g}).
\]

One way to add some structure in the calculation of $P_k$, is to use the Faà di Bruno’s formula, see for instance [31], for the calculation of the $k$-th derivative of a composite function, as is the exponential in (7). Let us define $f(t) = e^t$ and $g(t) = \left(4r\lambda_g (t-1)+4\lambda r \left(\frac{e^{2r\lambda_g (t-1)}-1}{2r\lambda_g (t-1)}-1\right)\right)$. Leveraging on that $f^{(k)}(t) = e^t$, where $f^{(k)}$ denotes the $k$-th derivative, the Faà di Bruno’s formula is simplified to [31, Eq. (2.2)]

\[
\frac{d^k}{dt^k} f^{(g(t))} = e^{g(t)} \sum_{m=1}^{k} B_{k,m} (g'(t), g''(t), \ldots, g^{(k-m+1)}(t)) = e^{g(t)} B_k(g'(t), g''(t), \ldots, g^{(k)}(t)),
\]

where $B_{k,m}$ is the partial and $B_k$ the complete exponential Bell polynomials.

The Bell polynomials emerge in set partitions. For instance, $B_{4,2}(x_1, x_2, x_3, x_4) = 3x_2^2 + 4x_1x_3$, indicates that there are three ways to separate the set $\{x_1, x_2, x_3, x_4\}$ into two subsets of size two, and four ways to separate it into a block of size three and another of size one. Note that the total number of partitions, i.e., seven, is the Stirling number of second kind which, in general, counts the ways to separate an $m$-element set into $k$ disjoint and non-empty subsets, e.g., $S(4, 2) = 7$.

The calculation of the Bell polynomials is widely available in today’s numerical software packages like Mathematica.

Recall from (7) that the $k$-th derivative of the composite function has to be evaluated at $t = 0$. It is also noted that $P_0 = e^{g(0)}$. Combining equations (7) and (10) yields

\[
P_k = \frac{P_0}{k!} \sum_{m=1}^{k} B_{k,m} (g'(0), g''(0), \ldots, g^{(k-m+1)}(0)).
\]

The equation above is insightful to understand why the calculation of $P_3$ in (9) consists of three terms. The first term over there, $(4r\lambda_g + 4r \left(a_0 - e^{-2r\lambda_g}\right))^3$, is essentially equal to $B_{3,3}(g'(0)) = g''(0)^3$. It is also straightforward to verify that the remaining two terms in (9) are equal to $B_{3,2}(g'(0), g''(0)) = 3g'(0)g''(0)$ and $B_{3,1}(g'(0), g''(0), g'''(0)) = g'''(0)$.

In the next section, we will use enumerative combinatorics, revealing a simple numerical algorithm to evaluate $P_k$ for arbitrary $k$, without involving higher-order derivatives as in (11).
V. Calculating $P_k$ Using Integer Partitions

Let us assume there are $k$ facilities in $B$ and separate their allocation into two sets: along the typical segments $L_x, L_y$ and in the rest of $B$. The probability $P_k = \mathbb{P}(N_p(\Phi_L \cap B) = k)$ can be read as $P_k = \sum_{i \leq k} \mathbb{P}(N_p((L_x \cup L_y) \cap B) = i \cdot \mathbb{P}(N_p(\Phi_L \cap B) = m)$, \hspace{1cm} (12)

where $(m = k - i)$, and the product of probabilities follows from the independent locations of intersections along $L_x$ and $L_y$, and the independent locations of facilities along each line of $\Phi_L$.

The first probability term in (12) can be calculated as $\mathbb{P}(N_p((L_x \cup L_y) \cap B) = i)$

$$= \mathbb{P}(N_p(L_x \cap B) + N_p(L_y \cap B) = i)$$

$$= \frac{(4\lambda_r r)^i e^{-4\lambda_r r}}{i!} \mathbb{P}(N_p(\Phi_L \cap B) = m)$$ \hspace{1cm} (13)

where $(a)$ uses the fact that the superimposed PPPs of facilities along $L_x$ and $L_y$ is another PPP with twice the intensity $2\lambda_r$.

The calculation of the second probability term in (12) is more involved, but similar to (13), it helps to consider just a single PPP of intersection points with twice the intensity, $2\lambda$, along $L_x$ instead of two line processes $\Phi_{L_1}, \Phi_{L_2}$. Let us denote the resulting distribution of vertical lines by $\Phi'_{L_1}$. Obviously, $\mathbb{P}(N_p(\Phi_{L_1} \cap B) = m) = \mathbb{P}(N_p((\Pi_{L_1} \cup \Pi_{L_2}) \cap B) = m) = \mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m)$. The latter can be written as $\mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m) = \sum_{n=0}^{\infty} \mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | N = n) \cdot \mathbb{P}(N = n)$

$$= \sum_{n=0}^{\infty} \frac{(4\lambda_r r)^n e^{-4\lambda_r r}}{n!} \mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | N = n)$$ \hspace{1cm} (14)

In order to calculate the conditional probability in (14), we have to enumerate the number of ways of allocating $m$ facilities (or objects) into $n$ lines (or urns), with both objects and urns being indistinct. For each possible allocation, we need to obtain its probability of occurrence, and finally we will sum over all obtained values. For $n \geq m$, the number of ways to allocate $m$ objects into $n$ urns is equal to the number of integer partitions of $m$, denoted by $p(m)$, because only the number of objects going to each urn is relevant. For $n < m$, the restricted integer partitions of size at most $n$, $p_n(m)$, have to be considered. Empty urns are obviously allowed. Next, we sum over the probabilities of all partitions.

$$\mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | N = n) = \sum_{\xi \in p(m)} \mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | N = n, \xi)$$

where $p_n(m) = p(m)$ for $n \geq m$ and $\xi$ is the set associated with a random partition, e.g., for the integer partitions $p(3)$, $\xi \in \Xi$ with $\Xi = \{\{3\}, \{2, 1\}, \{1, 1, 1\}\}$ being the set of all partitions.

After substituting the above equality in the last line of (14), and interchanging the orders of summations we end up with

$$\mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m) = \sum_{\xi \in p(m)} \sum_{n = 0}^{\infty} \frac{(4\lambda_r r)^n e^{-4\lambda_r r}}{n!} \mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | N = n, \xi)$$

where the operator $| \cdot |$ denotes the cardinality of the set representing the integer partition. For instance, for the set $\Xi$ above containing all partitions $p(3)$, the cardinalities of the sets $\xi \in \Xi$ are one, two and three respectively.

At this point, it helps to define the parameter $a_k$, $k \in \mathbb{N}$ describing the probability that a vertical line with abscissa $z > 0$, uniformly distributed between the origin and the point $(r, 0)$, contains exactly $k$ facilities in $B$. With reference to Fig. 1, the line with abscissa $z_1$ does not contain any.

$$a_k = \frac{1}{r} \int_0^r \frac{(2\lambda y(r - z))^k e^{-2\lambda y(r - z)}}{(k - 1)\Gamma(k + 1)} dz = \frac{2\lambda y r^k}{\Gamma(k + 1)},$$

where $\Gamma(k + 1) = k!$, $\Gamma(a, x) = \int_0^x t^{a-1}e^{-t}dt$ is the lower incomplete Gamma function, and for $k = 0$ we obtain the parameter $a_0$ defined under equation (1).

Let us consider the partition $\xi = \{1, 1, \ldots, 1\}$ with $m$ 1’s. The inner sum in equation (15), conditionally on this partition, yields $\mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | \xi) = \sum_{n=m}^{\infty} \frac{(4\lambda_r r)^n e^{-4\lambda_r r}}{n!} \cdot \mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | N = n, \xi)$

$$= \sum_{n=m}^{\infty} \frac{(4\lambda_r r)^n e^{-4\lambda_r r}}{n!} \frac{(4\lambda_r r)^k e^{-4\lambda_r r}}{(k - 1)!} \frac{(\lambda r)^m}{m!} = \frac{(4\lambda_r r)^k e^{-4\lambda_r r}}{(k - 1)!} \frac{(\lambda r)^m}{m!},$$

where the binomial coefficient $\binom{n}{m}$ represents the number of ways to select the $m$ lines containing just one facility in $B$.

Substituting the above equation and equation (13) into (12) yields the conditional probability, $P_k|\xi$, of $k$ facilities in $\{\Phi'_{L_1} \cup L_x\} \cap B$ given the partition $\xi$.

$$P_k|\xi = \frac{\sum_{\xi \in p(m)} \frac{(4\lambda_r r)^k e^{-4\lambda_r r}}{(k - 1)!} \frac{(\lambda r)^m}{m!} B_{n}^m}{m!}$$

In a similar manner, we can evaluate $P_k|\xi$ for all $\xi \in p(k)$ and complete the calculation of $P_k = \sum_{\xi \in p(k)} P_k|\xi$ in (12). However, this might be cumbersome, unless a simple pattern is identified. Next, we will derive a simple expression for $P_k|\xi$, depending on the number of times an integer appears in the partition $\xi$. Let us consider the partition $\xi = \{q, \ldots, q, 1, \ldots, 1\}$, where the integer $q$ appears $f$ times and there are also $(m - qf)$ number of 1’s. The inner sum in equation (15), conditionally on the partition $\xi$ with cardinality $|\xi| = (m - qf) + f$, yields $\mathbb{P}(N_p(\Phi'_{L_1} \cap B) = m | \xi)$

$$= \sum_{n=m-(q-1)f}^{m} \frac{(4\lambda_r r)^n e^{-4\lambda_r r}}{n!} a_{qf} a_{1}^{m-qf} a_{0}^{n-(m-(q-1)f) \times} \frac{(\lambda r)^m}{m!} = \frac{(m-(q-1)f)^f a_{q}^{m-qf} (4\lambda_r r)^{m-(q-1)f} e^{-4\lambda_r r}}{(m-(q-1)f)!}$$

where $|\xi| = (m - qf - f)$ is the number of lines containing facilities in $B$. They are selected with $(m-(q-1)f)$ ways from the available $n$ lines, and $(m-(q-1)f)$ is the number of ways to select $f$ out of the segments containing facilities in $B$, and allocate $q$ facilities to each one of them.

Keeping in mind that due to the existence of $f$ replicas of the integer $q$ in the partition, only up to $(k - qf)$ facilities
may be located along \( L_k \), we substitute the above equation and (13) into (12), ending up with

\[
P_{k|x} = \sum_{i=0}^{k-q} \frac{(4\lambda r)^i e^{-4\lambda r}}{i!} \frac{(k-i-(q-1)f)}{f!} q^i a_{k-i-qf}^i \prod_{j=q}^{k} \frac{(4\lambda r a_{k-j+1})}{(k-j)!} P_0.
\]

(17)

It is straightforward to generalize the above calculation to include partitions with more than one \( q > 1 \). Let us assume that the integer \( q_1 > 1 \) appears \( f_1 \geq 1 \) times in the partition \( \xi \). Equation (17) can be generalized as

\[
P_{k|x} = \frac{(4r(\lambda_a + \lambda_1))^k - \sum_i f_i q_i P_0}{(k - \sum f_i q_i)!} \prod_{i=q}^{k} \frac{(4\lambda r a_{k-j})}{(k-j)!} P_0.
\]

(18)

To sum up, in order to evaluate \( P_{k|x} \), we start with \( P_{k|x} = P_0 \). Given that the integer \( q > 1 \) appears in the partition \( f \geq 1 \), we set \( P_{k|x} \leftarrow P_{k|x} \frac{(4\lambda r a_{k-j})}{(k-j)!} P_0 \). For \( q = 1 \), we set \( P_{k|x} \leftarrow P_{k|x} \frac{(4r(\lambda_a + \lambda_1))^k}{(k-q)!} \). We update \( P_{k|x} \) for all integers \( q \in \xi \). Next, we repeat the same procedure for all partitions \( \xi \), and we compute the probability \( P_k \rightarrow \sum_{\xi \in \mu(k)} P_{k|x} \).

For illustration purposes, in Table I, we list the contributions of the seven different terms involved in the calculation of \( P_0 \). The inputs in the rightmost column, which is equal to the product of the terms in the middle column, are generated based on equation (18). For instance, for the partition \( \{3, 2\} \) we have \( f_1 = f_2 = 1 \), because each of the numbers \( q_1 = 2, q_2 = 3 \) appears only once in the partition. Furthermore, \( (f_1 q_1 + f_2 q_2) = 5 \) and thus, the exponent of the term \( (\lambda_a + \lambda_1) \) is zero. Therefore, equation (18) degenerates to the product of just two terms, \( 4\lambda r a_2 \) and \( 4\lambda r a_3 \), scaled by \( P_0 \), and the result for the probability \( P_0 \{3, 2\} \) directly follows.

Now, it is also clarified that in the calculation of \( P_0 \) in equation (9), the first term corresponds to the partition \( \{1, 1, 1\} \) with \( a_1 = a_0 = e^{-2\lambda r} \), the second term to the partition \( \{2, 1\} \) with \( a_2 = a_0 = e^{-2\lambda r} \), and the last term to the partition \( \{3\} \) with \( a_3 = \frac{1}{3} \). Thus, \( P_0 \{3, 2\} \) directly follows.

Algorithm 1 Compute \( P_k \)

1. \( a_q \leftarrow \frac{\Gamma(q+1, 2\lambda r)}{2\lambda r^{q-1}}, q = 0, 1, \ldots (k-1) \)
2. \( P_0 \leftarrow e^{-4\lambda r(1-a_0)} P_k \leftarrow 0 \)
3. \( \Xi = \text{IntegerPartitions}(k) \)
4. for all \( \xi \in \Xi \) do
5. \( P_{k|x} \leftarrow P_0 \)
6. for all \( q \in \xi \) do
7. \( f_q \leftarrow \text{card}_\xi(q) \)
8. if \( q = 1 \) then
9. \( P_{k|x} \leftarrow P_{k|x} \frac{(4(\lambda_a + \lambda_1) r)^q}{f_q} \)
10. else
11. \( P_{k|x} \leftarrow P_{k|x} \frac{(4(\lambda_a r)^q)}{f_q} \)
12. end if
13. end for
14. \( P_k \leftarrow P_k + P_{k|x} \)
15. end for

Fig. 3. The distance distributions for the \( k \)-th nearest facilities to the origin \( k \in \{1, 2, \ldots 10\} \) following a MPLCP with dense streets \( \lambda = 10 \text{ km}^{-1} \) and sparse facilities \( \lambda_a = 0.5 \text{ km}^{-1} \). The red lines are averages over 50 000 simulations and the dashed blue lines are (exact) calculations using Algorithm 1. The simulations are carried out within a square area of 400 km².

VI. NUMERICAL ILLUSTRATIONS & APPLICATIONS

In Fig. 3 we have validated the calculation of the path distance distribution for \( k \leq 10 \). Fig. 4 illustrates that the Euclidean distance (L2 norm) is a bad approximation to the Manhattan distance (L1 norm) distribution for a MPLCP. The approximation quality deteriorates for a larger \( k \). The planar PPP of equal intensity, \( \mu = 2\lambda_a \), where the locations of facilities are not constrained by the road network, is not a better approximation either. Note that for the PPP, the distance to the \( k \)-th nearest neighbor follows the generalized gamma distribution with PDF [32, Theorem 1]:

\[
f_{R_k}(r) = \frac{2 e^{-\mu r^2} \left(\mu r^2\right)^k}{r \Gamma(k)}.
\]

(19)

Having justified that the planar PPP and the Euclidean distances are not good approximations to the path distances, we will next present some case studies where the path distance distributions can be of use.

A. Distance distributions in spatial database queries

Let us consider an electric vehicle at an intersection looking for the nearest charging station. The charging stations might
be closed or fully occupied, depending on the time of day and the road traffic conditions. In that case, the database should return the nearest available charging station to the vehicle.

The distance distributions we have developed in this paper can be used to calculate the path distance distribution and the distribution of travel time to the nearest available facility too. Given that any facility is available with probability \( q \), independently of other facilities, and the average travel speed is \( v \), the CDF of the average travel time to the nearest available facility follows from the geometric distribution:

\[
P(t \leq \tau) = \sum_{i=1}^{\infty} q (1-q)^{i-1} F_R(\tau),
\]

where \( \tau = rv^{-1} \).

See Fig. 5a for the validation of (20). Note that the underlying assumption in equation (20) is that the delay at the intersection and the traffic-related delays are not modeled explicitly but are incorporated into the model through the average velocity \( v \).

C. Urban vehicular communication networks

Turning our interest to wireless communications applications, we assume that a RSU is deployed at the typical intersection and the locations of vehicles follow a MPLCP. The RSU broadcasts messages to the vehicles. For wireless propagation along urban street micro cells, the pathloss model should be different for line- and non-light-of-sight (NLoS) vehicles [33, Fig. 5]. The vehicles with NLoS connections suffer from serious diffraction losses due to the propagation of wireless signals around the corner. In Fig. 6 the distribution of the signal-to-noise ratio (SNR) for the 10 nearest vehicles with NLoS connection to the RSU is depicted. Assuming a distance-based propagation pathloss \( r^{-\eta} \), where \( r \) stands for the Manhattan distance, and a diffraction loss \( \zeta \), it is straightforward to convert the distance distributions into received signal level distributions. Then, it also remains to scale the obtained CDFs by the noise power level \( N_0 \). Specifically, for
the \(k\)-th nearest vehicle with NLoS connection we have

\[
P(SNR_k \leq \theta) = P \left( R_k^{\gamma} \leq \theta N_0 \right) = 1 - P \left( R_k \leq \left( \theta N_0 / \mathcal{L} \right)^{-1/\gamma} \right)
\]

where the vehicles along \(L_x, L_y\) with a line-of-sight connection to the RSU are neglected, hence, (18) degenerates to

\[
P_k|\xi = e^{-4\lambda r(1-\alpha_0)} \cdot \prod_{q_i} \frac{(4\lambda r a_{q_i})^{f_i}}{f_i!}.
\]

Given the size of the cell \(B\), see Fig. 1, it is straightforward to convert the distance distributions for the NLoS vehicles into the distribution of their number inside the cell — network load distribution. Since the NLoS vehicles have much lower link gains than the vehicles along the typical lines \(L_x, L_y\), the RSU must allocate to them more spectral resources under some fair scheduling scheme. Therefore our ability to quickly characterize the network load distribution for NLoS vehicles, see Fig. 7 for an example, is important.

VII. CONCLUSIONS

In this paper, we have devised a low-complexity numerical algorithm to calculate the distribution of the path distance between a randomly selected road intersection and the \(k\)-th nearest node of a Cox point process driven by the Manhattan Poison line process. This algorithm can be used to identify the minimum required density of facilities (modeled by a MPLCP), e.g., charge stations for electric vehicles, to ensure that a vehicle at an intersection can reach the nearest available facility within a target time under a probability constraint. The distance distributions derived in this paper can also be used to calculate the distribution of network load within a cell of a V2X system. It is straightforward to incorporate into our approach path distance distributions towards a specific direction, e.g., south, north-east, etc. In the future, it would be interesting to validate these distributions with real datasets.

REFERENCES


