Urban sensing as a random search process

Kevin O’Keeffe,1 Paolo Santi,1 Brandon Wang,1 and Carlo Ratti1
1Senseable City Lab, Massachusetts Institute of Technology, Cambridge, MA 02139

We study a new random search process: the taxi-drive. The motivation for this process comes from urban sensing, in which sensors are mounted on moving vehicles such as taxis, allowing urban environments to be opportunistically monitored. Inspired by the movements of real taxis, the taxi-drive is composed of both random and regular parts; passengers are brought to randomly chosen locations via deterministic (i.e. shortest paths) routes. We show through a numerical study that this hybrid motion endows the taxi-drive with advantageous spreading properties. In particular, on certain graph topologies it offers reduced cover times compared to persistent random walks.

I. INTRODUCTION

Random search processes [1–4] are a well studied topic with a bounty of practical applications. Examples include the spreading of diseases and rumours [5], gene transcription [2], animal foraging [6–9], immune systems chasing pathogens [10], robotic exploration [11], and transport in disordered media [12, 13]. Early research on random searches focused on symmetric random walks in Euclidean spaces. Over the years however, many variants have been considered, such as persistent random walks [14–16], intermittent random walks [7, 17, 18], and Levy flights [19–22], each of which offer advantages in certain contexts. Topologies other than Euclidean spaces, such as random graphs or real-world networks, have also been studied [23–25].

Here we explore a new random search process: the taxi-drive. As the name suggests, this process models the movement of taxis. The motivation for studying such a process comes from a recent (theoretical) work in urban sensing [26] in which sensors are deployed on taxis, thereby allowing air pollution, road congestion, and other urban phenomena to be monitored ‘parasitically’. As such, this drive-by approach to urban sensing can be viewed as a random search process, in which a city’s environment is ‘sensed’ (i.e. searched) by sensor-bearing taxis as they drive around serving passengers.

Similar to Levy walks [19, 20] or the run-and-tumble motion of bacteria [31, 32], the motion of taxis is part-random, part-regular: passenger destinations are chosen randomly, but the routes taken to those destinations are (approximately) deterministic. This mix of ballistic and random motion makes the spreading properties of taxis unusual; as shown in [26] – and reproduced in Figure 2(a) – the stationary distribution of the taxi-drive process on real-world street networks follow Zipf’s law, in agreement with large, real-world taxi data from nine cities worldwide. This behavior is unusual because it differs significantly from that of classic random search processes such as the random walk, which, as shown in Figure 2(b), produces stationary distributions on the same street networks which are skewed and unimodal.

The purpose of this work is to further explore the taxi-drive process. We do not study its ability to capture real-world data; instead, our goal is theoretical: to study the taxi-drive as a stochastic process. We focus on cover times, which we numerically compute and compare to those of other well known stochastic processes, namely the persistent random walk, and the regular random walk. We hope our paper inspires further work on the taxi-drive process, as well as more theoretical interest in urban sensing.

II. MODEL

Model definition. Consider a street network S whose edges represent street segments, and whose nodes represent road intersections. Under the assumption that passengers can only be picked up and dropped off at intersections, nodes represent also possible passenger pickup and dropoff locations. The taxi-drive runs on S, and as depicted in Figure 1 is defined by three steps:

1. A taxi picks up a passenger at node A, who wishes to travel to a randomly chosen node B.
2. The taxi travels along the shortest path from node A to node B at unit speed. In the event of multiple shortest paths between A and B, one is chosen at random.
3. The process then repeats.

In order to capture the behavior of real taxis, the desi-
Random walk

Taxi drive

Segment popularities

FIG. 2: Segment popularities. Histograms of segment popularities \( s_i \), defined as the relative number of times the \( i \)-th street segment (represented by edges in the street network) is visited on the Manhattan street network. The Manhattan street network was found using the python package ‘osmnx’. Segment popularity distributions from the taxi-drive process agree well with those obtained from empirical data (panel (a)), whereas those from the random walk do not (panel (b)). The taxi-drive process was run for \( 10^7 \) timesteps with \( \beta = 1 \). Note, this figure is taken from [26] and reproduced here for convenience.

Relation to other models. As discussed in the Introduction, the taxi-drive is closely related to the Levy walk. Before showing our results, we contrast the two processes. Like the taxi-drive, Levy walkers travel to randomly chosen destinations at constant speed along a trajectory, but instead of choosing a destination node directly (as the taxi-drive does) the walker jumps a distance \( r \) sampled from \( P(r) \propto r^{-1-\alpha} \) in a direction chosen uniformly at random. When the Levy walk takes place on a graph – the case we are concerned with in this work – the node nearest to the walkers landing site is chosen as the destination. Thus, the graph on which the Levy walker moves must be embedded in Euclidean space (although generalizations of Levy-type strategies to networks have been considered [34, 35]). Since the taxi-drive process is not defined in terms of distance, it choosing nodes as destinations directly, it can be applied to graphs with arbitrary topology. (Note, when the taxi-drive chooses shortest path routes to destinations, these shortest paths are calculated with respect to the number of edges, and not with respect to a euclidean distance.)

III. RESULTS

A. Stationary densities

We begin with perhaps the simplest topology, namely the linear graph which consists of \( N \) nodes arranged in a line with edges between adjacent nodes. We have normalized the betweenness and degree so that their sum is 1. The \( p_i \) are distributed similarly to the betweenness (orange and blue shaded areas overlapping perfectly); the \( p_i \) are not distributed in the same way the degree \( d_i \) as would be expected if the \( p_i \) were generated from a random walk. In panel (b) we show \( p_i \sim b_i \) relation for the taxi-drive holds true for other graphs, both regular and random. Note \( b'_i \) is the adjusted betweenness as defined by [6]. Note also that we have re-scaled by \( p_i \rightarrow p_i/p_{i,max} \) and \( b_i \rightarrow b_i/b_{i,max} \) so that they lie in the interval \([0,1]\).

FIG. 3: Stationary densities (a) Distributions of \( p_i \), betweenness, and degree for the linear graph, which consists of \( N \) nodes arranged in a line with edges between adjacent nodes. The relation between the taxi-drive holds true for other graphs, both regular and random. (Note, when the taxi-drive chooses shortest path routes to destinations, these shortest paths are calculated with respect to the number of edges, and not with respect to a euclidean distance.)
from the similarities between the taxi-drive and the definition of a node’s betweenness $b_i$, given by

$$b_i := \frac{1}{(N-1)(N-2)} \sum_{s,t,i \neq s,t} \sigma_{st}(i) \sigma_{st}$$

(1)

where $N$ is the number of nodes in the graph, $\sigma_{st}$ is the number of shortest paths connecting the start node $s$ and end node $t$, and $\sigma_{st}(i)$ is the number of those paths that contain the node $i$. Since asymptotically every start-end node combination will be sampled by the taxi-drive, and shortest paths are taken between these start and end nodes, a relationship between $b_i$ and $p_i$ is expected. They are however not precisely equivalent. Notice that the start and end nodes are not included in Eq. (1), as denoted by $i \neq s, t$ in the iterator of the sum. This is what makes $b_i \neq p_i$, since in the taxi-drive destination nodes are ‘counted’ when they are traversed by the taxi. Origins are however not, since the destination of one trip is the origin of the following trip (note taxis spend just one time unit on each node, and do not ‘stall’ between dropping off one passenger and picking up another). Hence a minor modification to equation (1)

$$b_i' := \frac{1}{(N-1)(N-2)} \sum_{s,t,i \neq s} \sigma_{st}(i) \sigma_{st}$$

(2)

which we call the adjusted betweenness leads to $b_i' = p_i$. Figure 3(b) shows the relation $b_i' = p_i$ holds true for a variety of different graphs.

B. Cover times

We next investigate the cover times of the taxi-drive process, which leads us to pose the the curious tourist problem: A curious tourist arrives in a city. She decides to explore the city by taking taxis to randomly chosen locations. How long does it take her to cover every road at least once?

Due to the non-markovian nature of the taxi-drive (the markovian property is violated since taxis move deterministically when serving passengers), we were unable to solve the curious tourist problem analytically (even for the simple, symmetric random walk exact results for cover times are rare; see 36 for a review). We thus resort to numerical evaluation.

Figure 4 shows how the mean cover time of the taxi-drive $\langle T \rangle_{TD}$ varies with graph size $N$ for the ring (panel (a)) and linear (panel (b)) graphs. For comparison, we also plot the mean cover time for the persistent random walk $\langle T \rangle_{PRW}$. The persistent random walk is a simple extension of the random walk with efficient covering properties 14, and so is a natural baseline for the taxi-drive. It differs from the regular random walk in that at each step the walker’s direction of motion persists with probability $(1-\epsilon)/2$ – and thus a reversal in direction occurs with probability $(1+\epsilon)/2$ – with $-1 < \epsilon < 1$. Note the regular random walk is recovered at $\epsilon = 0$. Note also that in order for the persistent random walk to be well-defined, the graph on which it is run must be embedded in some space, so that a random walker can persist in some ‘direction’. The ring and linear graphs have this property, and so the persistent random walk can be run on them (for the linear graph, the boundary conditions are reflective, so that the walker changes direction when these endpoints are reached).

The mean cover times of the persistent random walk on the ring and linear graph are known exactly 37 and are given by

$$\langle T \rangle_{PRW}^{ring} = \frac{1 - \epsilon}{2(1 + \epsilon)} N^2 + \frac{5\epsilon - 1}{2(1 + \epsilon)} N - \frac{2\epsilon}{1 + \epsilon}.$$  

(4)

From these the mean cover time of the random walk $\langle T \rangle_{RW}$ can be recovered by setting $\epsilon = 0$.

Figure 4 shows that taxi-drive covers the linear and ring graphs more efficiently than the regular random walk ($\epsilon = 0$) for all graph sizes $N$. For the ring graph (panel (a)), this holds true for the persistent random walk for all but extreme values of $\epsilon$ (i.e. $\epsilon \sim 0.999$). The fact that the persistent random walk beats the taxi-drive as $\epsilon \rightarrow 1$ makes sense, because at $\epsilon = 1$ the motion is purely ballistic and trivially covers the ring in $N - 1$ steps (assuming at $t = 0$ the walker has covered the node it starts on, leaving $N - 1$ nodes to be covered). For the linear graph (panel (b)) however the persistent random walk beats the taxi-drive for moderate values of $\epsilon$ (i.e $\epsilon = 0.5$).
We next study how $\langle T \rangle_{TD} \sim N^2$ for the ring and linear graphs. The curves in Figure 4 suggest the scaling might be simple polynomials $\langle T \rangle_{TD} \sim N^a$, so we fit the data to curves of the form $aN^2 + bN$ (the constant term being negligible as $N \to \infty$). In order for the fitting to be accurate, simulation at large $N$ must be collected. Beyond $N = O(10^4)$ however simulations were prohibitively costly – run times being $O(weeks)$ – so we did not collect data beyond this point. The results of the fitting were

$$\langle T \rangle_{TD}^{linear} = (0.501 \pm 0.001)N^2 + (0 \pm 2)N$$

$$\langle T \rangle_{TD}^{ring} = (10^{-6} \pm 10^{-6})N^2 + (2.52 \pm 0.01)N$$

From these we conjecture

$$\langle T \rangle_{TD}^{linear} \sim O(N^2)$$

$$\langle T \rangle_{TD}^{ring} \sim O(N)$$

Eq. 7 follows from Eq. 5. Eq. 8 follows from Eq. 6; given the closeness of the coefficient of the $N^2$ term in Eq. 6 to zero we conjecture that $\langle T \rangle_{TD} = O(N)$. Of course, since the data these predictions are based on vary over only four decades, we cannot rule out the presence of higher order terms with small coefficients. Thus we restate that Eq. 7 and Eq. 8 are intended as conjectures, whose proofs are open problems.

We remark that the different scaling properties for the ring and linear graphs are unusual; given the close similarities between the topologies of these graphs, properties of random searches on the graphs (such as cover times [4]) are typically similar in the $N \to \infty$ limit.

Next we study cover times on graphs with more complex topology, namely 2D lattices, barbells, Erdos-Renyi, and scale-free graphs. The barbell graph consists of two cliques joined by a single node (variants with more than a single node connecting the two cliques have also been considered). This topology is known to extremize the spreading properties of the random walk [38], so we include it as a baseline. Figure 5 plots the ratio of $\langle T \rangle_{TD}$ to $\langle T \rangle_{RW}$ – that is, compares the taxi-drive and the regular random walk only. We do not study the persistent random walk, since it is ill-defined on graphs not embedded in Euclidean spaces, as the aforementioned graphs are. Interestingly, as can be seen in Figure 5, for 2D lattices, Erdos-Renyi, and scale-free graphs below a certain size, the taxi-drive is more efficient than the random walk. This trend appears to hold true for all parameters $p$ of the Erdos-Renyi graphs, and parameters $m$ of the scale-free graphs (see caption of Figure 5 for a definition of these parameters). As expected for the barbell graph, $\langle T \rangle_{TD} < \langle T \rangle_{RW}$ for all graph sizes. The rationale here is that the random walker gets stuck in the bells of the barbell, which the ballistic aspect of the taxi-drive’s movements insulates it from this trapping.

C. m-Cover times

In some random search problems it is desired to cover each node more than once. This is necessary for example in urban sensing, where more than one sample at each spatial location is needed in order to capture the temporal fluctuations of the quantities being measured. This leads us to consider the $m$-cover time, the time it takes to cover each node at least $m$ times. Thus, we study how the $m$-cover time of the random walk compared to that of the taxi-drive, and so computed $\langle T \rangle_{TD} / \langle T \rangle_{RW}$ versus $m$ for the six graphs studied so far. We plot the result in Figure 6 which reveals unexpected trends. For the graphs with regular topology – the ring, linear, and 2D lattice – $\langle T \rangle_{TD} / \langle T \rangle_{RW}$ increases with $m$, eventually crossing the threshold value 1. Yet for the graphs with random topology – the Erdos-Renyi and scale-free – the opposite trend is observed: the taxi-drive beats the random walker as $m$ increases. (The barbell graph is an exception here; it has regular topology, but shows a decline in $\langle T \rangle_{TD} / \langle T \rangle_{RW}$ for increasing $m$. This isn’t too surprising since as discussed its topology maximizes $\langle T \rangle_{RW}$.)
Our work shows the taxi-drive is a competitive candidate for random search problems, since it covers graphs more efficiently than the persistent and regular random walk in various contexts. Moreover, the taxi-drive has the benefit of being defined on graphs with arbitrary topology, unlike the persistent random walk, Levy flight, and Levy walk which are confined to graphs embedded in Euclidean space (although as previously mention, generalizations of Levy-type strategies to networks have been considered [34, 35]).

The taxi-drive could be useful in community detection. Here, efficient algorithms have been designed by exploiting the relationship between the spreading properties of random walkers and graph topology; since the density \( p_i(t) \) of a random walker at a given node is related to its degree \( d_i \), nodes with similar degree— that is, nodes which form some ‘community’— can be detected by tracking \( p_i(t) \) of random walker. By swapping the random walk with the taxi-drive, for which our works shows \( p_i(t) \) is related to \( h_i \), perhaps other flavours of community could also be cheaply identified.

Our numerical study of the curious tourist problem raises many questions for future work. First, the conjectured scalings \( \langle T \rangle_{TD} = O(N) \) and \( \langle T \rangle_{linear} = O(N^2) \) cry out for theoretical explanation. Second, and more ambitiously, one expects that given the simpleness of the ring and linear graphs that an exact solution for \( \langle T \rangle_{TD} \) might be findable. Perhaps the techniques used in [37] to calculate \( \langle T \rangle_{PRW} \) could be useful for this purpose. Another interesting open problem is to determine which graph topology maximizes the cover time of the taxi-drive. The ‘stickiness’ of the bells in the barbell graph trap the random walker – what counterpart to this graph motif is needed to hamper the taxi-drive? Lastly, the puzzling behavior of the \( m \)-cover time begs explanation. Why does, as our work suggests, the regularity / randomness of the graph topology determine the scaling of \( \langle T \rangle_{TD}/\langle T \rangle_{RW} \) with \( m \)? We hope future work will solve these puzzles, as well as investigate other properties and applications of the taxi-drive.

Source code for the taxi-drive is available under the M.I.T. licence and can be found at [39].

Acknowledgments

The authors would like to thank Allianz, Amsterdam Institute for Advanced Metropolitan Solutions, Brose, Cisco, Ericsson, Fraunhofer Institute, Liberty Mutual Institute, Kuwait-MIT Center for Natural Resources and the Environment, Shenzhen, Singapore- MIT Alliance for Research and Technology (SMART), UBER, Vitoria State Government, Volkswagen Group America, and all the members of the MIT Senseable City Lab Consortium for supporting this research.

[1] S. Condamin, O. Bénichou, V. Tejedor, R. Voituriez, and J. Klafter, Nature 450, 77 (2007).
[2] O. Bénichou and R. Voituriez, Physics Reports 539, 225 (2014).
[3] S. Redner, A guide to first-passage processes (Cambridge University Press, 2001).
[4] V. Méndez, D. Campos, and F. Bartumeus, in Stochastic foundations in movement ecology (Springer, 2014), pp. 177–205.
[5] A. L. Lloyd and R. M. May, Science 292, 1316 (2001).
[6] G. M. Viswanathan, M. G. Da Luz, E. P. Raposo, and H. E. Stanley, The physics of foraging: an introduction to random searches and biological encounters (Cambridge University Press, 2011).
[7] O. Bénichou, C. Loverdo, M. Moreau, and R. Voituriez, Reviews of Modern Physics 83, 81 (2011).
[8] G. M. Viswanathan, V. Afanasyev, S. Buldyrev, E. Murphy, P. Prince, and H. E. Stanley, Nature 381, 413 (1996).
[9] F. Bartumeus, M. G. E. da Luz, G. M. Viswanathan, and J. Catalan, Ecology 86, 3078 (2005).
[10] M. L. Heuzé, P. Vargas, M. Chabaud, M. Le Berre, Y.-J. Liu, O. Collin, P. Solanes, R. Voituriez, M. Piel, and A.-M. Lennon-Duménil, Immunological reviews 256, 240 (2013).
[11] M. Vergassola, E. Villermaux, and B. I. Shraiman, Nature 445, 406 (2007).
[12] S. Havlin and D. Ben-Avraham, Advances in Physics 36,
695 (1987).
[13] D. Ben-Avraham and S. Havlin, *Diffusion and reactions in fractals and disordered systems* (Cambridge university press, 2000).
[14] V. Tejedor, R. Voituriez, and O. Bénichou, Physical review letters 108, 088103 (2012).
[15] P. Cénac, A. Le Ny, B. de Loynes, and Y. Offret, Journal of Theoretical Probability 31, 232 (2018).
[16] L. Basnarkov, M. Mirchev, and L. Kocarev, in *International Conference on ICT Innovations* (Springer, 2017), pp. 102–111.
[17] G. Oshanin, H. Wio, K. Lindenberg, and S. Burlatsky, Journal of Physics: Condensed Matter 19, 065142 (2007).
[18] M. Azaïs, S. Blanco, R. Bon, R. Fournier, M.-H. Pillot, and J. Gautrais, PLOS ONE 13, e0206817 (2018).
[19] M. F. Shlesinger, J. Klafter, and B. J. West, Physica A: Statistical Mechanics and its Applications 140, 212 (1986).
[20] A. Blumen, G. Zumofen, and J. Klafter, Physical Review A 40, 3964 (1989).
[21] G. Viswanathan, V. Afanasyev, S. V. Buldyrev, S. Havlin, M. Da Luz, E. Raposo, and H. E. Stanley, Physica A: Statistical Mechanics and its Applications 282, 1 (2000).
[22] M. A. Lomholt, K. Tal, R. Metzler, and K. Joseph, Proceedings of the National Academy of Sciences (2008).
[23] J. D. Noh and H. Rieger, Physical review letters 92, 118701 (2004).
[24] T. Weng, J. Zhang, M. Small, B. Harandizadeh, and P. Hui, Physical Review E 97, 032320 (2018).
[25] N. Masuda, M. A. Porter, and R. Lambiotte, Physics reports (2017).
[26] K. P. O’Keefe, A. Anjomshoaa, S. H. Strogatz, P. Santi, and C. Ratti, arXiv preprint arXiv:1811.10744 (2018).
[27] U. Lee and M. Gerla, Computer Networks 54, 4 (2010).
[28] B. Hull, V. Bukhovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, in *Proceedings of the 4th international conference on Embedded networked sensor systems* (2006), p. 125138.
[29] P. Mohan, V. N. Padmanabhan, and R. Ramjee, in *Proceedings of the 6th ACM conference on Embedded network sensor systems* (2008), p. 323336.
[30] A. Anjomshoaa, F. Duarte, D. Rennings, T. Matarazzo, P. de Souza, and C. Ratti, IEEE Internet of Things Journal pp. 1–1 (2018).
[31] M. J. Schnitzer, Physical Review E 48, 2553 (1993).
[32] H. C. Berg, *E. coli in Motion* (Springer Science & Business Media, 2008).
[33] C. Song, T. Koren, P. Wang, and A.-L. Barabási, Nature Physics 6, 818 (2010).
[34] A. P. Riascos and J. L. Mateos, Physical Review E 86, 056110 (2012).
[35] Q. Guo, E. Cozzo, Z. Zheng, and Y. Moreno, Scientific reports 6, 37641 (2016).
[36] M. Abdullah, arXiv preprint arXiv:1202.5569 (2012).
[37] M. Chapeau, O. Bénichou, and R. Voituriez, Physical Review E 89, 062129 (2014).
[38] G. Brightwell and P. Winkler, Random Structures & Algorithms 1, 263 (1990).
https://github.com/Khev/the_taxi_drive
[40] The exception being the 2D lattice; persistent random walks have been generalized to two dimensional spaces, but require an additional parameter to specify which direction is chosen when the walk changes its direction of motion. We wished to avoid this complication, so do not study this case.