HAL
open science

The Ariadne String against Covid-19 pandemic propagation part 2: oudoor path selection after lock-downs

Philippe Jacquet

To cite this version:

Philippe Jacquet. The Ariadne String against Covid-19 pandemic propagation part 2: oudoor path selection after lock-downs. 2020. hal-02611833

HAL Id: hal-02611833

https://hal.science/hal-02611833

Preprint submitted on 18 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Ariadne String against Covid-19 pandemic propagation part 2: oudoor path selection after lock-downs

Philippe Jacquet
Inria, Saclay Ile de France
France
philippe.jacquet@inria.fr

***Abstract*—In [1] we have presented the Ariadne String against Covid application, whose aim was to recommend safe outdoor path inside cities in order to limit the exposure to virus propagation during aimless outdoor leisure excursion. The main idea was to make a load balancing between streets via space-time diversity via a random walk. The application was shown to be efficient at the very first user. The present work is an extension of this previous work and can be applied after lock-down where outdoor excursions have a aim. We use drifted random walks which under the condition of uniform distribution of initial starting points and destination locations gives a perfect load balancing of the streets, thus minimizing the exposure to virus (by a factor around 3).**

I. Introduction

The Covid-19 pandemie has forced more than half of mankind into lock-down situation. During lock-down and in some countries, outdoor excursion were authorized under strict restriction; in France, no more than 1 hour less than 1km from home. These restrictions had aim to reduce the exposure rate to virus through outdoor contact with other persons. This problem is of course crucial in urban places, since in country places the distanciation are more easy to achieve. In [1] we have described an application, called Ariadne String against Covid, which reducte the contact rate thanks to a better load balancing the streets while avoiding crowded areas. The fundamental tool in Ariadne Covid is the use of random walk which naturally load balance edges in a graph. The application is shown to give a benefit to the first user, although to have an impact on the pandemie, it must be used by a majority of users. The average outdoor exposure rate reduction is around 3, but it is difficult to verify this claim since it is based on pre lock-down partial data on some cities.

In the present work we present Ariadne Covid2 as a followup application whose aim is to limit outdoor exposure to virus *after* lock-downs. The lock-downs aim is to "flatten the curve" in order to limit the congestion in hospitals. It is equivalent to the graphite bars used to slow down and reverse the reactions in the kernel of a nuclear plant. If you remove the graphite bars, the reaction will restart. Therefore the social distancing and other ways to limit exposure will need to be continued after lock-down before either a vaccine or a herd-immunity is developed. Indeed the situation is even more dangerous than before lock-down, since the infected population ratio is larger. In particular there would need be some care on our outdoor excursion.

But contrary to lock-down outdoor excursions which were set-up for healthy consideration and therefore were aimless (walking, jogging), the outdoor excursions after lock-down will have a aim, most likely in order to commute between home and work place. The public transportation is considered to be unsafe since people will packed in small areas, and it should be restricted in order to achieve a sufficient social distances. The city of Paris envisions to help people to rely on biking and other surface transportation by increasing the number of streets with biking lanes. However the biking lanes are narrow and sometimes very busy because people commute around the same hours. Our work is to introduce a policy which helps the user to find a biking route from home to work which limit the exposure rate by achieving a load balancing between lanes.

In this condition a simple random walk cannot work since it will take an eternity for a random walk to connect the home address with the work place, if the latter is a random but fixed location in the city. The shortest path will not be good, since it is likely to take the main streets and create a gathering on those streets. Our approach will be based on *drifted* random walks so that the penality with the shortest path will be limited, while the aggregation of path from all users will give an optimal load balancing over the streets and therefore minimize the exposure rate.

II. Description of the algorithm

The user sends its GPS coordinate $z_0 = (x_0, y_0)$ of its home address or initial position. She/he also sends the coordinate of the final destination $z_f = (x_f, y_f)$.

The algorithm operates on an abstract data set which is a set of street intersection linked by segment of a street. The intersection I is given by its GPS coordinates $(x(I), y(I))$. Two intersections I and J are connected if there exist a segment of street which connect the two intersections. Let V the set of intersections and E the set of street segments, The pair (V, E) forms a graph. We also have the set S of streets, A street is a set of contiguous street segments.

The neighbors of an intersection I are the list of the intersections to which I is connected. We use a number $\epsilon > 0$ fixed for the protocol.

The initial position of the user is on a segment $(I_U, I_D) \in E$. The path will take the intersection among I_U and I_D which lies ahead of destination, *i.e* if $\langle z_f - z_0 | I_U - I_D \rangle \geq 0$ then I_U is selected: $z_1 = I_U$, otherwise $z_1 = I_D$.

At step number k, let z_k be the intersection were the path is currently ending. Let d be the degree of the intersection z_k. If $d = 1$, then the path backtracks. If $d > 1$ then with probability ϵ the path proceeds with the non backtracking random walk, *i.e.* the path take any other neighbor edge distinct of (z_{k-1}, z_k) with equal probability. Otherwise, with probability $1 - \epsilon$, the path take one the two the best sectored edge as defined as follow. Assuming the angle of the edge (z_k, z_{k-1}) is θ_0, and we enumerate θ_1, $\theta_2, \ldots \theta_{d-1}$ the angle in increasing order of the other edges of intersection z_k. Let θ be the angle of the vector (z_k, z_f) toward the destination.

Let j such that $j\frac{2\pi}{d} < \theta \leq (j+1)\frac{2\pi}{d}$. Of course $j+1$ must be considered *modulo* d. Let $\alpha_d(\theta)$ and $\beta_d(\theta)$ such that $\alpha + \beta = 1$ and $\alpha e^{2ij\pi/d} + \beta e^{2i(j+1)\pi/d}$ be proportional to $e^{i\theta}$. Numerically we have

$$\begin{cases} \alpha_d(\theta) & = & \frac{\sin(2(j+1)\pi/d - \theta)}{\sin(\theta - 2j\pi/d) + \sin(2(j+1)\pi/d - \theta)} \\ \beta_d(\theta) & = & \frac{\sin(\theta - 2j\pi/d)}{\sin(\theta - 2j\pi/d) + \sin(2(j+1)\pi/d - \theta)}. \end{cases} \quad (1)$$

The selection of the next step z_{k+1} is done as follow: with probability α it selects the edge corresponding to angle θ_j, and with probability β the edge corresponding to the angle θ_{j+1}.

We notice that there is no reason that the selected edge actually heads toward the destination (in fact it will never do), but in average it will do, under mild conditions on the actual street angles distribution in the city.

Notice that when θ_0 is the actual angle selected by the path algorithm selection, then the path backtracks. In this case the loop removal should be applied to the actual path, this will even more reduce the actual weight of the path.

The algorithm is much simpler than the actual Dijskstra shortest path algorithm, indeed the shortest algorithm leads to at least an average quadratic complexity in terms of the number of vertices, while the randomized algorithm is at most linear, in fact proportional to the diameter of the graph. This will greatly help the computational power required to have a server capable of serving several millions of requests. However we shall not expect the algorithm to provide a shortest path, but we expect that the randomized algorithm will provide a reasonable penalty for having non optimal path.

We can show that under the *isotropic walk* condition the algorithm leads to a uniform distribution.

Definition 1 (Isotropic walk condition)**.** *A walk is isotropic at a given intersection I, when reaching intersection I, the difference of the angle θ toward the destination with the angle θ_0 of the arriving edge, satisfies a fixed distribution $P_I(\theta - \theta_0)$.*

Notice that the walk arriving at a destination has randomness assuming the randomness of the initial and destination coordinates. Notice that the isotropic condition does not implies that $\theta - \theta_0$ must always be close to π (most direct path) and when $\theta - \theta_0$ is small the path will most likely backtrack.

Theorem II.1. *Under the isotropic walk condition, and assuming same constant speed v of the travellers, the aggregation of path leads to a uniform densities on the streets.*

Proof. We use the classic proof of stationary random walks in undirected graphs. Let assume that at time t every street in city has the same exit rate $\rho(t)$ on each of its end points. We will prove that all streets have the same entrance rate. And consequently the same exit rate on the other end. Thus the uniform distribution is the stationary distribution of the entrance rates.

In the following we assume that there is no loss between entrance and exit rates, assuming for example that path initial points and destination points exactly balance in each streets. Or equivalently that an arrival point coincides with a departure point like in a random way point mobility model.

Let consider a traveller arriving at an intersection I which has d arriving streets. According to the uniformization hypothesis, the traveller has an equal probability to arrive on any of the d intersecting edges. With probability ϵ it selects any of the $d-1$ other streets. With probability $1-\epsilon$ it select one of the best sectored edge. Enumerating the edges in counter clockwise way, initialising with the entrance edge, the isotropic walk conditions leads to the following expression of the probability $p_d(j)$ that the edge j is selected is

$$p_d(j) = \int_{2j\pi/d}^{2(j+1)\pi/d} P(\theta)\alpha_d(\theta) + \int_{2(j-1)\pi/d}^{2j\pi/d} P(\theta)\beta_d(\theta). \quad (2)$$

Therefore the probability that a edge is selected, independently of the entrance edge is

$$\frac{d-1}{d}\epsilon + (1-\epsilon)\sum_{j=0}^{j=d} p_d(j). \quad (3)$$

Thus the exit rates are uniform, and have value $\rho(t)$ and consequently the stationary rate distribution is uniform with some value ρ. Since the speed is considered to be the same on each street the density of traveller on each street is ρ/v per unit length. □

As an intermediate step, there is the randomized geo-routing algorithm which is very close to the isotropic walk with the difference that instead of assuming equal angular sectors $j\frac{2\pi}{d}$ for the exit streets at each intersection, we take the actual angle of the streets. The consequence is that (i) the average exit path is now well aligned with the angle toward the destination, but (ii) we lose the isotropic property, since the densities on the exit street will now vary with the variation of the angles between the streets (which are null in the grid model but tend to be large in the Delaunay model as we will see in the next sections).

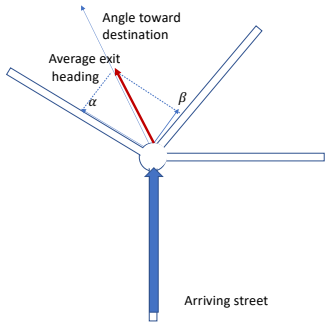

Figure 1: Illustration of the functioning of the randomized georouting algorithm. The traveller enters the intersection via the large blue arrow.

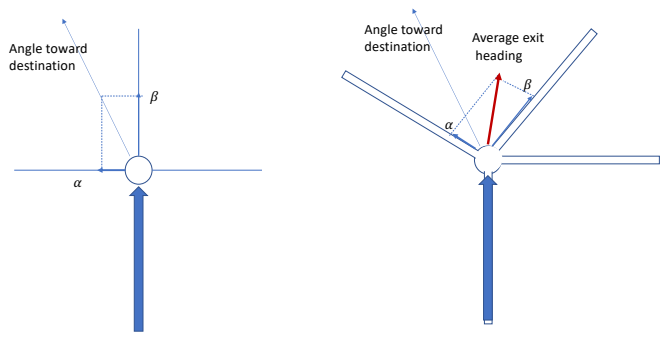

Figure 2: Illustration of the functioning of the randomized isotropic walk algorithm. The traveller enters the intersection via the large blue arrow.

III. Simulation of the Algorithm

A. Simulation on a grid

We have chosen an abstraction of Paris map via a 100x100 map. Each grid point is an intersection in Paris. Each segment represents a street segment of one unit length. From North to South the streets represents 100 length units, and the total cumulative street length length is 20,000 length units. This compares well with the actual diameter of Paris (8km) and its cumulative street length of 1500 km (thus 2900 km of street sides).

The mayor of Paris have decided to create bike lanes parallel to the 14 subway lines, in order to foster the use of bikes instead of public transportation after lock-down. To simulate those bike lanes we have highlighted 14 paths (seven paths West-East, seven paths North South) on the grid map.

We have run $N_0 = 8,000$ sources and destination pair on the two algorithms. One algorithm is the randomized algorithm, the second algorithm is the preferential path algorithm. In order to simulate the attraction effect of the highlighted path, we have artificially distorted the graph by decreasing the segment path weight by 20% and run a shortest path Dijskstra algorithm. This a moderate ratio but it is sufficient to create a significant attraction effect, otherwise the difference would be not visible due to the large diversity of shortest paths in a grid structure. We simulate the isotropic randomized algorithm with $\epsilon = 0$.

Figure 3 and 4 show the histogram of the segment density loads consequence of the application of the algorithms. For each segment, we collect the traffic for both ways, but for the figures we add the two traffic per segment to display the histogram. We notice that the preferential path algorithm shows that the preferential streets are drastically more busy than the other streets. The randomized algorithm shows a more balanced distribution of traffics. There are not perfectly uniform since the isotropic condition fails on the borders of the map and holds in the large central part of the map.

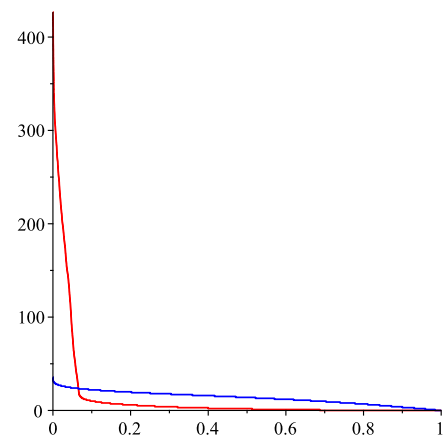

Figure 3: Histogram of segment traffic load in Grid Paris, in red the preferential path algorithm, in blue the randomized algorithm

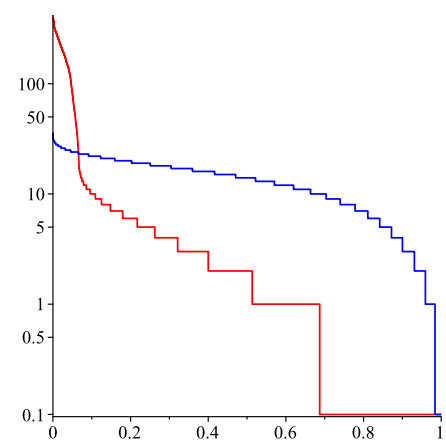

Figure 4: Histogram of segment densities in Grid Paris, in logarithmic scale

Figure 5 shows the map of the traffic of street segments with the preferential path algorithm. In green are the streets with load larger than 1 but smaller than 15, in blue, traffic

smaller than 25, in red, smaller than 50, in black, smaller than 200. The preferential paths are clearly visible and marked as very busy (black and red) as expected. The figure 6 shows the map of empty streets (lined in grey) which is a dense network. Figure 7 and figure 8 show the same data for the randomized algorithm, but the picture are now different, since the segment load are very balanced and the empty street seldom.

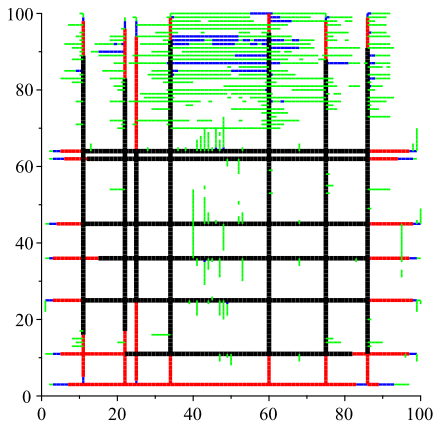

Figure 5: Map of segment densities in Grid Paris for the preferential path algorithm.

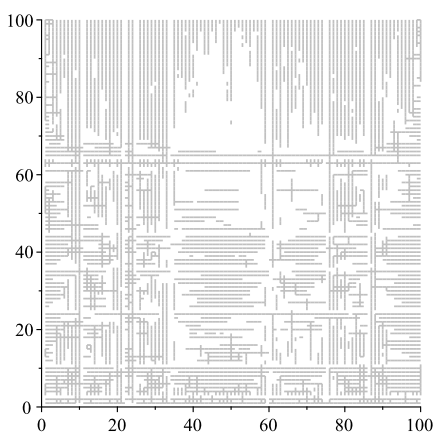

Figure 6: Map of empty streets in Grid Paris for the preferential path algorithm.

Now our aim is to derive the average exposure time of the travellers in Grid Paris. As in [1] we denote by $\mathcal{S}$ the set of street segment, with the difference, that the segment are directed and we differentiate a segment with its reverse segment. We denote $\ell(s)$ the segment length, thus the cumulative length is $L = \sum_{s\in\mathcal{S}} \ell(s)$. We denote $\lambda(s)$ the traffic load after N_0 initial-destination random pairs. The average path length is $L_G = \frac{1}{N_0}\sum_{s\in\mathcal{S}} \lambda(s)\ell(s)$. The average travel time is L_G/v.

If the simulated time is T, on segment s the entrance frequency rate is $\frac{\lambda(s)}{T}$ and the density on the segment is $\frac{\lambda(s)}{vT}$, assumed to be Poisson. If the number of traveller is N then the density on the segment is $\frac{\lambda(s)}{vT}\frac{N}{N_0}$ where N_0 is the number of travellers needed to simulate the estimate $\lambda(s)$.

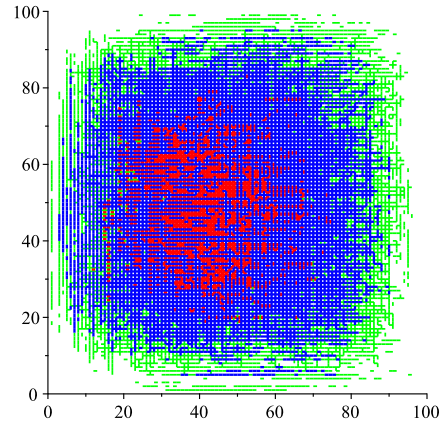

Figure 7: Map of segment densities in Grid Paris for the randomized algorithm.

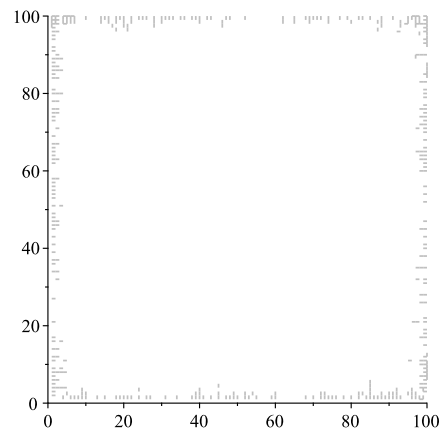

Figure 8: Map of empty streets in Grid Paris for the randomized algorithm.

Given this Poisson density, the probability that a random point at a random time on the segment does not have a traveller at distance within R_0 is $\exp\left(-\lambda(s)\frac{2R_0}{vT}\frac{N}{N_0}\right)$. If R_0 is the safe distance against virus, for biking $R_0 = 10m$, for a random traveller the average cumulative exposure time $E(N)$ is

$$E(N) = \sum_{s\in\mathcal{S}} \frac{\lambda(s)\ell(s)}{vN_0}\left(1-\exp\left(-\lambda(s)\frac{2R_0}{vT}\frac{N}{N_0}\right)\right). \quad (4)$$

If we want to know the exposure time $E_{12}(N)$ for a single traveller of the randomized algorithm when all other travellers use the preferential paths, we get the expression

$$E_{12}(N) = \sum_{s\in\mathcal{S}} \frac{\lambda_2(s)\ell(s)}{vN_0}\left(1-\exp\left(-\lambda_1(s)\frac{2R_0}{vT}\frac{N}{N_0}\right)\right) \quad (5)$$

where $\lambda_1(s)$ is the density of segment s for the preferential algorithm, and $\lambda_2(s)$ is for the randomized algorithm.

The figures 9 and 10 show the average cumulative exposure time versus the travelling population, when all travellers are on preferential path, when all travellers are on the isotropic randomized algorithm, and the average cumulative exposure time when a single traveller is on randomized algorithm, and

the other are on preferential paths. We display for a peak traffic period duration of 2 hours, and for a peak duration of 4 hours. For one day these quantities should be multiplied by two, since the travellers commute twice a day. For the traveller speed we have opted for $v = 12$km per hour, which is the average speed of bike commuting in Paris.

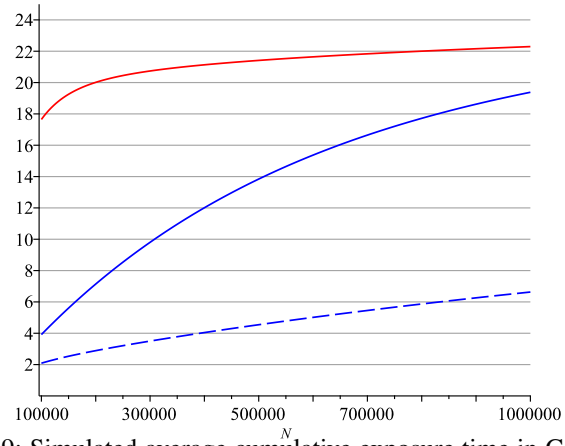

Figure 9: Simulated average cumulative exposure time in Grid Paris during a peak time of 2 hours versus total number of travellers: red: all traveller on preferential paths, blue: all travellers on randomized algorithm, dashed: one traveller on randomized algorithm.

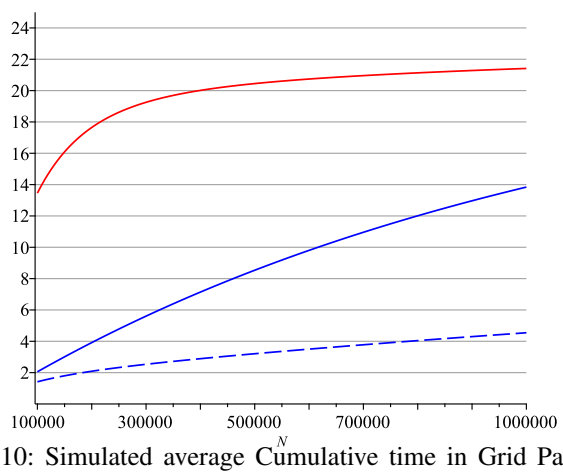

Figure 10: Simulated average Cumulative time in Grid Paris during a peak time of 4 hours versus total number of travellers: red: all traveller on preferential path, blue: all travellers on randomized algorithm, dashed: one traveller on randomized algorithm.

B. Simulation on a Voronoi triangulation

We have created a map of 4,000 random points on a 10kmx10km square and connected them through the Delaunay triangulation. In this simulation the streets are the edges of the triangulation and the nodes are the intersections. With 4,000 nodes we get a total street length of the order of 200 times the diameter of the map, thus showing a similar street density than Paris street map. As in the previous simulation we have connected 14 pairs of points, 7 North-South, seven West-East to simulate the subway network. The bike lane supposed to follow the subway lines is made of the path obtained via shortest path algorithm between the points of each pair.

We have simulated four algorithm:

1) The preferential path algorithm;
2) the shortest path algorithm;
3) the isotropic walk algorithm;
4) the geo-routing algorithm

The three last algorithms perform the same in a grid network, this is why we did not distinguish them in the previous section. The isotropic walk algorithm is the algorithm described in the paper. Notice that the shortest path algorithm is the most expensive of all three since its quadratic in the number of nodes and therefore maybe expensive in term of server complexity.

In the preferential path algorithm we still use the shortest path algorithm but now we reduce the weight of the edges in the preferred path by 90%. This is kind of a drastic reduction but it simulates well the use of subway where the commuter takes the path to the closest subway station and enlights on the subway station closest to his/her destination location.

We have run $N_0 = 5,000$ pairs of initial point and destination. Figure 11 and 12 show the histogram of the segment density loads consequence of the application of the algorithms. In red the preferential algorithm, in brown the isotropic walk algorithm, in blue the geo-routing algorithm, in green the shortest path algorithm. As expected the isotropic walk algorithm shows the best balance (better look at the logarithmic scale).

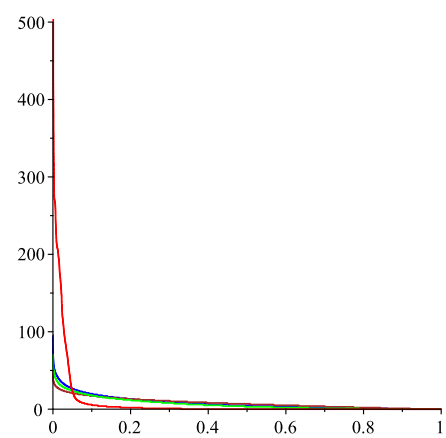

Figure 11: Histogram of segment traffic load in Delaunay Paris, in red the preferential path algorithm, in blue the randomized algorithm

Figure 13 shows the map of the traffic of street segments with the preferential path algorithm. In green are the streets with load larger than 1 but smaller than 15, in blue, traffic smaller than 25, in red, smaller than 50, in black, smaller than 200. The preferential paths are clearly visible and marked as very busy (black and red) as expected. The figure 14 shows the map of empty streets (lined in grey) which forms a dense

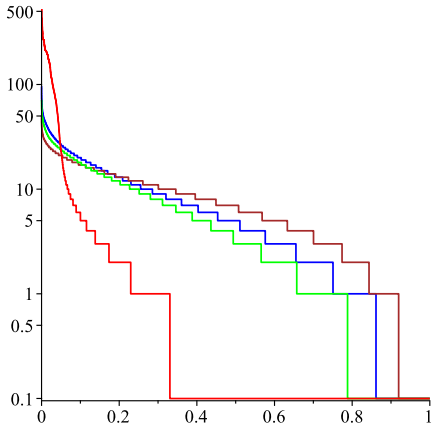

Figure 12: Histogram of segment densities in Delaunay Paris, in logarithmic scale

network. Figure 15 and figure 16 show the same data for the shortest path algorithm, but the picture are now different, since the segment load are better balanced and the empty street much less dense. Figure 17 and figure 18 show the same data for the isotropic walk path algorithm, since the segment load are very well balanced and the empty street very seldom. As expected the empty streets are on the border and the density balance more in the central part. Figure 19 and figure 21 show the same data for the geo-routing path algorithm, since the segment load are a little less well balanced and the empty street less seldom.

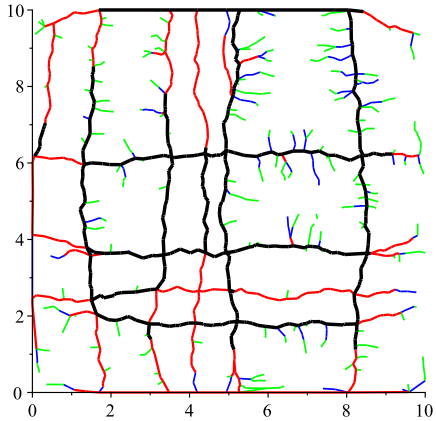

Figure 13: Map of segment densities in Delaunay Paris for the preferential path algorithm.

Figure 28 shows the cumulative exposure time experienced by the travellers in the different routing algorithms during a peak of 2 hours versus the commuting population size. The shortest path algorithm provides the smallest exposure because the average path length is shorter despite the path density is larger. Indeed the exposure rate discrepancy diminishes when the commuting population diminishes. But the shortest path algorithm is too expensive in terms of complexity to answer to millions of request. The georouting and isotropic walk algorithms show similar performance. Although the isotropic

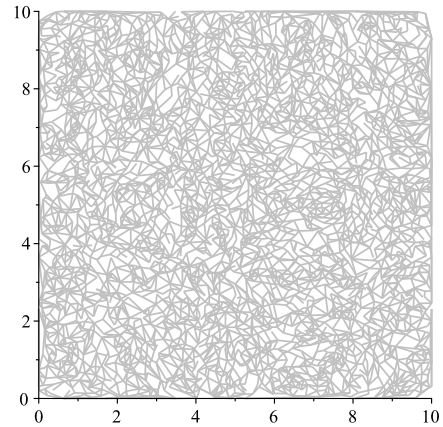

Figure 14: Map of empty streets in Delaunay Paris for the preferential path algorithm.

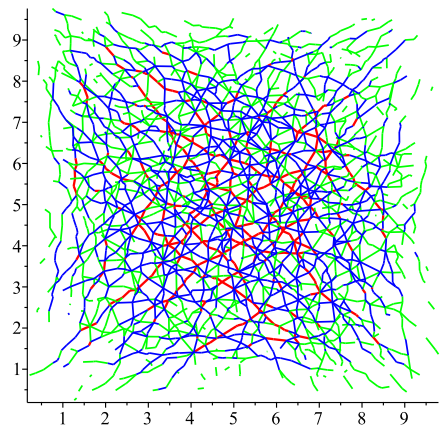

Figure 15: Map of segment densities in Delaunay Paris for the shortest path algorithm.

algorithm benefits from its more balanced street loads at lower traffic. Figure 29 shows the respective average path length for each of the algorithms.

Since the preferential algorithm shows terrible performance, we investigate the possibility to multiply the parallel lanes to the preferential path in order to reduce the exposure time via a reduced density. The figure 23 shows the average exposure time versus the multiplicative factor of the parallel lanes. It turns out that the preferential lanes must be multiplied by at least ten in order to get closer to the performance of the isotropic algorithm.

C. Single pair simulation

In this section we evaluate the performance of the algorithms when we consider a single pair of initial point and destination point. We have simulated 1,000 travellers commuting between these two points. Of course the preferential path and shortest path algorithm are blocked on a single path because there a unique solution to the Dijkstra algorithm. But for the geo-routing algorithm and the isotropic walk algorithms have a random component and provide a diversity in the path. That way the two later algorithms offer a lower exposure

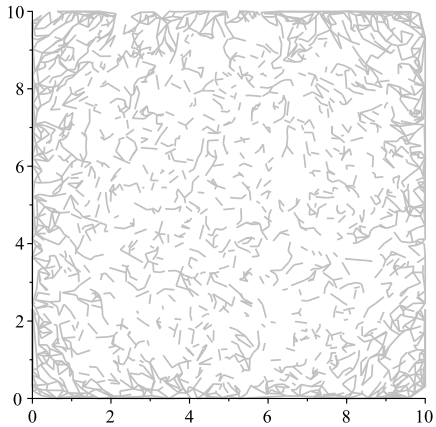

Figure 16: Map of empty streets in Delaunay Paris for the shortest path algorithm.

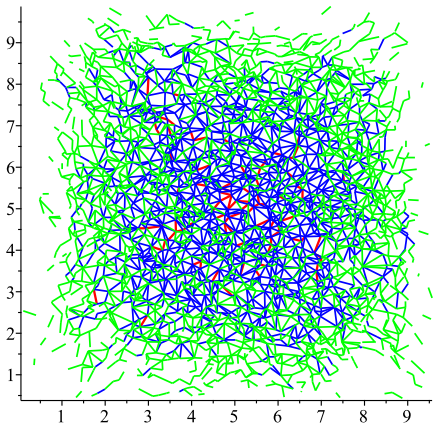

Figure 17: Map of segment densities in Delaunay Paris for the isotropic walk path algorithm.

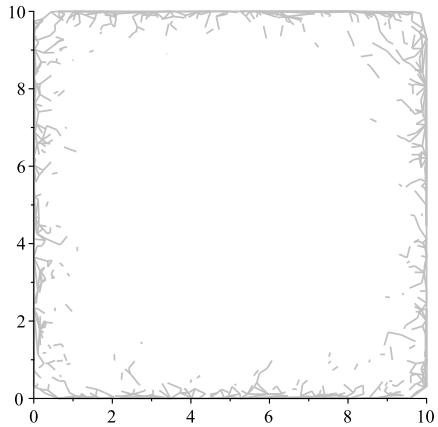

Figure 18: Map of empty streets in Delaunay Paris for the isotropic walk path algorithm.

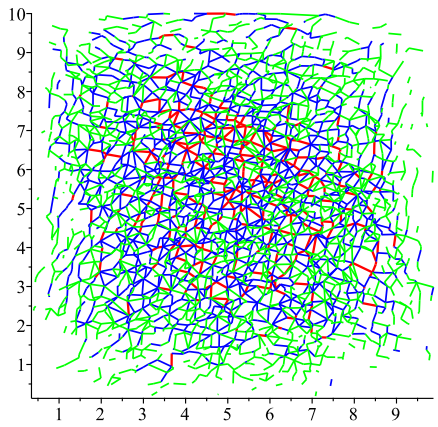

Figure 19: Map of segment densities in Delaunay Paris for the geo-routing path algorithm.

thanks to the path diversity. The figures 24, 25 26,and 27, shows the traffic of the segment for the four algorithms. yellow is for a traffic between 1 and 15 passage, green between 15 and 25, blue between 25 and 50, red between 50 and 200, black above 200. We notice that the path network of the geo-routing algorithm is more vascular and therefore uses less segments. Figure 28 shows the cumulative exposure time of the four algorithm under this situation.

Figure 29 shows the average path length for the various algorithms under the condition of travel between a single pair.

D. Single destination point or single initial point simulation

In this section we evaluate the performance of the algorithms when we consider a single destination or a single initial point. We have simulated 4,000 travellers commuting from or toward this point. Of course the preferential path and shortest path algorithm show the same paths because there a unique solution to the Dijkstra algorithm and the solution is symmetric on both ways. But for the geo-routing algorithm and the isotropic walk algorithms have a random component and are asymmetric with respect to the order in the pair made by the initial point and the destination point. The figures 30, 31, 32,and 34, shows the traffic on street segments for the four algorithms when diverging from the same initial point, and figures 33,and 35, shows the traffic on street segments for the two last algorithms when converging on the same destination point. Figure 37 shows the cumulative exposure time of the four algorithm under this situation.

Figure 36 shows the average path length for the various algorithms under the condition of travel diverging from a fixed initial point.

References

[1] Jacquet, P. (2020). The Ariadne String against Covid-19 pandemic propagation during lock-downs. https://hal.archives-ouvertes.fr/hal-02546347/

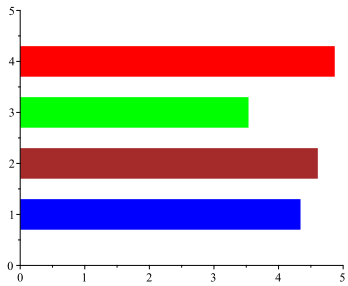

Figure 20: Average path length for the four algorithms, in km, from top to bottom: preferential algorithm, shortest path algorithm, isotropic walk algorithm, georouting algorithm.

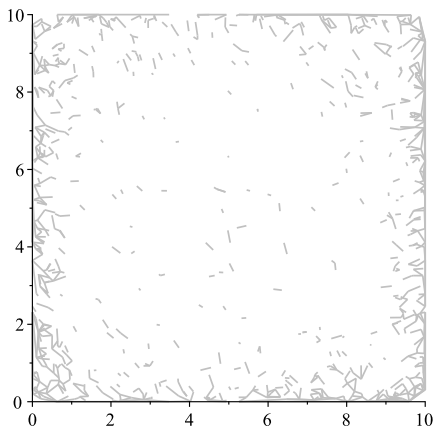

Figure 21: Map of empty streets in Delaunay Paris for the geo-routing path algorithm.

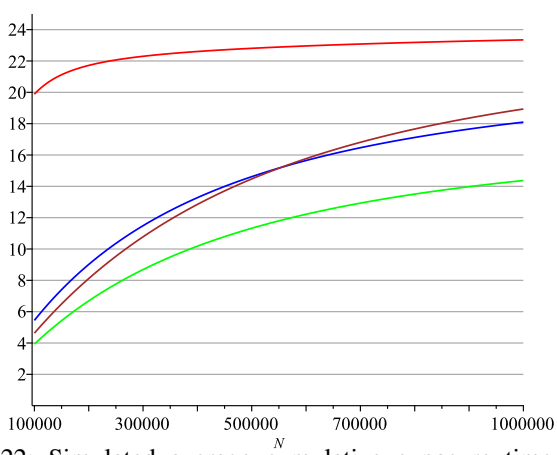

Figure 22: Simulated average cumulative exposure time in Delaunay Paris during a peak time of 2 hours versus total number of travellers: red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.

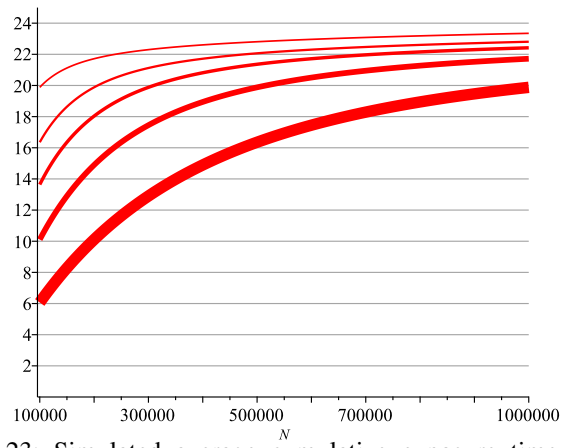

Figure 23: Simulated average cumulative exposure time in preferential path algorithm. From top to bottom: one lane, two parallel lanes, three parallel lanes, five parallel lanes, 10 parallel lanes.

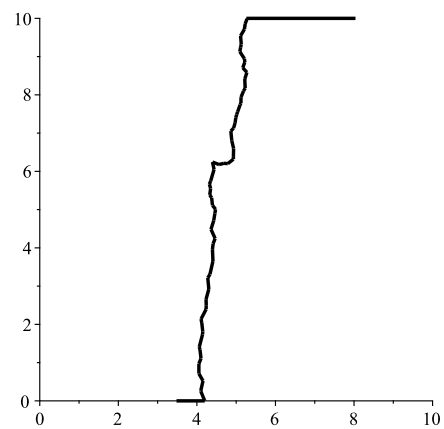

Figure 24: Map of segment densities in Delaunay Paris for the preferential path algorithm on a single pair.

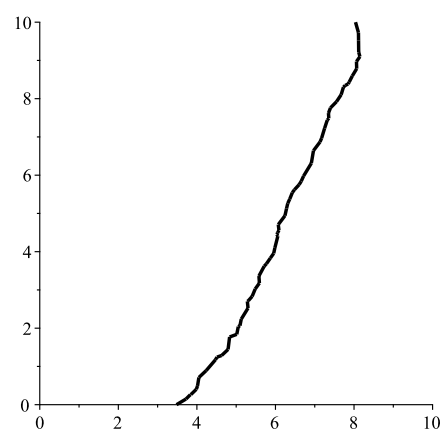

Figure 25: Map of segment densities in Delaunay Paris for the shortest path algorithm on a single pair.

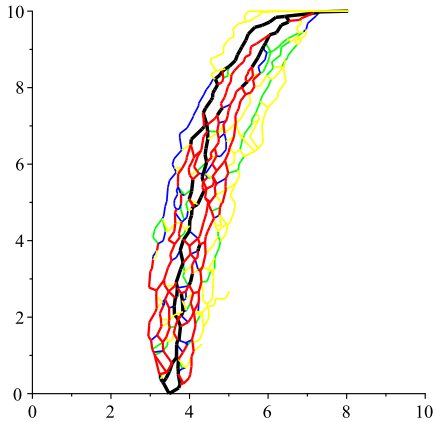

Figure 26: Map of segment densities in Delaunay Paris for the geo-routing path algorithm on a single pair.

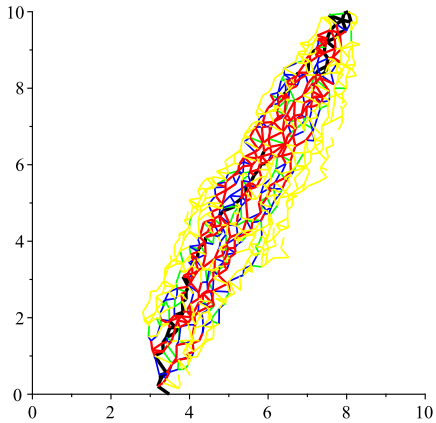

Figure 27: Map of segment densities in Delaunay Paris for the isotropic walk path algorithm on a single pair.

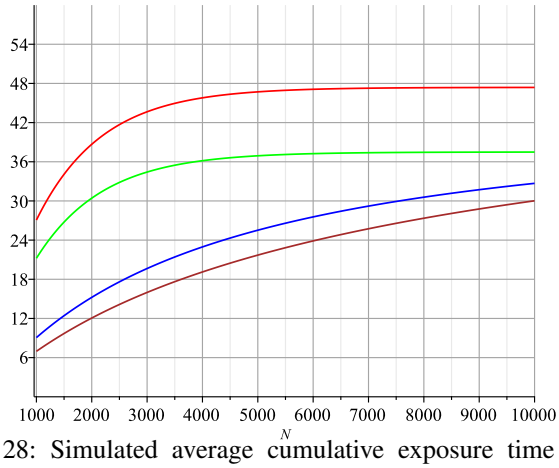

Figure 28: Simulated average cumulative exposure time in Delaunay Paris during a peak time of 2 hours versus total number of travellers on a single pair initial point and destination: red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.

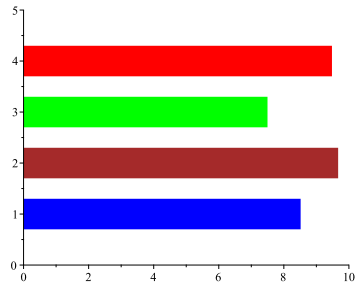

Figure 29: Average path length for a single pair: red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.

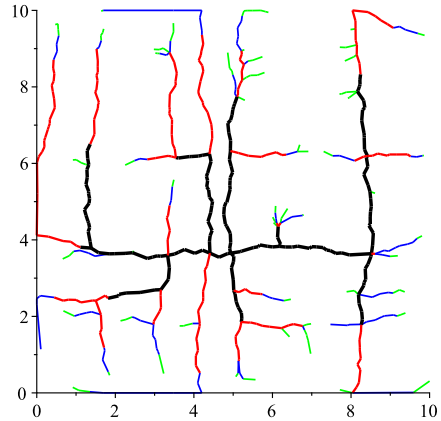

Figure 30: Map of segment densities in Delaunay Paris for the preferential path algorithm diverging on a single initial point.

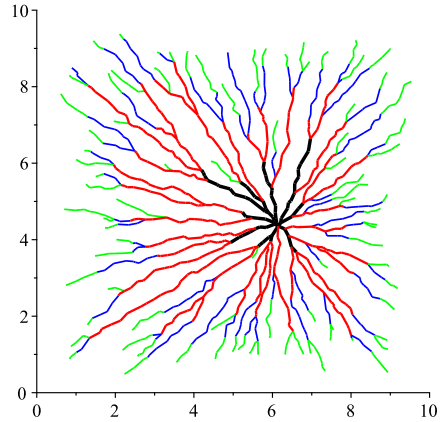

Figure 31: Map of segment densities in Delaunay Paris for the preferential path algorithm diverging on a single initial point.

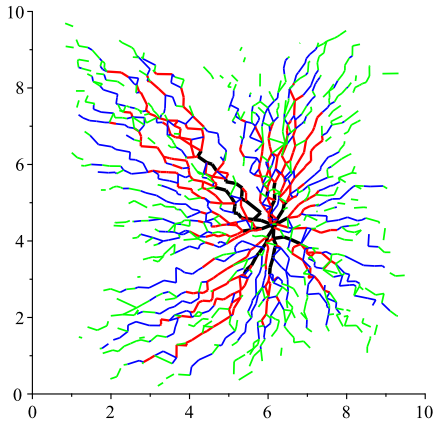

Figure 32: Map of segment densities in Delaunay Paris for the geo-routing path algorithm diverging from a single initial point.

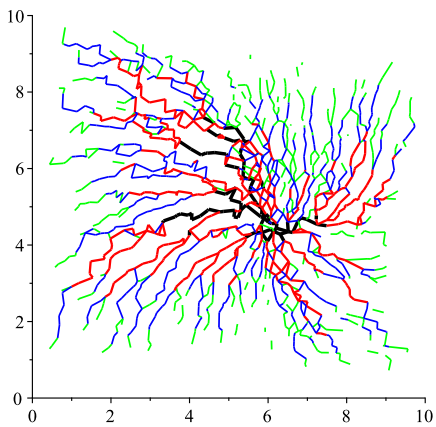

Figure 33: Map of segment densities in Delaunay Paris for the geo-routing path algorithm converging on a single destination point.

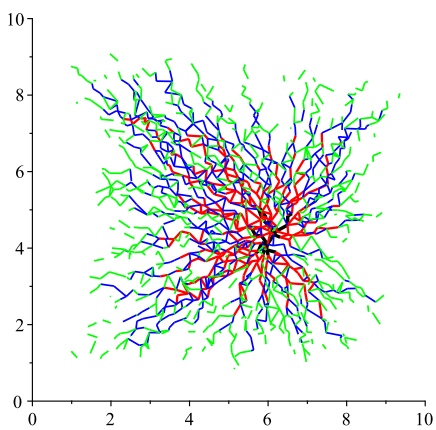

Figure 34: Map of segment densities in Delaunay Paris for the isotropic walk path algorithm diverging from a single initial point.

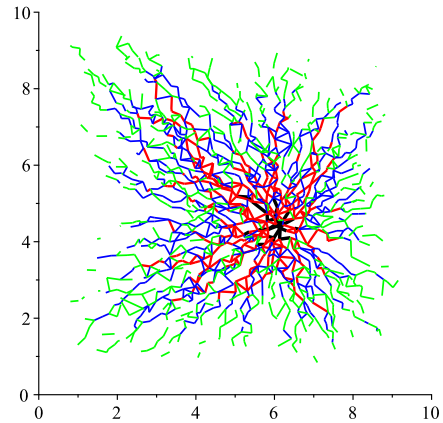

Figure 35: Map of segment densities in Delaunay Paris for the isotropic walk path algorithm converging on a single destination point.

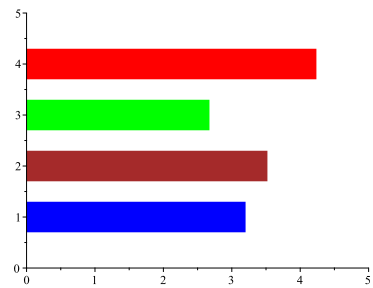

Figure 36: Average path length for a single initial point: red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.

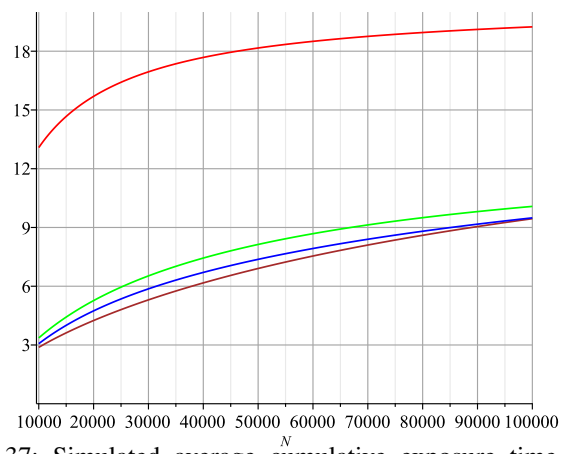

Figure 37: Simulated average cumulative exposure time in Delaunay Paris during a peak time of 2 hours versus total number of travellers on a single initial point: red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.