Decentralized routing on stochastic temporal networks

R. Lambiotte
Department of Mathematics
Research group on Complexity and Networks
Namur Center for Complex Systems
University of Namur, Belgium

Decentralized Routing on Spatial Networks with Stochastic Edge Weights, Till Hoffmann, R.L. and Mason Porter, PRE 2013
From static to temporal networks

Empirical data on interacting systems are more and more often time-resolved: timings at which events take place in the system.

Observations
Node and link dynamics exhibits complex temporal patterns

Models
Burstiness and temporal correlations affect spreading

Algorithms?
Modularity
?
Routing algorithms for static networks

On static networks: finding shortest paths can be done efficiently
Routing algorithms for static networks

On static networks: finding shortest paths can be done efficiently ... but requires complete knowledge of the system.
However, some networks can be navigated efficiently by using only local information (see Milgram)
Decentralized algorithms for networks embedded in space.
Routing algorithms for temporal networks

Definition of the *stochastic shortest path* problem when the network is not deterministic

**Algorithms** to find shortest paths

Approximate, *decentralized* algorithm finding good paths does not require the knowledge of the whole system.
Routing on deterministic networks

We associate a weight $T_{ij}$, representing a cost or travel time, with each edge $(i, j)$. A path $\ell$ with $k$ steps is a sequence of $k + 1$ nodes $\ell = \{i_1, \ldots, i_{k+1}\}$ that are connected to one another via edges. The weight $T_{\ell}$ of a path $\ell$ is given by the sum of the weights of its constituent edges:

$$T_{\ell} = \sum_{j=1}^{k} T_{ij}i_{j+1}.$$ 

The shortest-path problem (SPP) aims to determine the path from an origin node to a target node that has the smallest total weight. In the DSPP, each edge weight $T_{ij}$ is deterministic, and a path with minimal total weight is considered to be \textit{optimal}. 
Routing on stochastic networks

The weights are now stochastic variables, distributed with a given probability. The probability that edge $ij$ has weight $t$ is:

\[ p_{ij}(t) \]

This random variable can be:
- a nondeterministic travel time in a transportation networks
- the waiting time of a random walker before an edge appears on a temporal network

In this work, we assume that:
- weights are independent of each other,
- the PDFs do not change during the routing process

\[ \Rightarrow \text{Distributions are assigned to edges instead of weights (scalar)} \]
The probability for a path $\ell$ to have a certain weight is:

$$p_\ell(t) = \left( \prod_{j=1}^{k} p_{i_j, i_{j+1}} \right) (t),$$  

where the right-hand side denotes $k$ consecutive convolutions. The probability to traverse the path $\ell$ and incur a weight $T_\ell \leq t$ is given by the cumulative distribution function (CDF)

$$u_\ell(t) = \int_0^t dt' p_\ell(t').$$
What is the shortest path?

In contrast with the deterministic case, there is no longer a unique concept of optimality.

Frank defines a path to be optimal if its cdf surpasses a threshold within the shortest time.

Fan suggests maximizing the cdf for a given time budget.

FIG. 1: Comparison of path optimality criteria using CDFs of three paths. Fan et al.’s criterion prefers paths 2 and 3 to path 1 but cannot discriminate between the CDFs of paths 2 and 3. Frank’s criterion prefers path 2 to path 3 but cannot be applied to path 1. The joint criterion is applicable to all CDFs and chooses path 2 as the optimal one.

What is the shortest path?

In contrast with the deterministic case, there is no longer a unique concept of optimality.

Frank defines a path to be optimal if its cdf surpasses a threshold within the shortest time.

Fan suggests maximizing the cdf for a given time budget.

FIG. 1: Comparison of path optimality criteria using CDFs of three paths. Fan et al.’s criterion prefers paths 2 and 3 to path 1 but cannot discriminate between the CDFs of paths 2 and 3. Frank’s criterion prefers path 2 to path 3 but cannot be applied to path 1. The joint criterion is applicable to all CDFs and chooses path 2 as the optimal one.


What is the shortest path?

In contrast with the deterministic case, there is no longer a unique concept of optimality.

Frank defines a path to be optimal if its cdf surpasses a threshold within the shortest time.

Fan suggests maximizing the cdf for a given time budget.

![Comparison of path optimality criteria using CDFs of three paths.](image)

**FIG. 1:** Comparison of path optimality criteria using CDFs of three paths. Fan et al.’s criterion prefers paths 2 and 3 to path 1 but cannot discriminate between the CDFs of paths 2 and 3. Frank’s criterion prefers path 2 to path 3 but cannot be applied to path 1. The joint criterion is applicable to all CDFs and chooses path 2 as the optimal one.


What is the shortest path?

In contrast with the deterministic case, there is no longer a unique concept of optimality.

Frank defines a path to be optimal if its cdf surpasses a threshold within the shortest time.

Fan suggests maximizing the cdf for a given time budget.

Joint criterion: Frank criterion if it finds a path, otherwise we use Fan.

FIG. 1: Comparison of path optimality criteria using CDFs of three paths. Fan et al.'s criterion prefers paths 2 and 3 to path 1 but cannot discriminate between the CDFs of paths 2 and 3. Frank's criterion prefers path 2 to path 3 but cannot be applied to path 1. The joint criterion is applicable to all CDFs and chooses path 2 as the optimal one.

Finding the best path: A centralized algorithm

Routing table to reach a target $r$ from all other nodes, solution of

$$u_i(t) = \max_{j \in J_i} \left[ \int_0^t p_{ij}(t') u_j(t - t') \, dt' \right] ,$$  \hspace{1cm} (2)

$$u_r(t) = 1 ,$$  \hspace{1cm} (3)

$u_i(t)$ is the probability to arrive at node $r$ starting from node $i$ with a total time no longer than $t$

When on $i$, the node to chose to optimize the path is:

$$q_i(t) = \arg \max_{j \in J_i} \left[ \int_0^t p_{ij}(t') u_j(t - t') \, dt' \right]$$

Finding the best path: A centralized algorithm

Routing table to reach a target r from all other nodes, solution of

\[ u_i(t) = \max_{j \in J_i} \left[ \int_0^t p_{ij}(t') u_j(t - t') \, dt' \right], \quad (2) \]

\[ u_r(t) = 1, \quad (3) \]

\( u_{i}(t) \) is the probability to arrive at node \( r \) starting from node \( i \) with a total time no longer than \( t \)

When on \( i \), the node to chose to optimize the path is:

\[ q_i(t) = \arg \max_{j \in J_i} \left[ \int_0^t p_{ij}(t') u_j(t - t') \, dt' \right] \]

... can be solved recursively, but requires global knowledge of the system because of the recurrence

Finding the best path: A decentralized algorithm

We build an estimation function $f(i,j,t)$ that estimates the arrival probability between $i$ and $j$.

1. The network is embedded in a metric space. Let $d_{ij}$ the physical distance between two nodes
(proximity will help us to guide travellers)

Finding the best path: A decentralized algorithm

We build an estimation function $f(i,j,t)$ that estimates the arrival probability between $i$ and $j$.

1. The network is embedded in a metric space. Let $d_{ij}$ the physical distance between two nodes (proximity will help us to guide travellers)

2. Let $g_{ij}$ be the physical distance of the shortest path between two nodes, and assume that $g_{ij} \approx h(d_{ij})$

$$h(d_{ij}) \approx 0.547(8) \text{ km} + 1.1176(4) \times d_{ij}$$
Finding the best path: A decentralized algorithm

We build an estimation function $f(i,j,t)$ that estimates the arrival probability between $i$ and $j$.

1. The network is embedded in a metric space. Let $d_{ij}$ the physical distance between two nodes (proximity will help us to guide travellers)

2. Let $g_{ij}$ be the physical distance of the shortest path between two nodes, and assume that $g_{ij} \approx h(d_{ij})$

3. One estimates the number of steps between $i$ and $j$ by

$$
\bar{k} = \left[ \frac{g_{ij}}{\lambda} \right] \approx \left[ \frac{h(d_{ij})}{\lambda} \right]
$$

The weight on each step is chosen uniformly at random from the known weights associated with edges

$$
\bar{p}(t) = \frac{1}{m} \sum_{(i,j) \in E} p_{ij}(t)
$$
Finding the best path: A decentralized algorithm

Under these assumptions, we build the estimation function

\[ f(i, j; t) = \begin{cases} \int_0^t dt' \left( \prod_{k=1}^{\bar{k}} \bar{p} \right)(t'), & \text{if } i \neq j, \\ 1, & \text{if } i = j \end{cases} \]

estimating the probability for a path from \( i \) to \( j \) to have a weight smaller than \( t \)

Advantage: the estimation function is calculated by using properties of the spatial embedding alone, and does not involve the knowledge of the graph as a whole.
Finding the best path: A decentralized algorithm

Under these assumptions, we build the estimation function

\[ f(i, j; t) = \begin{cases} \int_0^t dt' \left( \sum_{k=1}^k P_k(t') \right), & \text{if } i \neq j, \\ 1, & \text{if } i = j \end{cases} \]

and use it to build a local algorithm, where we estimate the exact cdf for nodes where we have no information.
Numerical tests

Simulations on two-dimensional lattices with short-cuts (à la Kleinberg)

To determine edge weights, we assign lognormal PDFs $p_{\ln}(\mu, \sigma; t)$ with mean $\mu$ and standard deviation $\sigma$ chosen uniformly at random in the interval $[0.5, 1.5]$. 
Numerical tests

When budget increases, the joint criterion outperforms Fan criterion, as expected.

Simulations on two-dimensional lattices with short-cuts (à la Kleinberg)

To determine edge weights, we assign lognormal PDFs $p_{\ln}(\mu, \sigma; t)$ with mean $\mu$ and standard deviation $\sigma$ chosen uniformly at random in the interval $[0.5, 1.5]$. 

![Graph showing mean travel time of successful routing versus budget for different criteria.](image)
Numerical tests

Simulations on two-dimensional lattices with short-cuts (à la Kleinberg)

To determine edge weights, we assign lognormal PDFs $p_{ln}(\mu, \sigma; t)$ with mean $\mu$ and standard deviation $\sigma$ chosen uniformly at random in the interval $[0.5, 1.5]$.

![Graph](image)

FIG. 1: Comparison of path optimality criteria using CDFs of three paths. Fan et al.'s criterion prefers paths 2 and 3 to path 1 but cannot discriminate between the CDFs of paths 2 and 3. Frank's criterion prefers path 2 to path 3 but cannot be applied to path 1. The joint criterion is applicable to all CDFs and chooses path 2 as the optimal one.
Numerical tests

Simulations on two-dimensional lattices with short-cuts (à la Kleinberg)

To determine edge weights, we assign lognormal PDFs $p_{1n}(\mu, \sigma; t)$ with mean $\mu$ and standard deviation $\sigma$ chosen uniformly at random in the interval $[0.5, 1.5]$. When budget increases, the joint criterion outperforms Fan criterion, as expected.
Numerical tests

Chicago road network (with 542 junctions)

When budget increases, the joint criterion outperforms Fan criterion, as expected.
Computational time

Centralized algorithm: finding the shortest path between two nodes requires operations on the full network $\Rightarrow t \sim N^a$, with $a \approx 1$.

Decentralized algorithm: only local searches $\Rightarrow$ sublinear scaling $t \sim N^b$, with $b < 1$.

![Graph showing computational time vs. number of nodes for centralized and decentralized algorithms.](image)

- $a = 1.16$
- $b = 0.36$
Conclusion

Need for algorithms for temporal networks

Temporal network is modelled as a stochastic process, where each edge is assigned a probability distribution

Routing: centralized (more efficient) versus decentralized (faster) algorithms

Efficient algorithms for community detection, block modelling in such stochastic systems?
Possibility to take advantage of the observed temporal patterns to guide the development of algorithms?
Thanks to:

Till Hoffmann and Mason Porter (Oxford): Temporal networks

Martin Rosvall (Umea) and Andrea Lancichinetti (Northwestern): Pathways and Memory

J.C. Delvenne (Louvain), Luis Rocha (Louvain->Namur/Karolinska) and Lionel Tabourier (Namur): Burstiness

Financial support of IAP DYSCO from Belspo, Région Wallonne and FNRS