Le rôle de la dimension dans la recherche de chemins optimaux dans les petits mondes
George Giakkoupis, Nicolas Schabanel

To cite this version:
George Giakkoupis, Nicolas Schabanel. Le rôle de la dimension dans la recherche de chemins optimaux dans les petits mondes. 13es Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel), 2011, Cap Estérel, France. 4 p. hal-00587955

HAL Id: hal-00587955
https://hal.science/hal-00587955
Submitted on 23 Apr 2011

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.
Le rôle de la dimension dans la recherche de chemins optimaux dans les petits mondes

George Giakkoupis\(^2\) and Nicolas Schabanel\(^1,2,3\)

1 CNRS, France
2 Université Paris Diderot, LIAFA (UMR n°7089), France
3 IXXI, Université de Lyon, École normale supérieure de Lyon, France

Dans cet article, nous apporterons une réponse exacte (asymptotiquement) et surprenante à une question ouverte depuis plusieurs années dans le domaine de la modélisation des comportements sociaux : est-il possible de naviguer dans un graphe petit-monde de Kleinberg en temps optimal ? C’est-à-dire, est-il possible de suivre un chemin optimal d’un point à un autre en utilisant seulement les informations disponibles localement. Cette question a clairement des applications dans le design de tables de routage et de protocoles pair-à-pair. La réponse que nous apportons est la suivante : si la dimension sous-jacente du graphe petit-monde est 1, la réponse est non, les chemins optimaux sont de longueur \(\Theta(\log n)\) alors que le mieux que l’on puisse faire à partir des informations locales est \(\Theta(\log n \cdot \log \log n)\) : lorsque la dimension sous-jacente est \(\geq 2\), on démontre qu’un simple algorithme de parcours en largeur réussit à naviguer optimallement.

1 Introduction

Milgram a révélé dans son célèbre essai [Mil67] que non seulement les individus sont proches les uns des autres, mais qu’ils sont également capables de trouver de tels chemins courts entre eux, malgré leur vue locale extrêmement faible du réseau social mondial. En 2000, Kleinberg [Kle00] a proposé un modèle de réseau aléatoire simple qui capture cette propriété surprenante des réseaux sociaux. Outre cette motivation sociologique naturelle, son modèle a eu un impact important sur le design de plusieurs protocoles de pair à pair (e.g., [ZGG02]), car il aborde la question générale de comment les algorithmes distribués peuvent trouver de courts chemins dans des réseaux partiellement inconnus. Kleinberg’s small world model consists of a \(d\)-dimensional grid \(\{-n, \ldots, n\}^d\) (representing local acquaintance between individuals, such as geographic or professional) augmented with one “long-range” directed link per node pointing to a random node at distance \(r\) chosen with probability proportional to \(1/r^s\) where \(s\) is a parameter of the model (this long-range link represents a random acquaintance met in the past for instance). Kleinberg defined a decentralized routing algorithm as an algorithm that tries to route locally a message from a node (the source) to another (the target), that is to say, by visiting only neighbors (local or long-range) of already visited nodes (starting from the source). Kleinberg’s most striking result is that no decentralized algorithm can find short paths (i.e., of length \(\text{polylog}(n)\)) if \(s \neq d\), even when the diameter of the augmented graph is \(\Theta(\log n)\) as it was shown later on by [MN04, MN05]. Only when \(d = s\), decentralized algorithm may found short paths between random pairs. Indeed, the simple greedy algorithm that simply routes the message to the closest neighbor (local or long-range) of the current message holder computes paths of expected length \(O(\log^2 n)\) [Kle00]. Several decentralized algorithms [FGP04, LS05] have been proposed to compute better paths when \(s = d\) efficiently (i.e. by visiting at most \(\text{polylog}(n)\) nodes with high probability) and the best so far [LS05] for Kleinberg’s original model computes path of length \(O(\log n \log \log n)^2\) between arbitrary pairs, which is still significantly larger than the diameter \(\Theta(\log n)\) of the graph. The question of the ability of efficient decentralized algorithms to find optimal paths was then stated by Kleinberg in 2006 in its celebrated paper [Kle06] (open problem n°3).

We answer precisely to this question by first showing that if \(d = 1\) no efficient decentralized algorithm may found shorter paths than \(\Omega(\log n \log \log n)\) on expectation, and second, by providing a new efficient decentralized algorithm (largely inspired from the work of [LS05]) which computes optimal paths of expected...
length $O(\log n \log \log n)$ for $d = 1$ and $O(\log n)$ for $d \geq 2$, respectively. To our knowledge, this is the first time that such a transition in performance is observed in this model when $s = d$.

Path length is a critical indicator of performance in routing protocols (speed, fault tolerance,...). A lot of the peer-to-peer protocols (Symphony, Chord,...) so far use an augmented 1D ring, and our paper shows that performance might be improved by considering 2D rings instead, with a modest overhead cost since the degree of each node remains constant (5 for $d = 2$). We may also wonder if other networks (like the one in [FG10]) present similar transition where, even if the diameter is small and short paths can be computed by decentralized algorithms, diameter may not be obtained by any efficient decentralized algorithm.

2 Definitions and Main Results

The network. We consider the $d$-dimensional Kleinberg’s small world random network $\mathcal{G}_d$ with one long-range links per node [Kle00] defined as follows. The network consists of a $d$-dimensional toric lattice $L_d = \{-n, \ldots, 0, \ldots, n\}^d$ of $(2n+1)^d$ nodes, where each node $u$ has, in addition to its $2d$ neighbors in the lattice (its local contacts), an extra directed link pointing to a random node $v \neq u$ (its long-range contact) chosen independently with probability $1/(Z_d \cdot \beta_{|u-v|})$, where $|u-v|$ denotes the ($\ell_1$-)distance in the underlying toric lattice $L_d$ (i.e., $|u-v| = \sum_{i=1}^d |u_i - v_i|$), and $\beta_r$ denotes the size of the balls of radius $r$ in $L_d$ ($\beta_r = \# \{ u : |u| \leq r \}$), and $Z_d$ is the normalizing constant $Z_d = \sum_{|u| \leq 0} 1/\beta_{|u|} = \Theta(\log n)$.

Efficient decentralized routing algorithms. We study algorithms that compute a path to transmit a message or a file from a source to a target, along the local and (directed) long-range links of the network. Following Kleinberg’s definition, such an algorithm is decentralized if it navigates through the network using only local information to compute the path. Precisely, it has the knowledge 1) of the underlying lattice structure (the $d$-dimensional torus), 2) of the coordinates of the target in the lattice, and 3) of the nodes it has previously visited as well as their long-range contacts. But, crucially, 4) it can only visit nodes that are local or long-range contacts of previously visited nodes, and 5) does not know the long-range contacts of any node that has not yet been visited. However, 6) the algorithm (but not the path it computes) is authorized to travel backwards along any directed links it has already followed. As Kleinberg pointed out in [Kle02], this is a crucial component of human ability to find short paths: one can interpret point 6) as a web user pushing the back button, or an individual returning the letter to its previous holder (who wrote his address on the envelope before sending it). We ask furthermore the algorithm to be efficient, in the sense that it has to visit a number of intermediate nodes which is at most poly-logarithmic in the size of the network ($\log \log(n)$) with high probability, when computing a path between a pair of nodes. This is a standard requirement in social network where the size of the network is typically exponentially larger than the capacity of each individual. Efficient decentralized algorithms are thus likely to model the capacity of individuals to route messages in a social network.

Since each node has a constant degree $2d + 1$, the diameter of $\mathcal{G}_d$ is $\Omega(\log n)$. It was shown by [MN04] that the diameter of Kleinberg’s smallworld is indeed $\Theta(\log n)$ for all dimension $d$; their analysis suggests a decentralized algorithm that would visit $\Omega(\sqrt{n})$ nodes to compute a path of length $O(\log n)$ for each pair of nodes, which is obviously unrealistic in the framework of social network (where $n \approx 10^9$). [Kle00] initially showed that greedy routing (that always passes the message to the neighbor (local or long-range) of the current holder that is the closest to the target) computes path of expected length $O(\log^2 n)$ while visiting $O(\log^2 n)$ nodes. [MNW04] showed that all decentralized routing algorithm has indeed to visit at least $\Omega(\log^2 n)$ nodes on expectation to compute a path between a random pair of nodes. [LS05] proposed an efficient decentralized algorithm that computes near-optimal paths of expected length $O(\log n \log \log n)$ between each pair of nodes while visiting an optimal expected number of nodes $O(\log^2 n)$. The question of finding an efficient decentralized algorithm computing optimal paths of length $O(\log n)$ between any pair of nodes was finally formulated as the open problem $n^3/3$ by Kleinberg in its celebrated paper [Kle06]. The following theorem, which sums up our main results, solves precisely this problem.

\[ \text{Note that our results are still valid with } O(1) \text{ long-range links per node instead of one.} \]

\[ \text{with the notable exception of } [FG10] \text{ whose routing algorithm visits } 2^{O(\sqrt{n} \log n)} \text{ nodes.} \]
**The rôle de la dimension dans la recherche de chemins optimaux dans les petits mondes**

**Theorem 1 (Main results)**

1. For any dimension $d$ and any $\varepsilon > 0$, there is an efficient decentralized routing algorithm that computes a path of expected length at most $O(\log n \log \log n)$ between any pair of nodes $(s,t)$, while visiting at most $O(\log^{d+\varepsilon} n)$ intermediate nodes with high probability.

2. For any dimension $d \geq 2$, there is an efficient decentralized algorithm that computes paths of optimal expected length $O(\log n)$ between any pair of nodes, while visiting at most $O(\log^2 n)$ nodes with high probability.

3. For dimension $d = 1$, for all efficient decentralized routing algorithm $\mathcal{A}$, there exists a constant $c > 0$ and a pair of nodes $(s,t)$ such that the expected length of the path computed between $s$ and $t$ is at least $c \log n \log \log n$ (showing that (1) is asymptotically optimal for $d = 1$).

**Overview of previous results.** Our optimal decentralized routing algorithms (propositions 1 and 2 in Theorem 1) rely on simplifying and then improving the approach of [LS05]. The key steps of the results in [LS05] were obtained through rather complicated calculations. We obtain here a simple geometric rereading of their results that allows us to improve them and get an optimal routing algorithms. In our re-formulation of [LS05], if we forget about the technical details, their algorithm rely on the fact that in a Kleinberg’s network, the long-range link of any node $u$ has an equal probability $\frac{1}{\log n}$ to fall in each of the rings centered on $u$ and whose distance ranges in $[r, 2^{r+1}]$ for $i \in \{1, \ldots, \log n\}$. It follows that if $u$ at distance $r \in [2^{i}, 2^{i+1}] - 1$ from the target $t$, its long-range links has probability $\approx \frac{1}{\log n}$ to be closer to $t$ than $u$.

In order to get proposition 1, we now fix some $\varepsilon > 0$ and consider concentric rings of radius $\log^{(j)} n$, with $j \in \{1, \ldots, \log \log n\}$, centered on the target. Consider that the message has reached a node $u$ at distance $r \in [\log^{(j)} n, \log^{(j+1)} n]$ among which one has a positive constant probability to fall in the $i$th ring (the one that contains $t$) and thus has a positive constant probability to fall at distance at most $r/2$ from $t$. Routing the message from $u$ to this leaf and repeating the process until reaching $t$ yields a path whose expected length is at most $\approx \sum_{i=1}^{\log n} \log n \log \log n = \log n \cdot \log \log n \cdot H_{\log n} \approx \log n \cdot \log \log n^2$.

In order to get proposition 2, we improve the algorithm of the algorithm above by taking into account that the cardinal of the spheres grows at least linearly with their radius when $d \geq 2$. It follows from our calculation that the expansion factor of the BFS tree rooted at a node at distance $r \in [\log^{(d)} n, \log^{(d+1)} n]$ is now at least $\approx 1 + \sqrt{\frac{\log n}{\log \log n}}$. Thus, we just need to explore each BFS tree up to depth $h \approx (1 + \varepsilon) \log \log n \sqrt{\log n \log \log n}$ to gather, with constant probability, $\log^{1+\varepsilon} n$ leaves among which one long-range link points to a node $\log n$ times closer to the target. It follows that the expected length of the computed path between any pair of nodes is at most $\approx (1 + \varepsilon) \log \log n \sum_{i=1}^{\log \log n} \frac{1}{\sqrt{\log \log n}} \approx \log \log n \sum_{i=1}^{\log \log n} \frac{1}{\sqrt{\log \log n}} \approx \frac{1}{\sqrt{\varepsilon}} \sqrt{\log \log n \log n} \sqrt{\log \log n \log \log n} = \frac{1}{\varepsilon} \log n$. It follows that this exploration based efficient decentralized algorithm computes asymptotically optimal paths on expectation for any pair of nodes.

Our last result, proposition 3, shows that, even if it does not matches the diameter of the graph, no other efficient decentralized algorithms can do better when $d = 1$. Consider an efficient decentralized routing
algorithm which is bounded to visit at most \( m = O(\log^c n) \) nodes, for some constant \( c > 0 \). The intuition for this last result is the following. First, in a Kleinberg’s network, with high enough probability, none of the long-range contacts of any set of \( m \) nodes can be more than \( O(\log^{-c} n) \) times closer to the target than any node in \( S \). It follows that the algorithm has to visit at least one node in each concentric rings of radii \( m^i \) centered on the target. The first step consists of partitioning the underlying grid into concentric rings of radii \( m^i \), for \( i \in \{1, \ldots, \log_{9} n\} \), centered on the target. According to the above, no matter how high the exponent \( c \) is in \( m = O(\log^c n) \), the algorithm will need to go through \( \Omega(\log_{\log m} n) = \Omega(\log n / \log \log n) \) rings. The second step consists of proving by a coupling argument that the algorithm has almost no control over the first nodes it will visit each times it enters a ring. Indeed, we prove that since the algorithm visits at most \( m \) nodes in total, we can force the algorithm to enter each ring through a set of at most \( m^i \) nodes (the entry points), which are close to the farthest border of the ring from the target, and chosen randomly and independently of the algorithm. The third and last step consists then of bounding the extent of the BFS trees rooted on these entry points. Since the entry points are independent of the algorithm, we are left with a purely geometric analysis of these trees. This requires a finer analysis than for the routing algorithm we proposed before, since we need to bound precisely the total length of the possibly explored long-range links. By bounding the cumulated length of the long-range links used in these BFS trees, we are able to prove that, with high enough probability, none of these BFS trees reaches the next ring, closer to the target, before depth \( \geq \frac{\log \log \log n}{\log r} \), at least for the range of distances \( r \in [2^{\log^{0.9} n}, 2^{\log^{1.0} n}] \). We conclude that the expected length of the computed path is at least \( \sum_{r=\log_{9} \left(2^{\log^{0.9} n}\right)}^{\log_{9} \left(2^{\log^{1.0} n}\right)} \frac{\log \log \log n}{\log m^r} \geq \frac{\log \log \log n}{9 \log m} \left( H_{\log^{0.9} n/2 \log m}^{0.9} - H_{\log^{1.0} n/2 \log m}^{0.1} \right) \geq 0.8 \log \log n = \Omega(\log n \log \log n) \). It follows that no efficient decentralized routing algorithm can do better than our algorithm in proposition 1.1

Références


