Supplementary Document for "EARTH: Accelerating Spatiotemporal Network K-function-based Analytics"

Tsz Nam Chan[1], Leong Hou U[2], Yun Peng[3] and Jianliang Xu[1]

[1]Hong Kong Baptist University
[2]University of Macau
[3]Guangzhou University

Recently, Chan et al. [1] propose efficient algorithms for reducing the time complexity of computing a network K-function and generating a network K-function plot, which are closely related to this work. In this supplementary document, we will deeply discuss why these state-of-the-art methods are hard to be used for supporting a spatiotemporal network K-function (i.e., our work [2]).

To compute the network K-function for a location dataset $\mathbb{P} = \{p_1, p_2, ..., p_n\}$ (with size n) in a road network $G = (V, E)$, domain experts need to count all data points p_j that are within the spatial threshold s from each data point p_i (cf. Equation 1).

$$K_{\mathbb{P}}(s) = \sum_{p_i \in \mathbb{P}} \sum_{p_j \in \mathbb{P}} \mathcal{I}(d_G(p_i, p_j) \leq s) \tag{1}$$

where $d_G(p_i, p_j)$ and $\mathcal{I}$ denote the shortest path distance between p_i and p_j and the indicator function (cf. Equation 2 in [2]), respectively.

In order to efficiently compute a network K-function, Chan et al. [1] first expand the network K-function into the following expression.

$$\begin{aligned} K_{\mathbb{P}}(s) &= \sum_{\widehat{e} \in E} \sum_{p_i \in \mathbb{P}(\widehat{e})} \sum_{e \in E} \sum_{\substack{p_j \in \mathbb{P}(e) \\ p_j \neq p_i}} \mathcal{I}(d_G(p_i, p_j) \leq s) \\ &= \sum_{\widehat{e} \in E} \sum_{e \in E} C_{\mathbb{P}}^{(\widehat{e}, e)}(s) \end{aligned} \tag{2}$$

where $\mathbb{P}(e)$ is the set of data points in the edge e and $C_{\mathbb{P}}^{(\widehat{e}, e)}(s)$ denotes the $(\widehat{e}, e)$-count function (cf. Equation 3).

$$C_{\mathbb{P}}^{(\widehat{e}, e)}(s) = \sum_{p_i \in \mathbb{P}(\widehat{e})} \sum_{\substack{p_j \in \mathbb{P}(e) \\ p_j \neq p_i}} \mathcal{I}(d_G(p_i, p_j) \leq s) \tag{3}$$

Then, they propose two methods, namely count augmentation (CA) and neighbor sharing (NS), in order to reduce the time complexity for computing $C_{\mathbb{P}}^{(\widehat{e}, e)}(s)$, and thus

$K_{\mathbb{P}}(s)$. Here, we provide basic descriptions of these two methods and explain why they are hard to be extended for supporting spatiotemporal network K-function [2].

Count augmentation (CA): In this method, Chan et al. [1] aim to augment two aggregate terms, which are $|\mathbb{P}(p_j, u)|$ and $|\mathbb{P}(p_j, v)|$, for each data point p_j in each edge $e = (u, v)$ (cf. Figure 1), where

$$\mathbb{P}(p_j, u) = \{p \in \mathbb{P}(e) : d_G(u, p) \leq d_G(u, p_j)\} \tag{4}$$

$$\mathbb{P}(p_j, v) = \{p \in \mathbb{P}(e) : d_G(v, p) \leq d_G(v, p_j)\} \tag{5}$$

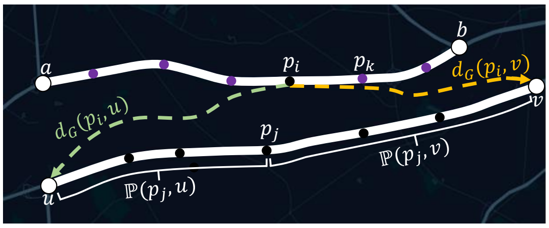

Figure 1: The core idea of the CA method (Modified from [1]).

With this augmentation, once they have obtained the shortest path distances from p_i to u and v, i.e., $d_G(p_i, u)$ (green dashed line) and $d_G(p_i, v)$ (orange dashed line), respectively, they can adopt the binary search method (with $O(\log |\mathbb{P}(e)|)$ time) to evaluate the inner summation term of $C_{\mathbb{P}}^{(\widehat{e},e)}(s)$ (cf. Equation 3). Based on this concept, they can efficiently evaluate $K_{\mathbb{P}}(s)$.

Since these aggregate terms, $\mathbb{P}(p_j, u)$ and $\mathbb{P}(p_j, v)$, do not consider the temporal part of the spatiotemporal network K-function (i.e., $d(\tau_{p_i}, \tau_{p_j})$ in Equation 1 of [2]), **the CA method cannot be used for computing the spatiotemporal network K-function**. Moreover, since the data point p_i in (a, b) (cf. Figure 1) may have any timestamp τ_{p_i}, $\mathbb{P}(p_j, u)$ and $\mathbb{P}(p_j, v)$ can possibly cover some data points with their timestamps τ_{p_j} that are far away from τ_{p_i}. Worse still, there can be multiple data points in the edge (a, b) (e.g., the purple data point p_k in Figure 1). It is **impossible to simply add some time constraints in Equation 4 and Equation 5 so that it can support the spatiotemporal network K-function.**

Neighbor sharing (NS): In the NS method, Chan et al. [1] propose to maintain four sets of data points (cf. Figure 2), which are $\ell_{au}(p_i)$, $\ell_{bu}(p_i)$, $\ell_{av}(p_i)$, and $\ell_{bv}(p_i)$ (cf. Equations 6 to 9).

$$\ell_{au}(p_i) = \{p_j \in \mathbb{P}(e) : d_G(u, p_j) \leq s_{au}(p_i)\} \tag{6}$$

$$\ell_{bu}(p_i) = \{p_j \in \mathbb{P}(e) : d_G(u, p_j) \leq s_{bu}(p_i)\} \tag{7}$$

$$\ell_{av}(p_i) = \{p_j \in \mathbb{P}(e) : d_G(v, p_j) \leq s_{av}(p_i)\} \tag{8}$$

$$\ell_{bv}(p_i) = \{p_j \in \mathbb{P}(e) : d_G(v, p_j) \leq s_{bv}(p_i)\} \tag{9}$$

where

$$s_{au}(p_i) = s - d_G(p_i, a) - d_G(a, u) \tag{10}$$

$$s_{bu}(p_i) = s - d_G(p_i, b) - d_G(b, u) \tag{11}$$

$$s_{av}(p_i) = s - d_G(p_i, a) - d_G(a, v) \tag{12}$$

$$s_{bv}(p_i) = s - d_G(p_i, b) - d_G(b, v) \tag{13}$$

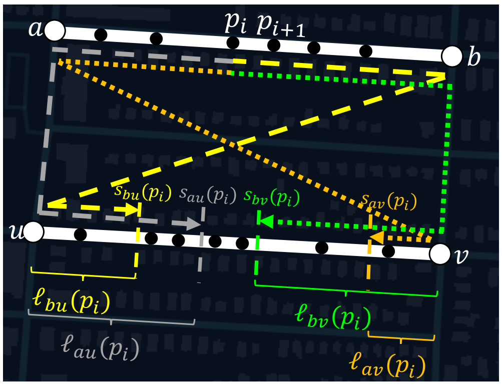

Figure 2: The core idea of the NS method (Modified from [1]).

With these four sets of data points, they show that the inner summation term of Equation 3 can be computed in $O(1)$ time. By iteratively shifting the data point from the data point p_i to the next data point p_{i+1} in the edge $\widehat{e} = (a, b)$, they show that these four sets can be efficiently maintained. As such, they can efficiently compute $C_{\mathbb{P}}^{(\widehat{e},e)}(s)$ (with $O(|\mathbb{P}(\widehat{e})| + |\mathbb{P}(e)|)$ time, instead of $O(|\mathbb{P}(\widehat{e})| \times |\mathbb{P}(e)|)$ time), and thus $K_{\mathbb{P}}(s)$.

Like the CA method, these four sets of data points (cf. Equations 6 to 9) do not consider the timestamps. Suppose that the data point p_i has the timestamp τ_{p_i}, these four sets may cover some data points p_j which have their timestamps τ_{p_j} that are far away from τ_{p_i}. As such, we **cannot adopt these four sets of data points to compute the spatiotemporal network K-function (cf. Equation 1 in [2]).** Furthermore, since different data points (e.g., p_i and p_{i+1} in the edge $\widehat{e} = (a, b)$ in Figure 2) may have different timestamps, we **cannot simply add time constraints in the four sets of data points in order to support the spatiotemporal network K-function.**

References

[1] T. N. Chan, L. H. U, Y. Peng, B. Choi, and J. Xu. Fast network k-function-based spatial analysis. *Proc. VLDB Endow.*, 15(11):2853–2866, 2022.

[2] T. N. Chan, L. H. U, Y. Peng, and J. Xu. EARTH: Accelerating spatiotemporal network k-function-based analytics. *ICDE 2024 (In submission)*.