Efficient Identification of Critical Links Based on Reachability Under the Presence of Time Constraint

Kazumi Saito[1,2](✉), Kouzou Ohara[3], Masahiro Kimura[4], and Hiroshi Motoda[5]

[1] Faculty of Science, Kanagawa University, Hiratsuka, Japan
k-saito@kanagawa-u.ac.jp

[2] Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan
kazumi.saito@riken.jp

[3] College of Science and Engineering, Aoyama Gakuin University, Sagamihara, Japan
ohara@it.aoyama.ac.jp

[4] Department of Electronics and Informatics, Ryukoku University, Kyoto, Japan
kimura@rins.ryukoku.ac.jp

[5] Institute of Scientific and Industrial Research, Osaka University, Suita, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. In this paper, we focus on an emergency situation in the real-world such as disaster evacuation and propose an algorithm that can efficiently identify critical links in a spatial network that substantially degrade network performance if they fail to function. For that purpose, we quantify the network performance by node reachability from/to one of target facilities within the prespecified time limitation, which corresponds to the number of people who can safely evacuate in a disaster. Using a real-world road network and geographical information of actual facilities, we demonstrated that the proposed method is much more efficient than the method based on the betweenness centrality that is one of the representative centrality measures and that the critical links detected by our method cannot be identified by using a straightforward extension of the betweenness centrality.

1 Introduction

Identification of critical links in a large complex network is an important research issue, and has been explored from various points of view such as contamination spread, wireless sensor network, information flow and disaster evacuation [1,8,13,17,18,20]. Recently, considerable attention has been devoted to analyzing real-world spatial networks [2,4,5,10,14,15,22]. We focus on spatial networks embedded in the real space, *e.g.*, urban streets, and address the problem of identifying critical links.

A critical link in a network is such a link that exerts a substantial effect on the network performance when the link fails to function properly. Thus, given a network structure and a performance measure, critical links can be found by solving the corresponding optimization problem, that is, solving the problem

A. C. Nayak and A. Sharma (Eds.): PRICAI 2019, LNAI 11671, pp. 404–418, 2019.
https://doi.org/10.1007/978-3-030-29911-8_31

of finding a link such that its blocking maximally degrades the network performance. As a motivating example, let us consider managing an urban road network from the perspective of an emergency situation such as disaster evacuation, ambulance call and fire engine dispatch. In these emergency situation, time is critical. Thus, the performance measure must take time criticalness into account. To the best of our knowledge, there has been no work that considered time criticalness in computing the performance measure.

As preliminaries, we assume that each node in the network represents the population around it, the traveling time $t(e; G)$ or the real distance $dist(e; G)$ in the real-world is assigned to each link e, and there exists a set of target nodes $\boldsymbol{\mathcal{U}}$. Note that $t(e; G)$ depends on means of transportation but $dist(e; G)$ does not. In case of disaster evacuation, $\boldsymbol{\mathcal{U}}$ consists of the evacuation facilities, and people living in the neighborhood of a node v evacuate to the facility $u \in \boldsymbol{\mathcal{U}}$ nearest to v. In case of ambulance call, $\boldsymbol{\mathcal{U}}$ consists of the emergency hospitals, and people living in the neighborhood of a node v are taken to the hospital $u \in \boldsymbol{\mathcal{U}}$ nearest to v. In case of fire engine dispatch, $\boldsymbol{\mathcal{U}}$ consists of the fire stations, and when fire breaks out, fire engines are dispatched from the station $u \in \boldsymbol{\mathcal{U}}$ to the nearest v (nearest to the scene of the fire). Note that each node is associated with people/houses around it. Thus, the number of people/houses varies according to the node. Our previous work [18] employed the network performance measure based on node reachability, and identified critical links in the same setting, but it igonored time criticalness, *i.e.* it simply assumed that there is no time limit for going to/from $\boldsymbol{\mathcal{U}}$. In this paper, we place a time limit that is allowed for the emergency actions to be completed.

Technically, in order to reflect more realistic emergency situations, given a road network G, a set of target nodes $\boldsymbol{\mathcal{U}}$ and a maximum permissible time τ, we adopt the number of people/houses that are reachable from $\boldsymbol{\mathcal{U}}$ within time τ as the network performance measure, and consider identifying the critical links. Here, we suppose that each node v of G represents a collection of people/houses and the traveling time $t(e; G)$ of each link e of G is given. As mentioned above, the critical links are defined by an optimization problem, where the objective function value $h(e; \tau, \boldsymbol{\mathcal{U}}, G)$ for a link e measures the reduction of the number of people/houses that are reachable from $\boldsymbol{\mathcal{U}}$ within time τ when e is blocked. Thus, for each link e, we can regard $h(e; \tau, \boldsymbol{\mathcal{U}}, G)$ as the criticalness of e. In this paper, we refer to it as the *time-bounded criticalness centrality* of e, and consider efficiently calculating the values of $h(e; \tau, \boldsymbol{\mathcal{U}}, G)$ for all the links of G.

If there is no time constraint imposed, the standard bridge detection technique [21] could be successfully applied to the efficient calculation of $\{h(e; \tau, \boldsymbol{\mathcal{U}}, G) \mid e$ is a link of $G\}$. In fact, a link in a connected component is called a bridge if its deletion divides the component into two connected components, and we can expect that $\{h(e; \tau, \boldsymbol{\mathcal{U}}, G)\}$ is efficiently calculated by checking whether each component intersects with $\boldsymbol{\mathcal{U}}$ or not. For example, Sariyuce et al. [19] successfully utilized the bridge detection technique to efficiently calculate conventional topological centrality measures. However, calculation of $h(e; \tau, \boldsymbol{\mathcal{U}}, G)$ requires both the traveling time information and the time constraint. In this paper, we propose a method that can efficiently calculate $\{h(e; \tau, \boldsymbol{\mathcal{U}}, G)\}$ based on the Best-First Search consisting of two phases: Forest Construction phase and Local Update phase.

Below we summarize our contributions in this paper. (1) We formulated the problem of identifying critical links in a road network considering the traveling time needed for each link under the presence of time constraint from the viewpoint of an emergency situation. (2) We defined the concept of time-bounded criticalness centrality for a link in terms of the reachability from a set of target nodes within a maximum permissible time to identify the critical links, and proposed an efficient method of calculating the centrality for all links. (3) We experimentally demonstrated the effectiveness of the proposed method for the road network in Tokyo in case of disaster evacuation, ambulance call and fire engine dispatch, showing that the proposed method is much faster than computing the major traditional centrality measures, and the critical links cannot be properly identified by the measure derived by straightforwardly extending a traditional centrality measure. Moreover, we revealed the characteristics of the critical links by varying the maximum permissible time.

The paper is organized as follows. After briefly describing the related work in Sect. 2, we mathematically formulate the critical link detection problem in Sect. 3. We present the proposed method in Sect. 4, and report the experimental results in Sect. 5. We give our conclusion in Sect. 6.

2 Related Work

Various kinds of centrality measures have been proposed so far [3]. The centrality measure we propose in this paper is considered as an extension of vitality index that is defined as the difference of the values of a real-valued function f on a simple, undirected and unweighted graph G and on G without an element x, where x is either a node or a link in G [9]. For example, closeness vitality adopts a function that returns the total sum of distances between arbitrary pair of nodes in a given graph. Similarly, our centrality measure is defined as the difference of a real-valued function on G and on G without a link e. But, our proposed measure is different from the original vitality index in that it is defined for an weighted spatial network considering the time constraint instead of simply taking into account network topology.

As mentioned in the previous section, several studies have been made on analyzing large spatial networks. Burckhart and Martin [4] and Wang et al. [22] explored traffic usage patterns in urban streets. Crucitti et al. [5] and Park and Yilmaz [15] examined the topological properties of road networks in terms of conventional centrality measures such as degree centrality, closeness centrality and betweenness centrality. Montis et al. [10], Opsahl et al. [14] and Ohara et al. [11] analyzed road networks considering the usage frequency or the real distance for each link by extending the conventional centrality measures. Ohara et al. [12] also extended the group closeness centrality to such a road network, and investigated the problem of inserting k new links to maximize it. There are studies that are not specific to spatial network but discuss the importance of links in complex networks. Grady et al. [7] presented a notion of salient links in complex networks, which are obtained by superimposing shortest path trees for each node. Piraveenan et al. [16] proposed a new centrality based on percolation states of individual nodes, which is referred to as percolation centrality. Fang et al. [6] compared link capacities allocation models that makes a power transmission network resilient to cascading failures with limited investment costs.

In this paper, we deal with the problem of identifying critical links for emergency situations, and define the concept of time-bounded criticalness centrality. We adopt link detection methods based on the conventional centrality measures as baselines to compare with the proposed method.

There are several investigations that are related to identifying critical links in road networks. For example, Oliveira et al. [13] explored the vulnerability of links in a road network using a transportation simulation software. Saito et al. [18] examined the critical link detection problem without time constraint in a situation where links are probabilistically blocked. Our current work is different from these studies. In order to take more realistic emergency situations into account, we focus on the problem of identifying critical links in a road network that maximize the degradation of network performance measure under the time constraint considering the time needed to pass each link. To the best of our knowledge, there is no work that utilizes the number of people/houses that are reachable from a set of target nodes within a maximum permissible time as the network performance measure, and provides an efficient method of calculating the time-bounded critical centrality for a road network. Note that it is straightforward to extend the proposed method to the situation where links are probabilistically disconnected.

3 Problem Formulation

Let $G = (\mathcal{V}, \mathcal{E})$ be a given simple undirected (or bidirectional) network without self-loops, where $\mathcal{V} = \{u, v, w, \cdots\}$ and $\mathcal{E} = \{e, \cdots\}$ are sets of nodes and undirected links, respectively. We also express each link e as a pair of nodes, *i.e.*, $e = (u, v)$. For each link $e = (u, v) \in \mathcal{E}$, we assign its traveling time $t(u, v)$ between these nodes. For each pair of nodes that does not have the direct connection, *i.e.*, $(u, w) \notin \mathcal{E}$, we define its traveling time $t(u, w; G)$ by the minimum traveling time over all possible paths between them. In our problem setting, we assume a fixed group of nodes $\mathcal{U} \subset \mathcal{V}$ such as evacuation facilities, emergency hospitals or fire stations on a road network. Here, for each node $v \in \mathcal{V}$, we can compute the minimum traveling time $f(v; \mathcal{U}, G)$ between v and some node $u \in \mathcal{U}$ as follows:

$$f(v; \mathcal{U}, G) = \min_{u \in \mathcal{U}} t(v, u; G). \tag{1}$$

Hereafter in case that v is not reachable to any node $u \in \mathcal{U}$, we assume $f(v; \mathcal{U}, G) = \infty$ for our convention.

Now, in case of a disaster such as tsunami right after a large-scale earthquake, people living in the flooded area must evacuate to some facility before the time of tsunami arrival. Let τ be such a maximum permissible time, and for each node $v \in \mathcal{V}$, we assume that v has some weight denoted by $\rho(v)$ which is intended to represent the number of residences or houses around node v in a road network. Then, we can compute the following weighted sum of nodes whose traveling times from/to some facility are less than a maximum permissible time.

$$g(\tau; \mathcal{U}, G) = \sum_{\{v \in \mathcal{V} \ | \ f(v; \mathcal{U}, G) \leq \tau\}} \rho(v). \tag{2}$$

Hereafter, $g(\tau; \mathcal{U}, G) / \sum_{v \in \mathcal{V}} \rho(v)$ is referred to as a success ratio for time τ.

For each graph G, let $G(e)$ be a graph constructed by deleting a link e, *i.e.*, $G(e) = (\mathcal{V}, \mathcal{E} \setminus \{e\})$. Then, we can define the following success contribution value of a link $e \in \mathcal{E}$ over G.

$$h(e; \tau, \mathcal{U}, G) = g(\tau; \mathcal{U}, G) - g(\tau; \mathcal{U}, G(e)). \tag{3}$$

Note that the value $h(e; \tau, \mathcal{U}, G)$ can be interpreted as the weighted sum of nodes, *i.e.*, people, who become unable to move to one of these evacuation facilities within a maximum permissible time when a link e is blocked in case of evacuation problem. Similar interpretation is possible for other problems. Hereafter, the measure defined in Eq. (3) is referred to as time-bounded criticalness centrality.

4 Proposed Method

For a given spatial network $G = (\mathcal{V}, \mathcal{E})$ with a weight $\rho(v)$ for each node $v \in \mathcal{V}$ and a traveling time $t(u, v)$ for each link $(u, v) \in \mathcal{E}$, together with a fixed group of nodes $\mathcal{U} \subset \mathcal{V}$ and a maximum permissible time τ, we describe our proposed algorithm for computing the time-bounded criticalness centrality value $h(e; \tau, \mathcal{U}, G)$ defined in Eq. (3) for each link $(v, w) \in \mathcal{E}$. Our algorithm based on the best-first search strategy consists of two phases: construction of a directed, rooted forest whose roots are nodes in $\mathcal{U}$ and computation of centrality values by locally updating the time $f(\hat{v}; \mathcal{U}, G(e))$. Below the former and latter phases are referred to as FC (forest construction) and LU (local update), respectively.

For a subset of nodes $\mathcal{W} \subset \mathcal{V}$, let $\mathcal{I}(\mathcal{W})$ be a set of incident links from $\mathcal{W}$, *i.e.*, $\mathcal{I}(\mathcal{W}) = \{(w, x) \in \mathcal{E} \mid w \in \mathcal{W}, x \notin \mathcal{W}\}$. Then, we can summarize our algorithm for computing the directed, rooted forest obtained as sets of nodes and links, $\mathcal{W}$ and $\mathcal{F}$, as follows.

FC1: Initialize $\mathcal{W} \leftarrow \mathcal{U}$, $\mathcal{F} \leftarrow \emptyset$, and $f(w; \mathcal{U}, G) \leftarrow 0$ for each $w \in \mathcal{U}$.
FC2: Select the best-first link $(\hat{w}, \hat{x})$ computed by $(\hat{w}, \hat{x}) \leftarrow \mathrm{argmin}_{(w,x) \in \mathcal{I}(\mathcal{W})} \{f(w; \mathcal{U}, G) + t(w, x)\}$, and set $\mu \leftarrow f(\hat{w}; \mathcal{U}, G) + t(\hat{w}, \hat{x})$.
FC3: Output the forest $(\mathcal{W}, \mathcal{F})$ if $\mu > \tau$, and then terminate.
FC4: Set $\mathcal{W} \leftarrow \mathcal{W} \cup \{\hat{x}\}$, $\mathcal{F} \leftarrow \mathcal{F} \cup \{(\hat{w}, \hat{x})\}$, and $f(\hat{x}; \mathcal{U}, G) \leftarrow \mu$, and go to FC2.

Here note that the links with positive centrality values are limited to those in $\mathcal{F}$, *i.e.*, $h(e; \tau, \mathcal{U}, G) = 0$ if $e \in \mathcal{E} \setminus \mathcal{F}$. Figure 1 illustrates construction of a directed, rooted forest from a road network, in which star shaped markers denote target nodes such as evacuation facilities. In each rooted tree in the forest, its root node is a unique target node and it is the nearest target node from the other nodes below in the tree, *i.e.*, all the nodes below can get to the target node within a given maximum permissible time. Indeed, the path between an internal/leaf node and the root node in the tree is the optimum path on the road network to reach the target node with minimum traveling time, which implies that any other link that does not appear in the forest (broken lines in Fig. 1) never affects the minimum traveling time for any node if it is blocked.

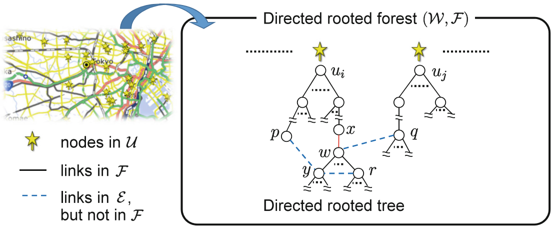

Directed rooted forest $(\mathcal{W},\mathcal{F})$
u_i
u_j
p
x
q
w
y
r
Directed rooted tree
nodes in $\mathcal{U}$
links in $\mathcal{F}$
links in $\mathcal{E}$, but not in $\mathcal{F}$

Fig. 1. Construction of a directed, rooted forest from a spatial network and local update of traveling time.

Next, let $\mathcal{X}(w)$ be a set of descendant nodes of w over the forest $(\mathcal{W},\mathcal{F})$. Here note that when a link $e = (w, x) \in \mathcal{F}$ is removed, we need to update the time $f(y;\mathcal{U},G(e))$ only for $y \in \mathcal{X}(w)$, *i.e.*, $f(y;\mathcal{U},G(e)) = f(y;\mathcal{U},G)$ for $y \in \mathcal{W} \setminus \mathcal{X}(w)$. For example, if we remove link (x, w) in Fig. 1, the subtree whose root node is w consists of $\mathcal{X}(w)$ and every node in $\mathcal{X}(w)$ becomes unable to reach their nearest target node u_i with the minimum traveling time. But, if any node in the subtree is connected to a node in $\mathcal{W}$ with a link in $\mathcal{E} \setminus \mathcal{F}$ just like node w is connected to node q with link (w, q) in Fig. 1, some nodes in $\mathcal{X}(w)$ may still reach a target node within a given maximum permissible time τ. The local update phase seeks such nodes while updating the minimum traveling time of each node in $\mathcal{X}(w)$. Note that link (y, r) that is not contained in $\mathcal{F}$ has to be tested to make a path from node r to node u_i if node y can reach node u_i within τ by passing link (y, p). Then, we can summarize our algorithm for computing the centrality values for each link $e \in \mathcal{F}$ by locally updating the time $f(y;\mathcal{U},G(e))$ for $y \in \mathcal{X}(w)$, as follows.

LU1: Repeat the following steps described in LU2 for every link $e = (w, x) \in \mathcal{F}$.

LU2.1: Initialize $\mathcal{Y} \leftarrow \mathcal{X}(w)$.

LU2.2: Select the best-first link $(\hat{y}, \hat{v})$ computed by $(\hat{y}, \hat{v}) \leftarrow \operatorname{argmin}_{(y,v)\in\mathcal{I}(\mathcal{Y})\setminus\{(x,w)\}}\{f(v;\mathcal{U},G(e)) + t(v, y)\}$, and set $\mu \leftarrow f(\hat{v};\mathcal{U}, G(e)) + t(\hat{v}, \hat{y})$.

LU2.3: Compute $h(e;\tau,\mathcal{U},G) \leftarrow \sum_{y\in\mathcal{Y}} \rho(y)$ if $\mu > \tau$, and then goto LU3.

LU2.4: Set $\mathcal{Y} \leftarrow \mathcal{Y} \setminus \{\hat{y}\}$ and $f(\hat{y};\mathcal{U},G(e)) \leftarrow \mu$, and go to LU2.2.

LU3: Output $h(e;\tau,\mathcal{U},G)$ for $e \in \mathcal{F}$ and 0 for $e \in \mathcal{E} \setminus \mathcal{F}$.

Evidently, the efficiency of our proposed algorithm is affected by several factors including the maximum permissible time τ, the number of nodes in $\mathcal{U}$ and so on. Thus, we evaluate the performance of our proposed algorithm which is referred to as TBC (time-bounded criticalness) in our computational experiments.

5 Experiments

Using real data of road network $G = (\mathcal{V}, \mathcal{E})$ and facilities $\mathcal{U}$, we evaluated the effectiveness of the proposed TBC algorithm.

5.1 Experimental Settings

In our experiments, we used the actual road network of Tokyo in Japan as $G = (\mathcal{V}, \mathcal{E})$, and considered three different realistic scenarios, *i.e.*, disaster evacuation, ambulance call, and fire engine dispatch. The spatial road network of Tokyo was extracted from the Open Street Map data[1]. As a matter of fact, we extracted all highways and all nodes, and constructed the spatial network by using the ends, intersections, and curve-fitting-points as nodes and the streets between them as links. The resulting network consists of $|\mathcal{V}| = 6,571,077$ nodes and $|\mathcal{E}| = 7,312,007$ links. For the facilities $\mathcal{U}$ of each scenario, we gathered geographical information about evacuation facilities, emergency hospitals, and fire stations, respectively, from the site of National Land Information Division of Ministry of Land, Infrastructure, Transport and Tourism (MLIT) of Japan[2], and mapped each of them to the nearest node in the spatial network. The numbers of evacuation facilities, emergency hospitals, and fire engines are $3,919$, 55, and 318, respectively. In this work, we assumed that a person moves at 1 m per second (3.6 km/h) on foot in the case of disaster evacuation, and that both an ambulance and a fire engine move at 10 m per second (36 km/h). We computed the traveling time $t(e; G)$ by dividing the distance $dist(e; G)$ by the velocity corresponding to each scenario. Here we should emphasize that we can arbitrary change $t(e; G)$ according to the other conditions such as road width. We used the 2015 census population aggregation data[3] for weight $\rho(v)$ setting of each node. More specifically, for a given population number n of each small region containing a subset of nodes $\mathcal{W} \subset \mathcal{V}$, we assigned to each node $v \in \mathcal{W}$ its average number as $\rho(v) = n/|\mathcal{W}|$.

Since a link between nodes having a high centrality value is considered to play an important role in the network, we used the straightforward extension of conventional link-centrality measures as the baselines to be compared with our proposed TBC centrality. More specifically, by incorporating $\mathcal{U}$ and $\rho(x)$ for $x \in \mathcal{V}$, we extended betweenness centrality and assigned a link-centrality value given by

[1] https://openstreetmap.jp/.
[2] http://nlftp.mlit.go.jp/ksj-e/index.html.
[3] https://www.e-stat.go.jp/gis/.

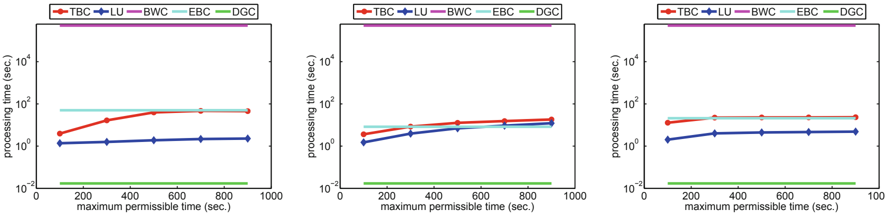
$$EBC(e) = \sum_{x \in \mathcal{V}} \frac{N^{sp}(x, y(x); e)}{N^{sp}(x, y(x))} \rho(x), \qquad \text{where} \qquad y(x) = \operatorname*{argmin}_{u \in \mathcal{U}} t(x, u; G). \quad (4)$$

$N^{sp}(x, y)$ stands for the number of the shortest paths from node x to node y, and $N^{sp}(x, y; e)$ denotes the number of those paths passing through a node e. Hereafter, we refer to this link-centrality as *EBC*. Here, we can naturally interpret $EBC(e)$ as the number of people passing through the link e during the movement to $\mathcal{U}$. All programs to compute link-centralities including TBC were implemented in C and run on a computer with a single thread (Xeon X5690 3.47 GHz CPUs) within a 192 GB main memory capacity.

5.2 Experimental Results

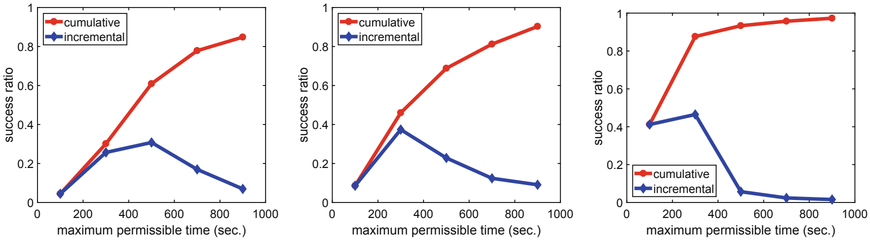
First, we evaluated the computational efficiency of our proposed TBC algorithm. To this end, we compared the processing time to compute the TBC centrality value for all links in $\mathcal{E}$ with the time to compute EBC for all links in $\mathcal{E}$, and the time to compute DGC and BWC for all nodes in $\mathcal{V}$, where DGC and BWC stand for standard degree centrality and betweenness centrality, respectively. Figure 2 shows the experimental results, where Figs. 2(a), (b), and (c) are cases using evacuation facilities, emergency hospitals, and fire stations as $\mathcal{U}$, respectively. To investigate the computational efficiency of the TBC algorithm more in depth, we further plotted the processing time spent for its second phase LU in each figure. In this experiment, we varied the maximum permissible time τ from 100 s to 900 s by 200 s. Note that the processing time of EBC, DGC and BWC is constant as their computation is not affected by the value of τ. It was 0.017 s for DGC and $513,525.800$ s for BWC regardless of the cases, while 50.13, 8.25, and 20.95 s for EBC with respect to the cases of evacuation facilities, emergency hospitals, and fire stations, respectively. From these results, it is obvious that the computation of DGC is the most efficient because it does not require to traverse the network to compute the centrality for every node $v \in \mathcal{V}$. The computation for EBC is modest and comparable to those of TBC. On the other hand, the computation of BWC is the most expensive. This is because its computational complexity approximately becomes a square order of the network size. Compared to BWC, the processing time for computing TBC is substantially small and less than 100 s even for a large value of the maximum permissible time τ in any scenario although it is much larger than that for DGC. This is attributed to the effect of the maximum permissible time which bounds the size of trees in the forest constructed at the first phase of the TBC algorithm. Besides, considering that the vertical axes of these figures are logarithmic, we see that most of the computation time is spent on the forest construction phase, FC, which is the difference between TBC and LU in Fig. 2, implying that the local update algorithm is quite efficient. Here, as naturally expected, the computation for EBC was comparable to those of FC. Actually, the processing time for FC depends on the size of $\mathcal{U}$, *i.e.*, $3,919$ for evacuation facilities, 318 for fire stations, and 55 for emergency hospitals. This is because the size of $\mathcal{U}$ is equivalent to the number of trees in the forest. In short, we can say that our proposed TBC algorithm has a desirable scaling property with respect to the number of facilities and the maximum permissible time.

Next, we investigated how the success ratio defined in Sect. 3 changes as the maximum permissible time τ gets longer. Indeed, in this experiment, the success ratio is given by $|\{v \in \mathcal{V} \mid f(v;\mathcal{U},G) \leq \tau\}|/|\mathcal{V}|$, where $f(v;\mathcal{U},G)$ is the minimum traveling time between v and some node $u \in \mathcal{U}$ defined by Eq. (1). We plotted this value as "cumulative" in Fig. 3, where Figs. 3(a), (b), and (c) are those in case of using evacuation facilities, emergency hospitals, and fire stations as $\mathcal{U}$. Also in this experiment, we varied the maximum permissible time τ from 100 (s) to 900 (s) by 200 (s). Note that the value represented by "incremental" in Fig. 3 is the difference between the success ratios at τ and at $\tau - 200$ for $\tau \geq 300$. From these results, we can observe that the success ratio monotonically increases as the maximum permissible time τ becomes longer. It is noted that the success ratio for $\tau = 100$ is less than 0.1 in Fig. 3(a). This is because we assumed that a person moves at 1 m per second on foot in the scenario of disaster evacuation, and thus, it is difficult for most people to reach the nearest evacuation facility within 100 s. This difficulty is alleviated by setting τ to a larger value and the success ratio improves accordingly. On the other hand, the "incremental" value decreases when τ becomes larger than 500 in Fig. 3(a) due to the fact that the number of people who need such a long traveling time to reach the nearest evacuation facility is limited. Similar tendency can be observed in the cases of ambulance call and fire engine dispatch in Figs. 3(b) and (c), where we assumed that both an ambulance and a fire engine move at 10 m per second. But, the success ratios for these scenarios when $\tau = 100$ are quite different, less than 0.1 for ambulance call as shown in Fig. 3(b) and around 0.4 for fire engine dispatch as shown in Fig. 3(c) because of the difference in the number of facilities. It is 318 for fire stations, while only 55 for emergency hospitals. Thus, the time limitation of 100 s becomes harder for ambulance call. In addition, we can find that the success ratio shown in Fig. 3 somewhat correlates to the processing time for computing TBC shown in Fig. 2.



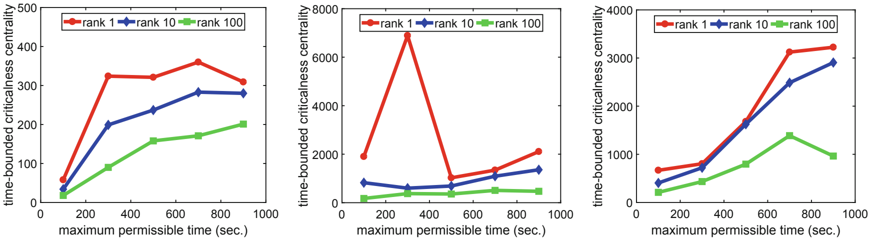
(a) Evacuation facilities (b) Emergency hospitals (c) Fire stations

Fig. 2. Evaluation of computational efficiency as a function of maximum permissible time.



(a) Evacuation facilities (b) Emergency hospitals (c) Fire stations

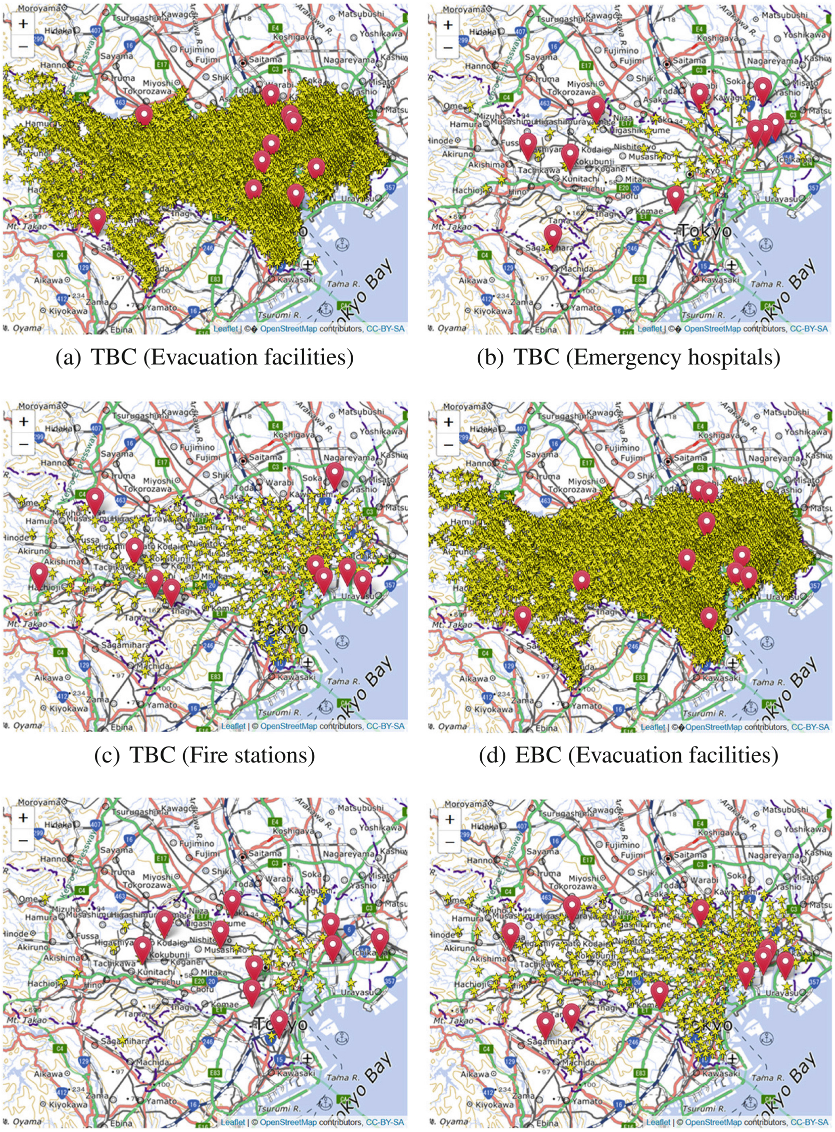
Fig. 3. Comparison with success ratio as a function of maximum permissible time.



(a) Evacuation facilities (b) Emergency hospitals (c) Fire stations

Fig. 4. Evaluation of time-bounded criticalness centrality values of the rank 1, rank 10 and rank 100 critical links as a function of maximum permissible time.

Next, we examined how the TBC centrality values of the detected critical links fluctuate as a function of the maximum permissible time τ. Figures 4(a), (b), and (c) show the results for the scenarios of disaster evacuation, ambulance call, and fire engine dispatch, respectively. In these figures, we plotted the TBC centrality values of the rank 1, rank 10, and rank 100 critical links. Again, we adopted the same values for τ as the ones used in the previous experiments. Note that the ranking of critical links may change according to the value of τ. Roughly speaking, the TBC centrality value of the critical links tends to increase as the maximum permissible time τ gets longer, but it does not necessarily monotonically change. This is because, according to the definition of Eq. (3), the TBC centrality value for a link $e \in \mathcal{E}$ is given by $|\{v \mid f(v; \mathcal{U}, G) \leq \tau \wedge f(v; \mathcal{U}, G(e)) > \tau\}|$ in this experiment, and $|\{v \mid f(v; \mathcal{U}, G) \leq \tau\}|$ is non-decreasing with respect to the value of τ, while $|\{v \mid f(v; \mathcal{U}, G(e)) > \tau\}|$ is non-increasing with respect to the value of τ. In fact, the TBC centrality value of the highly ranked critical links is likely to largely fluctuate when the number of facilities considered is relatively small as shown in Fig. 4(b) because the number of nodes $v \in \mathcal{V}$ that satisfy the condition $f(v; \mathcal{U}, G(e)) > \tau$ may largely fluctuate depending on the actual location of each facility $u \in \mathcal{U}$ in the network.



(a) TBC (Evacuation facilities)

(b) TBC (Emergency hospitals)

(c) TBC (Fire stations)

(d) EBC (Evacuation facilities)

(e) EBC (Emergency hospitals)

(f) EBC (Fire stations)

Fig. 5. Actual locations of the top-10 critical links by each of the TBC and EBC centralities.

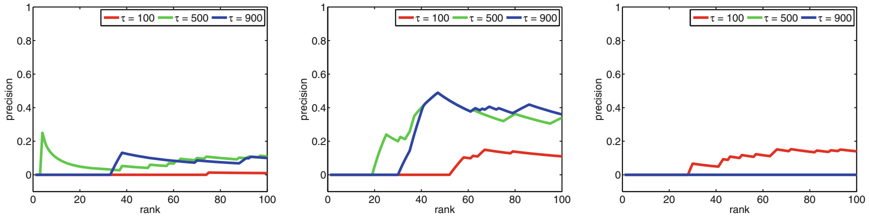
To qualitatively compare the top-10 regions containing critical links detected by different centrality measures, we marked them on an actual map of Tokyo as shown in Fig. 5, where each marker could indicate several consecutive links. Figures 5(a), (b), and (c) respectively show the top-10 critical links detected based on the TBC centrality for the scenarios of disaster evacuation, ambulance call, and fire engine dispatch in the case of $\tau = 500$. Figures 5(d), (e) and (f) show such top-10 regions detected by the values of EBC for those scenarios, respectively. From these figures, we can say that the links detected based on EBC are substantially different from those based on the TBC centrality. This implies that EBC cannot serve as an alternative to the TBC centrality.

To quantitatively confirm our claim, we further investigated the similarity of the ranking based on TBC and EBC in case of $\tau = 100$, 500 and 900 for our three scenarios. More specifically, we measured the similarity between the top k ranked nodes based on TBC, denoted by $\mathscr{A}_k^{(\tau)} = \{v_i\}_{i=1}^k$, and those based on EBC, denoted by $\mathscr{B}_k = \{v_i\}_{i=1}^k$, by using the precision $Prec_k^{(\tau)}$ defined as follows:

$$Prec_k^{(\tau)} = \frac{|\mathscr{A}_k^{(\tau)} \cap \mathscr{B}_k|}{k}.$$

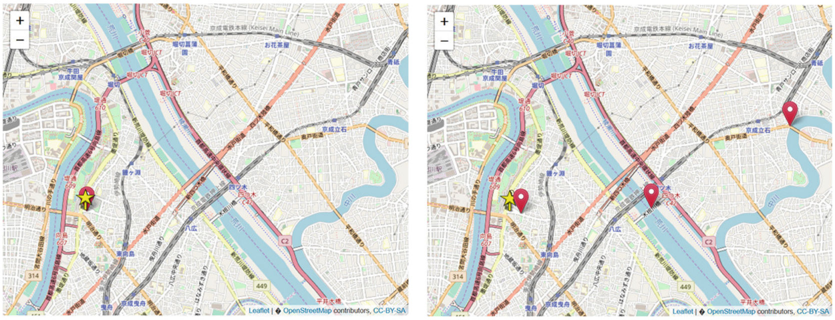
Figure 6 shows the results, where the horizontal and vertical axes denote the rank $k$up to top-100 and the precision $Prec_k^{(\tau)}$, respectively. Again, Figs. 6(a), (b) and (c) are the results for our three scenarios, respectively. From these results, we can see that there does not exist any similarity between TBC and EBC, especialy for disastaer evacuation and fire enging dispatch scenarios. In summary, the proposed TBC centrality can properly detect different critical links according to the task we consider.

Finally, we compare actual critical links detected based on the TBC centrality value with those based on the EBC in Fig. 7 for the scenario of ambulance call in the case of $\tau = 500$. In Fig. 7, the star shaped marker denotes the location of an emergency hospital. From Fig. 7(a) that shows the result for EBC, we can find that the nearest link from the hospital is detected as a critical link. This is natural because the number of people who pass the link is definitely larger than thoes who pass links further away when they intend to get to the hospital via the shortest pass. We observed the similar tendency for the other target nodes regardless of scenario considered. But, actually, that link is not really critical because an ambulance can easily get to the hospital within the given maximum permissible time by taking another route even if that link is blocked. On the other hand, TBC detected two bridges that are far away from the hospital as critical links as shown in Fig. 7(b) although a link near the hospital is also detected in this case. Note that such critical links, *i.e.*, bridges, detected in Fig. 7(b) can never be identified without considering a maximum permissible time.



(a) Evacuation facilities (b) Emergency hospitals (c) Fire stations

Fig. 6. Precision comparison in criticalness centrality of the top-100 links obtained by each centrality measure for different maximum permissible time ($\tau = 100$, 500 and 900).



(a) EBC (b) TBC

Fig. 7. Comparison of critical links detected by the EBC and TBC centralities in the case of ambulance call ($\tau = 500$), where the star shaped marker and the red pin marker denote the location of an emergency hospital and the detected critical link, respectively.

6 Conclusion

In this paper, we proposed an efficient algorithm to identify critical links in a spatial network based on node reachability assuming a realistic scenario in which a limitation is imposed on the traveling time from a node to the nearest target facility (or vice versa). Such time constraint is quite natural in an emergency situation such as disaster evacuation, ambulance call, and fire engine dispatch. To this end, we formalized the criticalness of a link as the reduction of the number of people that are reachable from/to one of target facilities within the maximum permissible time when the link is blocked, and named it the time-bounded criticalness centrality (TBC). The proposed algorithm can efficiently calculate the centrality by conducting tree search in two phases for a given spatial network. The first phase is forest construction (FC) that generates trees consisting of nodes reachable to one of target facilities in the spatial network within a given maximum permissible time. The second phase, local update (LU), actually calculates the time-bounded criticalness centrality of each link e by only checking a subtree isolated from the forest when e is removed. We experimentally evaluated the proposed algorithm using a real-world road network of Tokyo and the geographical information of actual evacuation facilities, emergency hospitals, and fire stations in Tokyo. The computation time of the proposed method is substantially small compared to the computation of betweenness centrality and less than 100 s even for a network having millions of nodes and links, in which the computation of FC that conducts the best first search is dominant and that of LU is much less implying the local search is quite efficient. The value of the time-bounded criticalness centrality of top-ranked critical links tends to increase as the maximum permissible time gets longer, but does not monotonically change. It depends on the number of target facilities and their locations. The critical links detected by the proposed centrality are quite different from the ones detected based on a straightforward extension of the traditional betweenness centrality and are distributed differently according to the scenario considered. Thus, the conventional centralities cannot serve as an alternative to the proposed time-bounded criticalness centrality.

Extending our algorithm so that it can handle road capacity is one future direction of this work.

Acknowledgments. This material is based upon work supported by JSPS Grant-in-Aid for Scientific Research (C) (No. 17K00314).

References

1. Akram, V.K., Dagdeviren, O.: Breadth-first search-based single-phase algorithms for bridge detection in wireless sensor networks. Sensors **13**(7), 8786–8813 (2013)
2. Barabási, A.L.: Network Science. Cambridge University Press, Cambridge (2016)
3. Brandes, U., Erlebach, T. (eds.): Network Analysis: Methodological Foundations. LNCS, vol. 3418. Springer, Heidelberg (2005). https://doi.org/10.1007/b106453
4. Burckhart, K., Martin, O.J.: An interpretation of the recent evolution of the city of barcelona through the traffic maps. J. Geogr. Inf. Syst. **4**(4), 298–311 (2012)
5. Crucitti, P., Latora, V., Porta, S.: Centrality measures in spatial networks of urban streets. Phys. Rev. E **73**(3), 036125 (2006)
6. Fang, Y.P., Pedroni, N., Zio, E.: Comparing network-centric and power flow models for the optimal allocation of link capacities in a cascade-resilient power transmission network. IEEE Syst. J. **99**, 1–12 (2014)
7. Grady, D., Thiemann, C., Brockmann, D.: Robust classification of salient links in complex networks. Nature Commun. **3**(864), 1–10 (2012)
8. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Trans. Knowl. Discov. Data **3**, 9:1–9:23 (2009)
9. Koschützki, D., Lehmann, K.A., Peeters, L., Richter, S., Tenfelde-Podehl, D., Zlotowski, O.: Centrality indices. In: Brandes, U., Erlebach, T. (eds.) Network Analysis. LNCS, vol. 3418, pp. 16–61. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31955-9_3
10. Montis, D.A., Barthelemy, M., Chessa, A., Vespignani, A.: The structure of interurban traffic: a weighted network analysis. Environ. Plan. **34**(5), 905–924 (2007)
11. Ohara, K., Saito, K., Kimura, M., Motoda, H.: Accelerating computation of distance based centrality measures for spatial networks. In: Calders, T., Ceci, M., Malerba, D. (eds.) DS 2016. LNCS (LNAI), vol. 9956, pp. 376–391. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46307-0_24

12. Ohara, K., Saito, K., Kimura, M., Motoda, H.: Maximizing network performance based on group centrality by creating most effective k-links. In: Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2017), pp. 561–570 (2017)
13. Oliveira, E.L., Portugal, L.S., Junior, W.P.: Determining critical links in a road network: vulnerability and congestion indicators. Procedia Soc. Behav. Sci. **162**, 158–167 (2014)
14. Opsahl, T., Agneessens, F., Skvoretz, J.: Node centrality in weighted networks: generalizing degree and shortest paths. Soc. Netw. **32**(3), 245–251 (2010)
15. Park, K., Yilmaz, A.: A social network analysis approach to analyze road networks. In: Proceedings of the ASPRS Annual Conference 2010 (2010)
16. Piraveenan, M., Prokopenko, M., Hossein, L.: Percolation centrality: quantifying graph-theoretic impact of nodes during percolation in networks. PLoS ONE **8**(1), 1–14 (2013)
17. Saito, K., Ohara, K., Kimura, M., Motoda, H.: Accurate and efficient detection of critical links in network to minimize information loss. J. Intell. Inf. Syst. **51**(2), 235–255 (2018)
18. Saito, K., Ohara, K., Kimura, M., Motoda, H.: Efficient detection of critical links to maintain performance of network with uncertain connectivity. In: Geng, X., Kang, B.-H. (eds.) PRICAI 2018. LNCS (LNAI), vol. 11012, pp. 282–295. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97304-3_22
19. Sariyüce, A.E., Kaya, K., Saule, E., Çatalyürek, U.V.: Graph manipulations for fast centrality computation. ACM Trans. Knowl. Discov. Data **11**(3), 26:1–26:25 (2017)
20. Shen, Y., Nguyen, N.P., Xuan, Y., Thai, M.T.: On the discovery of critical links and nodes for assessing network vulnerability. IEEE/ACM Trans. Networking **21**(3), 963–973 (2013)
21. Tarjan, R.E.: A note on finding the bridges of a graph. Inf. Process. Lett. **2**(6), 160–161 (1974)
22. Wang, P., Hunter, T., Bayen, A.M., Schechtner, K., Gonzalez, M.C.: Understanding road usage patterns in urban areas. Sci. Rep. **2**, 1001:1–1001:6 (2012)