A Ripple-Spreading Algorithm for Network Performance Assessment

Xiao-Bing Hu, Ming-Kong Zhang
Academy of Disaster Reduction and Emergence Management, Beijing Normal University
Beijing, China
dr_xiaobinghu@hotmail.co.uk

Jian-Qin Liao
Chengdu MiidShare Technology Ltd
Chengdu, China

Abstract*—To assess the performance of a network system against disturbances, existing methods are usually concerned with two different extreme situations: (i) how likely the system will degrade into some separated sub-graphs because of disturbances; (ii) how likely the 1st best paths will be cut off by disturbances. However, a more general situation, i.e., how likely those paths whose lengths are within a given range will be affected by disturbances, is barely discussed. Basically, to address this general situation, all (not just a proportion) of those paths whose lengths are within the given range must be found out between all pairs of origin and destination (OD) of interest. Unfortunately, no effective method has ever been reported to accomplish this task, although there are many methods capable of calculating all the 1st best paths between all OD pair of interest. This paper, for the first time, attempts to address the above general situation of network performance assessment. To this end, a novel ripple-spreading algorithm (RSA) is proposed to carry out a ripple relay race on the network, in order to identify all of those paths whose lengths are within the given range. Surprisingly, the proposed RSA can find all such paths between all OD pairs of interest by just a single run of ripple relay race. This work makes progress towards the general performance assessment of a network system against disturbances.

Keywords*—Ripple-spreading algorithm; network system; performance assessment; path optimization; disturbances.

I. Introduction

Inspired by the natural ripple-spreading phenomenon, we have recently reported quite a few ripple-spreading algorithms (RSAs) to some classical optimization problems [1]-[4]. Although the reported RSAs can resolve those classical optimization problems with second-to-none performance, there are usually full of other methods to well address the same classical optimization problems. In other words, the RSAs only play a role of "another interesting method but more or less not necessary" to those classical optimization problems in [1]-[4], because such problems have already been well resolved by other methods. Thus comes a fundamental question about the RSA: Is there any important problem existing methods can hardly resolve but the RSA can? Actually, we have shown that the RSA does exhibit certain unique value beyond other methods, e.g., the RSA can theoretically and practically guarantee the global optimality when dealing with networks with various time-delay or time-varying features [5]-[7]. This paper will shed a little more light on such unique value of the RSA beyond other methods.

Nowadays, network systems play an unprecedented important role in our daily life [8], and network performance assessment against disturbances (such as natural disasters, terrorist attacks, and component malfunction) has long been a hot topic in various research and engineering communities [9],[10]. The challenge of network performance assessment largely roots in the fact that the impact of a local disturbance event is usually not limited to the local area where the disturbance event occurs, but will easily spread out through the network due to the amplifying effect, cascading effect and/or ripple effect at nodes [11]-[13]. For example, the 2010 eruption of the Eyjafjallajokull Volcano in Iceland resulted in closures of European airports and routes at a very large scale, causing more than 10 million passengers delayed [14]. To address network related risk, there is an urgent demand for new theories, models and methods to assess the performance of a network system against disturbances [11]-[13].

As a fundamental feature of network systems, "connectivity" is always among those issues which receive the highest priority in the study of network performance assessment against disturbances. Two extreme situations related to connectivity are usually considered under disturbances. Situation 1, the network degrades into some separated sub-graphs because of disturbances; Situation 2, the 1st best paths between those pairs of origin and destination (OD) of interest are cut off by disturbances. "Degree of connectedness" (CND), indicating how many nodes are connected to a given node, is a key concept for addressing Situation 1. Based on CND, many theories, models and methods have been developed, which have promoted unprecedented advances in the study on network performance against disturbances in the last two decades [9],[10]. For instance, one of the most important findings in system science is that the CND distribution of most real-world complex networks significantly deviates from a Poisson distribution but has a power-law tail or a scale-free property, and a scale-free network is robust to random disturbances but vulnerable to intended attacks to hub nodes, i.e., attacking a few hub nodes can easily degrade a scale-free network into some separated sub-graphs, which means some OD pairs of interest might be isolated from each other. "Betweenness" is an important concept to study Situation 2, which indicates how many times a node/link appears as an intermediate node/link in all of those

This work was supported in part by the National Natural Science Foundation of China (Grant No.61472041), and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No.41321001).

1st shortest paths between all OD pairs of interest. Obviously, the failure of a node/link with a higher betweenness means that more OD pairs of interest cannot be connected in the most cost-efficient way (but they are usually still connected by the network).

Compared with Situation 1 and Situation 2, this paper is concerned with a more general situation: between all OD pairs of interest, all of those paths whose length/cost is within a given range are cut off by disturbances. So far, this general situation has barely been touched in the study of network performance against disturbances. One might argue that this general situation might be less important than those two well-studied situations. In this paper, we argue that this general situation is actually more important. Take the following scenario as an example. For a traveller who needs to go from O to D for a scheduled meeting, there are 3 questions: Question 1, how likely will the route network become apart so that there is no way to travel from O to D? Question 2, how likely will the 1st shortest, or the 1st most cost-efficient path between O and D become unavailable? Question 3, how likely will it turn out, due to disturbances to route network, there is no way to get to D in time, say, before 10:00am? Obviously, it is Question 3 that is often more concerned by normal people in our daily life. The above scenario and Question 3 can be extended to many areas of daily life, e.g., how likely is there no investment plan to achieve the minimal return ratio due to market uncertainties? Actually, Situation 1 and Situation 2 are just two special cases of the general situation. Fig.1 clearly illustrates the differences and relationships between the two extreme situations and the general one. Furthermore, we argue that the reason for why the general situation is barely studied is because it is very difficult, if not impossible, for existing methods, particularly those methods for Situation 1 and Situation 2, to find out all (not just some) of those paths whose length/cost is within a given range.

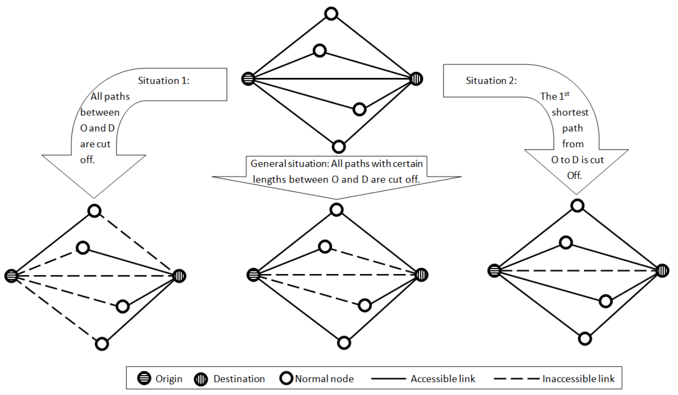
Situation 1:
All paths between O and D are cut off.
General situation: All paths with certain lengths between O and D are cut off.
Situation 2:
The 1st shortest path from O to D is cut Off.
Origin
Destination
Normal node
Accessible link
Inaccessible link

Fig.1 The general situation and two extreme situations in network performance assessment

To address the general situation in the study of network performance against disturbances, this paper will propose an effective ripple-spreading algorithm (RSA) to calculate, for all OD pairs of interest, all of those paths whose length/cost is within a given range (please note that different length ranges may be applied to different OD pairs of interest). Thanks to the decentralized agent-based nature of RSA, we can find out all of such paths between all OD pairs by just a single run of RSA. It should be pointed out that, in the study of Situation 2, a method for calculating the 1st best path, e.g., the A* or the Dijkstra's algorithm, usually has to be run repeatedly for many times, each time targeting one OD pair, in order to cover all OD pairs. Given a directed network with N_N nodes, there may be up to $N_N \times /(N_N-1)$ OD pairs of interest. This means existing methods for calculating the 1st best paths between all OD pairs has a scalability problem, which will become even worse when for every OD pair, all of those paths whose length is within a given range need to be identified (i.e., for each OD pair, it is not just 1 suitable path, but many suitable paths that need to be calculated). As will be demonstrated later, the RSA proposed in this paper, compared with existing methods, has a very good computational efficiency, because it can find out all suitable paths between all OD pairs in just a single run of algorithm, no need of any repetitions.

The remainder of this paper is organized as following. Section II gives the mathematical description of the problem under consideration. Section III explains how to design the RSA for network performance assessment. Section IV conducts some theoretical analyses. Section V shows some preliminary experimental results. The paper ends with some conclusions in Section VI.

II. Problem Description

Assuming a route network $G(V,E)$ is composed of node set V and link set E. V has N_N different nodes, and E has N_L links between nodes. This route network can be recorded as an $N_N \times N_N$ adjacency matrix A. The matrix entry $A(i,j)=1$, $i=1,\ldots,N_N$ and $j=1,\ldots,N_N$, defines a link from node i to node j. Otherwise, $A(i,j)=0$ means no link. We assume $A(i,i)=0$, i.e., no self-connecting link is allowed in this study. There is a cost, i.e., $C(i,j)$, associated with each link $A(i,j)$, and $C(i,j)$ will be used to calculate the length of a path between a pair of origin and destination (OD).

In many studies on path optimization problems (POPs), usually only one OD pair is given and concerned. In this study, we consider multiple OD pairs in a single POP, because in real world, a network system is established to serve not just a single pair of nodes, but usually many pairs simultaneously. Thus, we suppose there are N_{OD} OD pairs to be considered, and O_h and D_h are the origin and destination of the hth OD pair, $h=1,\ldots,N_{OD}$. Theoretically, for a directed network with N_N nodes, there could be up to $N_{OD}=N_N \times /(N_N-1)$ OD pairs of interest, i.e., every node in the network will be used as origin for (N_N-1) times, and as destination for (N_N-1) times. Of course, in some real world POPs, many nodes in a network are only used as intermediate nodes, and then we have $1 \leq N_{OD} < N_N \times /(N_N-1)$.

Between an OD pair, say, the hth OD pair, suppose a feasible/candidate path is recorded as an integer vector whose element $P(i)=j$ means node j is the ith node in the path, $i=1,\ldots,N_P$, and $j=1,\ldots,N_N$, where N_P tells how many nodes, including O_h and D_h, are included in the path. Obviously, $P(1)=O_h$ and $P(N_P)=D_h$. As a feasible/candidate path, P must satisfy

$$A(P(i),P(i+1))=1,\ i=1,\ldots,N_P-1, \quad (1)$$

i.e., there must exist a link between any two successive nodes in a path. Since no loop is allowed in this study, no node can appear in a path for more than once, i.e.,

$$P(i) \neq P(j),\ \text{if } i \neq j,\ i=1,\ldots,N_P, \text{ and } j=1,\ldots,N_P. \quad (2)$$

For a given path P, we can calculate its total cost, or length, from O_h to D_h as

$$f(P)=\sum_{i=1}^{N_P-1} C(P(i),P(i+1)). \tag{3}$$

In a POP which focuses on the 1st shortest path, the goal is the find a P that minimize $f(P)$. In a POP which is concerned with the k shortest paths, the goal is to find out all those paths which have the first k minimal values of $f(P)$. Differently, this paper aims to find out all of those paths whose length is within a given range, or to be more general, within some given ranges. Suppose there are $N_{GR,h}$ given ranges for the hth OD pair, $h=1,\ldots,N_{OD}$, let $[R_{GR}(h,i,1),R_{GR}(h,i,2)]$ denote the ith given range, $i=1,\ldots,N_{GR,h}$, and

$$R_{GR}(h,i,2)<R_{GR}(h,i+1,1). \tag{4}$$

To save space, hereafter, if a path connects an OD pair of interest and the length of the path is within one of such given ranges, i.e., if there exist a $1\le h\le N_{OD}$ and $1\le i\le N_{GR,h}$, such that

$$P(1)=O_h \text{ and } P(N_P)=D_h, \tag{5}$$

$$R_{GR}(h,i,1)\le f(P)\le R_{GR}(h,i,2), \tag{6}$$

then we call the path a *suitable* path. Please note the different meanings between a suitable path and a feasible/candidate path. The latter emphasizes on the fact that there physically exists a path connecting a pair of nodes (not necessary an OD pair), while the former demands more, i.e., a suitable path must be a feasible/candidate path first, and then it must connect an OD pair of interest and its length must be within a given range. Thus, the goal of this study is to find out all suitable paths between all OD pairs of interest, i.e., to find out all of those feasible/candidate paths which satisfy Conditions (5) and (6) for all $h=1,\ldots,N_{OD}$.

From Eq.(4) to Eq.(6), in a general sense, different OD pairs may have different given ranges, i.e., $N_{GR,h}$ may vary according to h, and so does $[R_{GR}(h,i,1),R_{GR}(h,i,2)]$, $i=,1\ldots,N_{GR,h}$.

If for every $1\le h\le N_{OD}$, $N_{GR,h}=1$ and

$$R_{GR}(h,1,1)=R_{GR}(h,1,2)=\infty, \tag{7}$$

then, the general situation becomes Situation 1, i.e., all paths between all OD pairs are concerned.

If for every $1\le h\le N_{OD}$, $N_{GR,h}=1$ and $R_{GR}(h,1,1)=-\infty$,

$$f(P^*_{h,1})\le R_{GR}(h,1,2)<f(P^*_{h,2}), \tag{8}$$

where $P^*_{h,1}$ and $P^*_{h,2}$ are the 1st and 2nd shortest paths from O_h to D_h, respectively, then, the general situation becomes Situation 2, i.e., only the 1st shortest paths between all OD pairs are concerned.

Now, what can we do with all suitable paths between all OD pairs for network performance assessment against disturbances? A straightforward way is to generalize the concept of "betweenness". The classical definition of betweenness is only concerned with the 1st shortest paths between all OD pairs: the betweenness of a node/link is defined as how many times the node/link appears as intermediate node/link in all the 1st shortest paths between all OD pairs. As is well known, the concept of betweenness plays a crucial role in network performance assessment and has been widely adopted in various real-world applications [9],[10]. Here we propose a *generalized betweenness* based on all suitable paths between all OD pairs: the generalized betweenness of a node/link is defined as how many times the node/link appears as intermediate node/link in all suitable paths between all OD pairs. Considering the scenario of "a meeting scheduled at 10:00am" in Section I again, the classical betweenness is of no help to answer the question how likely the traveller cannot get to the meeting in time, given the route network is exposed to certain disturbances. The generalized betweenness based on all suitable paths between all OD pairs makes it possible to answer Question 3, hopefully in a quantitative way. It is definitely worth further investigation regarding how to apply the generalized betweenness to address Question 3, and this paper will mainly focus on how to find out all suitable paths between all OD pairs, which apparently forms the foundation of the generalized betweenness.

A possible way to find out all suitable paths between all OD pairs is to borrow a method for the k shortest paths problem (k-SPP), i.e., for each OD pair, say, the hth OD pair, $h=1,\ldots,N_{OD}$, set up a k value, and then apply a k-SPP method to calculate the k shortest paths from O_h to D_h. If in such k shortest paths, there is at least one path whose length is larger than $R_{GR}(h,N_{GR,h},2)$, then all suitable paths for the hth OD pair can be determined according to such k shortest paths. Otherwise, we can increase k and re-run the k-SPP method until the condition of $R_{GR}(h,N_{GR,h},2)$ is triggered. This seems, theoretically, the general situation of network performance assessment can be addressed by any existing k-SPP method, such as those in [15]-[17]. However, if we go deeper, we can see there is a practical issue for k-SPP methods. For a single OD pair and a single k value, an existing k-SPP method usually needs an iteration and repetition process, i.e., to calculate the jth shortest path, $j>1$, many new networks need to be reconstructed based on the j-1 shortest paths, for each reconstructed network, a method for calculating the 1st shortest path needs to be repeated, and the shortest among all of those 1st shortest paths is then the jth shortest path in the original network. For a single OD pair, it is difficult to predict what k value will trigger condition of $R_{GR}(h,N_{GR,h},2)$. What one can do is to keep increasing and testing k. As the k value goes up, the computational burden of a k-SPP method will often soar up exponentially. Please note this is just for a single OD pair, and we may have up to $N_N\times/(N_N-1)$ OD pairs to handle.

Therefore, it is practicably very difficult, if not impossible for existing methods to find out all suitable paths between all OD pairs. In the next section, to address the general situation of network performance assessment, we will propose a novel ripple-spreading algorithm (RSA) with practicable computational efficiency and theoretical guarantee of optimality.

III. Design of RSA

A. The Basic Idea of RSA

As revealed in [1], the natural ripple spreading phenomenon reflects an optimization principle: a ripple spreads at the same speed in all directions, so it always reaches the closest spatial point first. This very simple principle can be easily applied to accomplish path optimization problems (POPs) effectively, and several ripple-spreading algorithms (RSAs) were developed in [1] to resolve one-to-one, one-to-all, many-to-many the 1st shortest path problems, and one-to-one the k shortest path problem. Most existing methods for path optimization problems are centralized, top-down, logic-based search algorithm. Differently, RSAs are actually decentralized, bottom-up, agent-based simulation model. By defining the behavior of individual nodes, optimality will automatically emerge as a result of the collective performance of the model. Simply speaking, RSA simulates a ripple relay race in route network. The race starts with an initial ripple at the origin; as the ripple spreads and reaches a node, a new ripple may be triggered at the node under certain condition; a ripple will become inactive when it has reached all the ends of links of its epicenter node; all active ripples keep spreading and more and more ripples are triggered at spatially farther away nodes; the relay race will terminate when certain criteria is met. As an agent-based model, RSA can easily adopt problem-specific micro agent behavior (i.e., how a node will generate a ripple) and macro termination criteria (i.e., when the ripple relay race should stop), so that it can resolve various POPs or any problem that can be converted into a POP [1].

B. The Design of RSA for Network Performance Assessment

As mentioned in the above sub-section, the key is to design problem-specific micro agent behavior at nodes and macro termination criteria for RSA. To calculate all suitable paths between all OD pairs by a single run of RSA, basically, we need to start an initial ripple at every origin node simultaneously, and each initial ripple is exclusively associated with an origin node. All initial ripples are active ripples. All active ripples will spread out along links at the same preset constant ripple spreading speed. New ripples may be triggered at nodes by incoming ripples. If an active ripple triggers a new ripple at a node, the new ripple will inherit the origin node from the incoming ripple. The so-far path of a ripple is defined as the path travelled by the ripple from its origin node to a node on the frontier of the ripple. If an active ripple arrives at a node which is not included in the so-far path of the ripple, then a new active ripple will be triggered at the node. Suppose the origin node of an active ripple is node n, and let L_{SFP} denote the length of the so-far path of this active ripple. Then, if

$$L_{SFP} \geq \max_{1\leq h\leq N_{OD}, O_h=n} R_{GR}(h, N_{GR,h}, 2), \quad (9)$$

this active ripple will become inactive, i.e., it will stop spreading out. An active ripple will also become inactive when it has reached all the ends of links of its epicenter node. If there is no active ripple any more, then the ripple relay race will be terminated, and all suitable paths will be identified by checking those ripples ever triggered at all destination nodes, i.e., D_h for h=1,…,N_{OD}.

The pseudocode of RSA for calculating all suitable paths between all OD pairs is described as following. Let N_R denote the number of ripples ever generated. Let $S_R(r)$ denote the state of ripple r, and $S_R(r)$=0/1 means ripple r is inactive/active. $E(r)$ denotes the epicenter of ripple r, i.e., ripple r is generated at node $E(r)$. $R(r)$ records the radius of ripple r. $T(r)$ records which ripple triggers ripple r. $O_R(r)$ records which origin node ripple r backtracks to. $L_{SFP}(r)$ records the length of the so-far path of ripple r.

Step 1: Initialize N_R=0. For the sake of both optimality and computational efficiency [1], set the ripple spreading speed as

$$v=\min(C(i,j)),\ i=1,\ldots,N_N,\ j=1,\ldots,N_N. \quad (10)$$

Step 2: For any node n, if there exists at least one $1\leq h\leq N_{OD}$ so that $O_h=n$, then set up an initial ripple at node n. Please note a node may have no more than 1 initial ripple. To set up an initial ripple at node n, first let $N_R=N_R+1$, then set $E(N_R)=n$, $R(N_R)=0$, $S_R(N_R)=1$, $T(N_R)=0$ (which means an initial ripple is self-triggered), $O_R(N_R)=n$, $L_{SFP}(N_R)=0$, and at last set simulation time t=0.

Step 3: If for every r=1,...,N_R, we have $S_R(r)$=0, i.e., if there is no active ripple any more, then go to Step 8; Otherwise, go to Step 4.

Step 4: Let $t=t+1$. For each active ripple r, i.e., if $S_R(r)$=1, $1\leq r\leq N_R$, update the radius of ripple r by v, i.e.,

$$R(r)=R(r)+v, \quad (11)$$

and update the length of the so-far path of ripple r by v as well

$$L_{SFP}(r)=L_{SFP}(r)+v. \quad (12)$$

Step 5: For any active ripple r, if it arrives at a node, say node n, i.e., if $S_R(r)$=1, $A(E(r),n)$=1 and $R(r)\geq C(E(r),n)$, and if node n is not included in the so-far path of ripple r, then, a new active ripple will be triggered at node n by ripple r: Let $N_R=N_R+1$; set $E(N_R)=n$, $T(N_R)=r$, $S_R(N_R)=1$, $R(N_R)=R(r)-C(E(r),n)$, $O_R(N_R)=O_R(r)$, and $L_{SFP}(N_R)=L_{SFP}(r)$.

Step 6: For any active ripple r, i.e., $S_R(r)$=1, if the length of its so-far path satisfies

$$L_{SFP}(r) \geq \max_{1\leq h\leq N_{OD}, O_h=E(r)} R_{GR}(h, N_{GR,h}, 2), \quad (13)$$

then set $S_R(r)$=0, i.e., ripple r becomes inactive.

Step 7: For any active ripple r, if for every node n that has $A(E(r),n)$=1, the following condition holds

$$R(r)\geq C(E(r),n), \quad (14)$$

then set $S_R(r)$=0. Go to Step 3.

Step 8: For each OD pair, i.e., for the hth OD pair, h=1,…,N_{OD}, check every ripple r which has $E(r)=D_h$ and $O_R(r)=O_h$. For such a ripple r, if the length of the path along which ripple r travels from O_h to D_h is within a given range for the hth OD pair, then the path is a suitable path from O_h to D_h. In this way, all suitable paths between all OD pairs can be found out by checking those ripples ever generated at all D_h, h=1,…,N_{OD}.

Fig.2 gives a simple example about how the proposed RSA can find out all suitable paths between all OD pairs of interest by just a single run of ripple relay race. In Fig.2, there are two OD pairs of interest: from O1 to D1 and from O2 to D2.Each OD pair has only 1 given range, $R_{GR}(1,1,1)=R_{GR}(2,1,1)=-\infty$, and $R_{GR}(1,1,2)<R_{GR}(2,1,2)$. At the beginning of the ripple relay race, two initial ripples, i.e, R1 and R2, are started at O1 and O2. At time t=2, R1 arrives at O2, and triggers a new ripple, i.e., R3 at O2. R3 has the same origin node, i.e., O1, as that of R1. Also at time t=2, R2 arrives at O1 and D1, triggering R4 at O1 and R5 at D1. R4 and R5 inherit the same origin node, i.e., O2, from R2. Since R2 has satisfied Condition (14) at time t=2, it becomes inactive and then is not plotted afterward. At time t=3, R1 arrives at D1, triggering R6, which identifies the 1st suitble path from O1 to D1.At the same time, R1 arrives at D2, triggering R7 at D2. Since R1 has satisfied Condition (14) at time t=3, it becomes inactive and then is not plotted afterward. At time t=4, R3 arrives at D1, triggering R8, which identifies the 2nd suitable path from O1 to D1. At the same time, R5 arrives at D2, triggering R9, which identifies the first suitable path from O2 to D2. At time t=5, all ripples whose origin node is O1, i.e., R3, R6, R7 and R8, have a $L_{SFP}>R_{GR}(1,1,2)$, i.e., satisfying Condition (13), so, they become inactive and then are not plotted. At time t=5, R5 arrives at O1, triggering R10 at O1. R9 identifies the 1st suitable path from O2 to D2. At the same time, R4 arrives at D2 and D1, triggering R11 at D2 and R12 at D1. R11 identifies the 2nd suitable path from O2 to D2. At time t=6, all ripples whose origin node is O2, i.e., R4, R5, and R9 to R12, have a $L_{SFP}>R_{GR}(2,1,2)$, i.e., satisfying Condition (13), so, they become inactive. By time t=6, there is no active ripple, so, the ripple relay race is terminated, and then all suitable paths between all OD pairs have been found out successfully.

IV. Theoretial Analyses

Regarding the optimality of the proposed RSA, it is guaranteed by the ripple spreading optimization principle revealed in [1]. Basically, the proposed RSA can be viewed as a natural integration of many one-to-one RSAs, which are running in a parallel manner. A detailed mathematical proof about the optimality of one-to-one RSA was given in [1], and such a proof can be easily extended to the RSA proposed in this paper.

Regarding the complexity of the proposed RSA, for a comparative purpose, first we analyze what could be the computational complexity of extending a k-SPP method to calculate all suitable paths between all OD pairs. As discussed in Section II, we need to apply a k-SPP method for each OD pair separately. Suppose, on average, for each OD pair, finding the k shortest paths is enough to identify all suitable paths between the OD pair. Thus, according to [18], an efficient k-SPP method may have a computational complexity of $O(k\times N_N+N_L\times\log(N_L))$ to handle a single OD pair. We may have up to $N_N\times(N_N\text{-}1)$ OD pairs in a single problem of network performance assessment. This means, when applying a k-SPP method to network performance assessment, the computational complexity will be $O(k\times N_N^3+N_N^2\times N_L\times\log(N_L))$.

Now we analyze the computational complexity of the proposed RSA for network performance assessment. Basically, for each origin node, there is an initial ripple. The ripple relay race started by this single initial ripple can identify all suitable paths from the origin node to all specified destination nodes, because each ripple will spread out in all directions to all relevant nodes simultaneously. The above analysis on the k-SPP method assumes that finding the k shortest paths is enough. The explanation of this assumption in the context of RSA is that each node needs to generate up to k ripples, in order to find out all suitable paths from the origin node to all specified destination nodes.

According to Section III, the basic computational step in the RSA is the spreading of a ripple along a link in a time unit. The basic computational step mainly includes a few addition operations and comparison operations, e.g., increasing the radius of a ripple by the ripple-spreading speed, and then comparing the new radius with the length of a link. Suppose a network has N_N nodes, N_L links (thus, each node has N_L/N_N links on average), and it takes N_{ATU} time units on average for a ripple to travel through a link.

For identifying all suitable paths from a single origin node to all specified destination nodes, no more than k ripples are needed at a node. Therefore, basically, in the worst case, since the source node only generates one ripple, it then needs $N_{ATU}\times N_L/N_N$ basic computational steps on average. For the rest (N_N–1) nodes, each needs up to $k\times(N_L/N_N\text{-}1)\times N_{ATU}$ steps on average. Thus, about N_{BCS} basic computational steps need to be conducted before all suitable paths from a single origin node to all specified destination nodes can be found, where

$$\begin{aligned} N_{BCS} &= N_{ATU}\times N_L/N_N+(N_N-1)\times k\times(N_L/N_N-1)\times N_{ATU}. \\ &< N_{ATU}\times N_L/N_N+k\times N_{ATU}\times N_L \end{aligned} \quad (15)$$

For a large scale network, we have $N_L>>N_L/N_N$. This means the computational complexity for calculating all suitable paths from a single origin node to all specified destination nodes can be assessed as $O(k\times N_{ATU}\times N_L)$.

The RSA reported in Section III can be viewed as a combination of many single-origin-node-based ripple relay races, because such single-origin-node-based ripple relay races run in a parallel and independent manner. In network performance assessment, we may have up to N_N origin nodes, i.e., every node in the network is origin node, so the proposed RSA is a combination of N_N single-origin-node-based ripple relay races. Therefore, the computational complexity for calculating all suitable paths between all OD pairs can be assessed as $O(N_N\times k\times N_{ATU}\times N_L)$.

When comparing with the computational complexity of a k-SPP method, i.e., $O(k\times N_N^3+N_N^2\times N_L\times\log(N_L))$, even if we could ignore the first part of $O(k\times N_N^3)$, the second part of $O(N_N^2\times N_L\times\log(N_L))$ is usually still larger than the complextiy of RSA, i.e., $O(N_N\times k\times N_{ATU}\times N_L)$, given $N_N\times\log(N_L)>k\times N_{ATU}$, which is often the case when the network scale is large.

As discussed in [1], a most challenging and complicated step in a classical k-SPP method is to keep reconstructing route networks in an iteration process, as illustrated in Fig.3. First of all, a classical k-SPP method applies, say, the Dijkstra's algorithm [19],[20], to find the 1st shortest path $1\rightarrow2\rightarrow3\rightarrow4$ in Fig.3.(a). Then the classical method proceeds to calculate the 2nd shortest path based on the 1st shortest path. To this end, the classical method has to reconstruct three new route networks (see Fig.3.(b) to Fig.3.(d)), each of which is generated by removing one of the three links that together compose the 1st shortest path. Then based on each reconstructed route network, the classical method applies the Dijkstra's algorithm to calculate the associated 1st shortest path. The associated 1st shortest paths in Fig.3.(b) to Fig.3.(d) are $1\rightarrow3\rightarrow4$, $1\rightarrow3\rightarrow4$, and $1\rightarrow2\rightarrow4$, respectively. Among these associated 1st shortest paths, $1\rightarrow3\rightarrow4$ is the shortest. Therefore, the classical method finds out that the 2nd shortest path in the original route network of Fig.3.(a) is $1\rightarrow3\rightarrow4$. Existing methods following this practice often focus on how to reconstruct route networks efficiently [15]-[18].

The proposed RSA never needs to reconstruct any network, but just runs a single multiple-origin-nodes-based ripple relay race on the very original route network, e.g., see Fig.2. This straightforward design based on the ripple spreading optimization principle of [1] significantly contributes to the computational efficiency of RSA.

V. Preliminary Experimental Results

As the general situation of network performance assessment against disturbances is bared discussed in literature, in particular, the generalized betweenness proposed in Section II is completely new to the authors' best knowledge, there is no reported benchmark test case which could be borrowed to test the proposed RSA for calculating all suitable paths between all OD pairs of interest. Therefore, the goal of this section is not to conduct a comprehensive experimental study on the RSA, but to develop some preliminary test cases for future extension and research.

In most studies on POPs, including our previous study on the RSA for classical POPs [1], route networks used in experiments are usually randomly generated. Basically, it is very difficult, if not impossible, to conduct theoretical analyses on a randomly generated network, e.g., for a random network, we cannot deduce out the 1st shortest path by analysis, but only run an algorithm to calculate the 1st shortest path. Is the result calculated by the algorithm for a random network correct? Basically, if the algorithm has a theoretical proof about its optimality, then the calculated result should be correct. What if there is no theoretical proof, or a theoretical proof is under doubt or argument? For a classical POP, there are usually many different algorithms already developed. Therefore, we can compare with relevant algorithms, in order to still get some conclusions on a random network.

Unfortunately, in this preliminary study on the general situation of network performance assessment against disturbances, it is not suitable to test on randomly generated networks, because there is no other algorithm we can compare with. In a test without other algorithms for comparison, if we cannot deduce out the theoretical optimal result of a test network, then the experimental results would not be convincing enough, although we have a general theoretical proof about the optimality of RSA.

Therefore, any test case to be developed in this preliminary study should be based on some networks whose optimal results can be theoretically deduced out rather than being found out by an algorithm.

Regarding network performance assessment against disturbances, scale-free topologies play an important role. Therefore, this preliminary study should develop some test cases which can reveal how different levels of scale-free feature will affect the performance of network against disturbances.

Based on the above concerns, in this preliminary study, we develop two extreme categories of regular networks, as illustrated in Fig.4. One category is regular scale-free network, where (N_N-1) spoke nodes (i.e., nodes with low CND) are evenly distributed on a circle, and a hub node is located in the circle center connecting to every spoke node. In a network of this category, (N_N-1) spoke nodes have the same very small CND, and only the hub node has a very large CND. The other category is grid network on a sphere, where N_N nodes are evenly distributed on a sphere, and each node is connected to those spatially closest nodes on the sphere. In a network of this category, all of the N_N nodes have the same CND, i.e., there is no scale-free feature at all. Actually, in a network of this category, all of the N_N nodes are exactly the same in terms of any network property, including the generalized betweenness proposed in this paper. It is practicable to conduct theoretical analyses on performance assessment for these two categories of regular networks. With such results of theoretical analyses, we can then fairly judge the results calculated by the proposed RSA. For example, for a given path length range, we can theoretically deduce out how many suitable paths there are between all OD pairs in the two networks of Fig.4. Then, we apply the RSA to the networks in Fig.4, and the calculated results are exactly the same as those theoretical results. With these two categories of regular networks, we will carry out more experiments in future research, in order to gain a deeper and more comprehensive understanding of the proposed RSA and the generalized betweenness for the sake of network performance assessment.

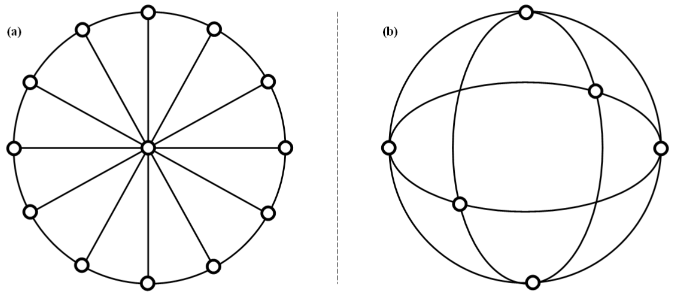

Fig.4 (a) A regular scale-free network; (b) A grid network on a sphere.

VI. Conclusions and Future Work

This paper proposes a novel ripple-spreading algorithm (RSA) to assess network performance against disturbances in a more general situation: how likely between all pairs of origin and destinations (OD) of interest, will all of those paths whose length is within a given range be cut off by disturbances? Existing studies, such as how likely a network system will degrade into some separated sub-graphs, or how likely all of those 1st best paths between all OD pairs will be cut off, are just special cases of the general situation concerned in this paper. The proposed RSA can address the general situation of network performance assessment in a highly efficient way, i.e., between all OD pairs of interest, the RSA can find out all of those paths whose length is within a given range by a single run of algorithm. Theoretical analyses and experimental results clearly show that the proposed RSA exhibits significant advantages when compared with existing methods for network performance assessment against disturbances.

As a preliminary study, more efforts are needed in future to further imporve the reported work. For example, conduct a more comprehensive analysis on the complexity of the propsoed RSA, compare with some state-of-the-art algorithms for the *k*-SPP, design a decentralized parallel version of the RSA in order to achieve better computatioal effciency, and test on various relevant real-world applications.

References

[1] X.B. Hu, M. Wang, M.S. Leeson, E. Di Paolo, and H. Liu, "Deterministic agent-based path optimization by mimicking the spreading of ripples," *Evolutionary Computation*, vol.24, no.2, 2016, doi:10.1162/EVCO_a_00156.

[2] X.B. Hu, M. Wang, Q. Ye, Z.-G. Han, and M.S. Leeson, "Multi-Objective New Product Development by Complete Pareto Front and Ripple-Spreading Algorithm," *Neurocomputing*, vol.142, pp. 4-15, 2014.

[3] X.B. Hu, E. Di Paolo, "A Ripple-Spreading Genetic Algorithm for the Aircraft Sequencing Problem," *Evolutionary Computation*, Vol. 19, No. 1, pp. 77–106, 2011.

[4] X.B. Hu, M.S. Leeson and E.L. Hines, "A Ripple-Spreading Genetic Algorithm for the Network Coding Problem", *The 2010 IEEE World Congress on Computer Intelligence (WCCI2010)*, Congress on Evolutionary Computation (CEC2010), 2010, Spain.

[5] X.B. Hu, Q. Sun, M. Wang, M.S. Leeson, and E. Di Paolo, "A Ripple-Spreading Algorithm to Calculate the *k* Best Solutions to the Project Time Management Problem," *2013 IEEE Symposium Series on Computational Intelligence* **(***IEEE SSCI 2013)*, 16-19 April 2013, Singapore.

[6] X.B. Hu, M.K. Zhang, J.Q. Liao, and H.L. Zhang, "An Effective Method to Find the k Shortest Paths in a Generalized Time-Window Network", *The 2016 IEEE World Congress on Computer Intelligence (WCCI2016), Congress on Evolutionary Computation (CEC2016)*, 24-29 July 2016, Vancouver, Canada.

[7] X.B. Hu, and J.Q. Liao, "Co-Evolutionary Path Optimization by Ripple-Spreading Algorithm", *The 2016 IEEE World Congress on Computer Intelligence (WCCI2016), Congress on Evolutionary Computation (CEC2016)*, 24-29 July 2016, Vancouver, Canada

[8] P. Ball, *Why Society is a Complex Matter*, Springer, 2012.

[9] S. Boccaletti, V. Latora, Y. Moreno, M. Chaves, and D.U. Hwang, "Complex networks: Structure and dynamics", *Phys. Rep.*, **424**, 175 (2006).

[10] R. Albert, and A.L. Barabasi, "Statistical mechanics of complex networks", *Reviews of Modern Physics*, **74**, 47 (2002).

[11] Future Global Shocks-Improving Risk Governance, OECD Reviews of Risk Management Policies, 2011.

[12] D. Helbing, "Globally networked risks and how to respond", Nature, doi:10.1038/nature12047, 2013.

[13] X.B. Hu, A.V. Gheorghe, M.S. Leeson, S.P. Leng and J. Bourgeois, "Risk and Safety of Complex Network Systems," *Mathematical Problems in Engineering*, vol.2016, Article ID 8983915, 2016.

[14] M. Mazzocchi, F. Hansstein, and M. Ragona, "The 2010 volcanic ash cloud and its financial impact on the European airline industry," *CESifo Forum*, 92–100, 2010.

[15] J.Y. Yen, "Finding the k shortest paths in a network," *Management Science*, vol. 17, no. 11, pp. 712-716, 1971.

[16] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 652-673, 1998.

[17] H. Aljazzar, and S. Leue, "K: A heuristic search algorithm for finding the *k* shortest paths", *Artificial Intelligence*, vol. 175, no. 18, pp. 2129-2154, 2011.

[18] K. Mohanta, "Comprehensive Study on Computational Methods for K-Shortest Paths Problem," *International Journal of Computer Applications*, vol.40, no.14, pp. 22-26, 2012.

[19] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms* (2nd ed.). MIT Press and McGraw-Hill, 2001.

[20] M. Sniedovich, "Dijkstra's algorithm revisited: the dynamic programming connexion," *Journal of Control and Cybernetics*, vol.35, no.7, pp. 599–620, 2006.

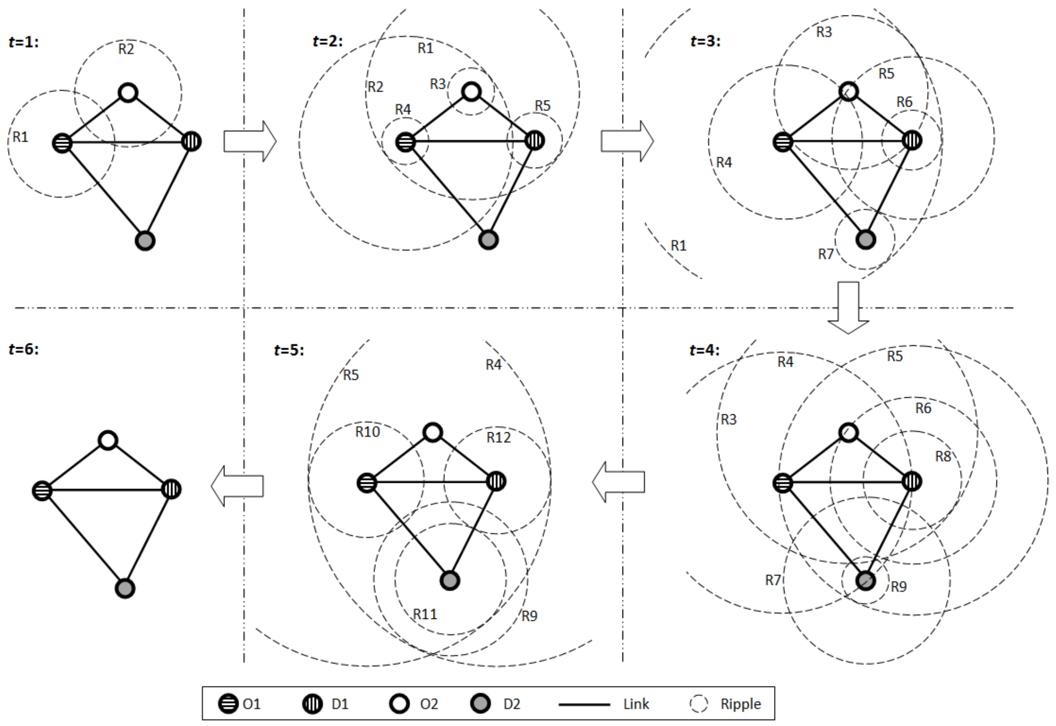

Fig.2. An example about how the proposed RSA can find out all suitable paths between all OD pairs of interest by just a single run of ripple relay race.

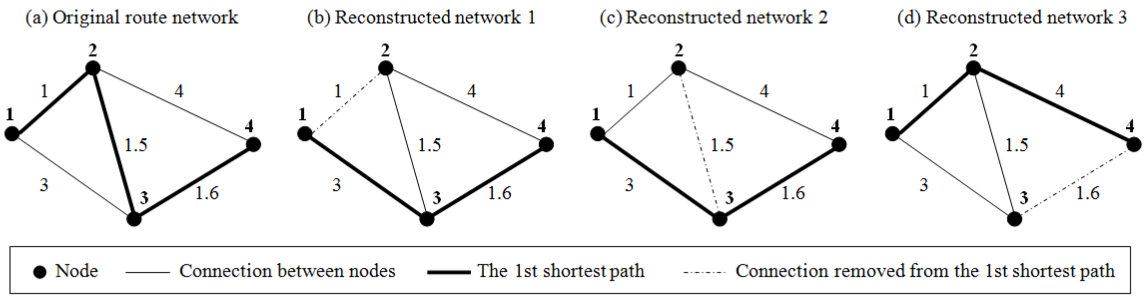
(a) Original route network
(b) Reconstructed network 1
(c) Reconstructed network 2
(d) Reconstructed network 3
1
2
3
4
1
4
1.5
3
1.6
Node
Connection between nodes
The 1st shortest path
Connection removed from the 1st shortest path

Fig.3. Reconstructing route networks in classical methods for the *k*-SPP