Optimal determination of source-destination connectivity in random graphs

Luoyi Fu, Xinbing Wang
Dept. of Electronic Engineering
Shanghai Jiao Tong University
Shanghai, China
{yiluofu,xwang8}@sjtu.edu.cn

P. R. Kumar
Dept. of Electrical and Computer Engineering
Texas A&M University
College Station, Texas
prk@tamu.edu

ABSTRACT

This paper investigates the problem of optimally determining source-destination connectivity in random graphs. We consider the classic Erdos-Renyi (ER) random graph with n nodes, where an edge independently exists between any two nodes with probability p. The problem examined is that of determining whether a given pair of nodes, a source S and a destination D, are connected by a path. Assuming that at each step one edge can be tested to see if it exists or not, we determine an optimal policy that minimizes the total expected number of steps.

The optimal policy has several interesting features. In order to establish connectivity of S and D, a policy needs to check all edges on some path to see if they all exist, but to establish disconnectivity it has to check all edges on some cut to see if none of them exists. The optimal policy has the following form. At each step it examines the condensation multigraph formed by contracting each known connected component to a single node, and then checks an edge that is simultaneously on a shortest S-D path as well as in a minimum S-D cut. Among such edges, it chooses that which leads to the most opportunities for connection. Interestingly, the optimal strategy does not depend on p or n, even though the entire graph itself undergoes a sharp transition from disconnectivity to connectivity around $p = \ln n/n$. The policy is efficiently implementable, requiring no more than $30\log_2 n$ operations to determine which edge to test next.

The result also extends to some more general graphs.

1. INTRODUCTION

Connectivity of random graphs has long been a topic of intensive study. A widely studied model, known as the $G(n,p)$ model, was proposed by Gilbert [1] in 1959. It refers to a graph with n nodes, with an edge existing independently between any pair of nodes with probability p. Erdos and Renyi [2, 3] showed that this graph is connected with probability approaching one as $n \to \infty$, if $p(n)$, the probability p as a function of n, satisfies $p(n) > \frac{(1+\epsilon)\ln n}{n}$, and contains isolated nodes with probability approaching one as $n \to \infty$, if $p(n) < \frac{(1-\epsilon)\ln n}{n}$. We will refer to the $G(n,p)$ model as the ER graph, in conformity with common usage. In a wireless networking context, Gilbert [10], and, more recently, Penrose [11] and Gupta and Kumar [12], have studied geometric random graphs where nodes have random uniform i.i.d. locations in a unit disk, and showed that if the radio transmission range of each node is $r(n) = \sqrt{\frac{\ln n + c(n)}{n\pi^2}}$, then the whole network is connected with probability approaching one as $n \to \infty$, if and only if $c(n) \to +\infty$. Ever since, there has been much research effort directed towards studying asymptotic connectivity of randomly distributed wireless networks [13]- [21].

In this paper we consider ER Graphs. The above works all focus on the connectivity of the entire network in the asymptotic regime where the number of nodes goes to infinity. The focus here is different in the following three ways: (i) It is non-asymptotic, (ii) it is specific to the specific realization of the random network that is being studied, and, (iii) instead of studying the whole network's connectivity, the issue is the connectivity between a specified source S and destination D. We examine the fundamental problem of designing a policy that determines the connectivity of S and D in minimum expected number of steps, where at each step one edge can be chosen and tested to see if it exists. The algorithm we present concludes either with the discovery of a route or the discovery of a cut between S and D. It is designed to reach a conclusion regarding one or the other in the shortest expected time in ER graphs. It should be noted that quickest discovery of a route is very different from discovering the shortest path route, which is a very well understood problem.

We start by considering the classic ER random graph, with a designated source S and destination D. The independence of edge occurrences and their equal probabilities turn out to give ER graphs a great advantage in terms of solvability over other more complicated graph structures. Subsequently, we consider some slightly more general graphs in Section 5, where there is some additional deterministic structure known a priori about the presence or absence of certain edges.

The question of whether S and D are connected can be resolved if one can either display an S-D path, or an S-D cut. However, we do not know a priori whether the graph is connected or not, thereby making it difficult to know what to do – try to find a path or try to find a cut. The optimal policy has to be dynamic, based on the presence or absence of previously tested edges. This inherent tension in the problem of determining connectivity makes the problem somewhat challenging. Another difficulty is that we do not know the optimal solution for general graphs where some edges are known to exist, some known not to exist, and others existing i.i.d. with probability p, which is what one generally has after some steps of testing. Thus the proof of optimality is not based on dynamic programming-like arguments. Yet another aspect of interest is that the random graph exhibits a phase transition depending on the value of p. For $p > \frac{(1+\epsilon)\ln n}{n}$, the entire graph itself, and not just the particular S-D pair, is connected with high probability for large n, while for $p < \frac{(1-\epsilon)\ln n}{n}$, it is disconnected with high probability. Thus, one may possibly expect that the optimal strategy will depend on the value of p. Very interestingly, however, the optimal policy does not depend on p or n at all!

The main result is the the optimality of the following simple testing strategy: At each stage, form the condensation multigraph by contracting each known connected component to a single node. It is a multigraph since the edges between nodes are inherited by the components to which they are contracted. In the set of edges that lie both on a shortest S-D path as well on the minimal S-D cut, test an edge that leads to the most opportunities for connectivity. The policy also has minimal complexity; it requires no more than $30\log_2 n$ operations to determine which edge to test next.

The policy is illustrated for a four-node ER graph in Figure 1. Figure 2 shows the number of steps, averaged over 100 simulations, that the optimal policy takes to establish S-D disconnectivity/connectivity in an ER graph with 1000 nodes, as a function of p. Determining S-D connectivity at around its phase transition can involve many steps. In between disconnectivity at very low p (say 10^{-5}) that takes about 999 steps to establish, and connectivity at very high p (around $p = 1$) that can be established in 1 step, the value of p passes through a phase transition around $p = 0.002$ (slightly smaller than the value of $p = \ln n/n \approx 0.006$ under the phase transition of connectivity of the entire network), where it takes a very large number of steps (about 15000) to determine if S and D are connected or not.

The proof of optimality of the policy follows from the proofs of optimality of the following three rules, which, when combined together give rise to the policy equivalently described above in terms of the condensation multigraph: Rule 1) The testing starts with a tactic that can lead to early termination by finding an S-D path: it first tests the edges, i.e., one-hop paths, connecting the component containing S and the component containing D. Rule 2) If there are no such edges, then it switches to checking for S-D disconnectivity by testing the edges on the S-D cut that contains the minimum number of untested edges. Rule 3) Among the edges in that cut, test the edge that leads to most opportunities for connection. These three rules optimally resolve the tension between checking for paths and cuts.

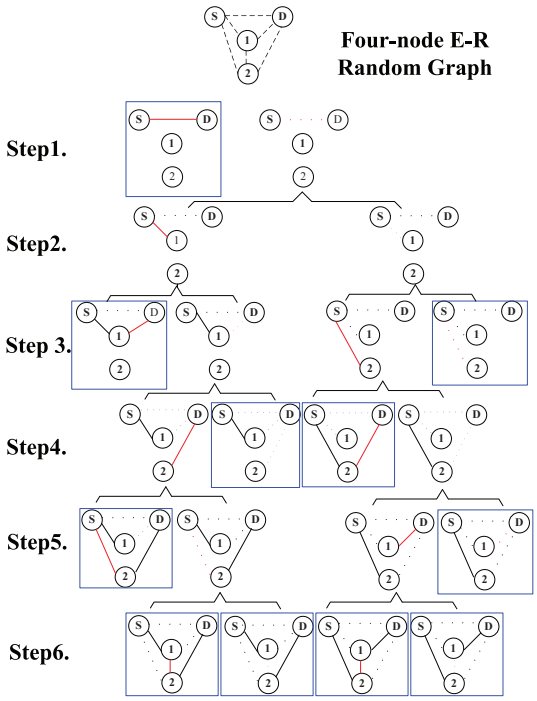

Figure 1: Illustration of the evolution of the optimal policy on a four-node ER random graph. The line in red shows the edge selected for testing at each step. A full line indicates that the edge was found to exist after testing, while a dotted line indicates that the edge was found to not exist after testing.

Related Works: We note that there is no other work, to the best of our knowledge, that addresses connectivity between a designated source and destination in random graphs. Rather, as noted earlier, it is asymptotic connectivity of the whole graph that has been intensively studied. Inspired by Erdos and Renyi [2, 3], there have been many works [4]- [9] dedicated to investigating connectivity of the entire network when the number of nodes is sufficiently large.

Our work also has ties to wireless networks where connectivity is determined by distance. Gilbert [10] initiated the study of random graphs when nodes are randomly distributed on a plane. Penrose [11] and Gupta and Kumar [12] have determined the critical range for establishing overall connectivity with probability ap-

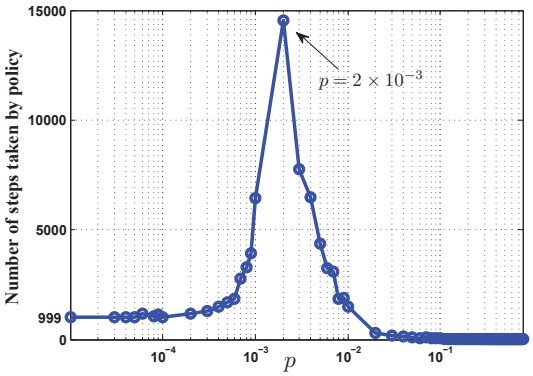

Figure 2: Number of steps needed to determine S-D disconnectivity/connectivity in a 1000 node ER Graph, as a function of p.

proaching one as n goes to infinity. This range results in each node connecting to $\log n$ neighbors on average. Xue and Kumar [14], and, subsequently, [15] and [16], further considered the number of nearest neighbors that nodes need to connect to, in order to ensure the network connectivity. The condition of full connectivity was relaxed by Dousse et. al. [17], which analyzed the scenario where the sink is connected (in a multihop fashion) to a set of nodes that span the entire network. The impact of node degree, density and network dimension on connectivity is also investigated [13], [18]. Due to the emergence of a,d hoc networks, mesh networks and sensor networks, there has also been research interest in analysis of k-connectivity [19], [20], [21].

2. MODEL AND DEFINITIONS

Consider an ER Random Graph with n nodes. An edge exists between any pair of nodes with equal probability p, independently of others. At each step, one can test a potential (i.e., untested) edge of the graph to see if it indeed exists. Two specific nodes are identified, labeled S and D. Our objective is to construct a sequential testing strategy terminating in the minimal expected number of steps, to determine if S and D are connected or not. Termination occurs with

- Connectivity, when it has verified the existence of all edges on some S-D path, or
- Disconnectivity, when it has verified the absence of all edges on some S-D cut.

We employ three terms, i.e., *known edge*, *known non-edge* and *potential edge* (also sometimes called an "untested edge."). A known edge is one that has already been tested and found to exist. Similarly, a known non-edge is one that has already been tested but found to not exist. Finally, a potential edge is an edge that is yet to be tested, which upon testing may turn out to exist or not exist. In Table 1, we list other definitions used.

Table 1: Notations

Notation	Definition
e_t	The edge tested at step t.
$\mathcal{G}_t$	The graph state known at step t. It is a list of (i) those potential edges that have been tested and found to exist, i.e., "known edges," (ii) those potential edges that were tested and found to not exist, called "non-edges," and (iii) the remaining potential edges that have not yet been tested. Abbreviated to $\mathcal{G}$ if there is no confusion (similarly for the notation below).
$C_{S,t}$ (or $C_{D,t}$, or $C_{i,t}$)	The connected component containing the source (or destination, or the i-th component not containing the source or destination), in the graph comprised of the known edges at step t.
$\mathscr{M}_{\mathcal{G}}$	A *minimum cut* for a graph state $\mathcal{G}$, defined as a set which contains the minimum number of potential edges at time t, which if they were found to not exist, would lead to the conclusion that C_S and C_D are not connected by a path. Note that a minimum cut may be non-unique.

3. POLICY FOR CHECKING S-D CONNECTIVITY IN ER RANDOM GRAPHS

We now present a policy $\pi^* = \{\pi_0^*, \pi_1^*, \ldots, \pi_N^*\}$, which we call the alternating policy, for checking S-D connectivity in an ER random graph. We label the nodes other than the designated S and D as $1, 2, \ldots, n-2$. Each π_t^* maps $\mathcal{G}_t$ (defined in Table 1) to one of the remaining potential edges. It specifies which potential edge should be tested at time t. Note that $N = n(n-1)/2$ is the upper bound on the number of edges in the graph, and thus the step by which the policy must necessarily have terminated.

The policy π^* has been defined in a brief manner in terms of the condensation multigraph in Section 1. Now we define it in a detailed manner in terms of three rules more be suitable for illustration and proof.

3.1 The Alternating Policy π^*

Rule 1: The policy tests a potential edge between C_S and C_D as long as there exists one such direct, one-hop, potential edge between them. Clearly, if such a direct edge is found to exist, then the policy terminates with the finding that S and D are connected.

Rule 2: If there are no potential edges connecting C_S and C_D as specified in Rule 1, then choose an edge from the list $\mathcal{L}$ defined as follows.

First list all the paths, comprised of known or potential edges, connecting C_S to C_D with the minimum number of potential edges. One such path is illustrated in Figure 3(a). Such a path must necessarily have potential edges only *between* connected components. If the path traverses only one component, say C_1, on its

path from C_S to C_D, then it must have exactly two potential edges on it, one connecting C_S to C_1, and another connecting C_1 to C_D, as shown in Figure 3(a). The remaining portion of the path must only consist of known edges *within* connected components C_S, C_1 or C_D. The same rule holds on the remaining potential *shortest* paths, each traversing through only one component, say C_i $(2 \leq i \leq r)$, on its path from C_S to C_D.

Fix a minimum cut $\mathscr{M}_\mathcal{G}$ for the graph state $\mathcal{G}$ at that time. Figure 3(a) shows two cuts, one separating C_S from $C_1 \cup C_2 \cup \ldots C_r$, and another separating C_D from $C_1 \cup C_2 \cup \ldots C_r$. (Suppose that, as in the case illustrated, the former is a Minimum Cut, and the latter contains more edges. In the sequel we will show that shortest paths are of length at most two, and minimum cuts will be precisely of this form, as long as the policy keeps testing the edges in the prescribed way.)

Among all the potential edges that are on the aforementioned shortest paths, list only those potential edges that are also in the chosen minimum set $\mathscr{M}_\mathcal{G}$. Define this as the list $\mathcal{L}$. Such a list is shown in Figure 3(b).

For convenience of definition of Rule 3 below, we also list the set of components $\{C_1, C_2, \ldots, C_r\}$ $(0 \leq r \leq n-2)$ to which the edges belonging to this list $\mathcal{L}$ are connecting. Let us call this set of components as $\mathcal{C}$, i.e., $\mathcal{C} := \{C_1, C_2, ..., C_r\}$. Figure 3(b) illustrates this set $\mathcal{C}$ of components too.

Rule 3: In Rule 3 we further sharpen Rule 2 by specifying which particular edge in $\mathcal{L}$ should be tested.

Rule 2 when followed will lead to the following structural property, as we prove in the sequel. All edges in $\mathcal{L}$ are either between C_S and $U_{i=1}^r C_i$, or all edges are between C_D and $U_{i=1}^r C_i$. (Certainly this is true at the outset for the ER graph, and if an edge is chosen from $\mathcal{L}$, as in Rule 2, then it retains that property). Suppose all edges in $\mathcal{L}$ connect to C_S, i.e., have one endpoint in C_S. Then, for each C_i, let n_i be the number of direct potential edges between C_i and C_D. Sort the components in $\mathcal{C}$ based on n_i in decreasing order.

Suppose that n_1 is the largest. Test any edge in $\mathcal{L}$ that connects C_S to C_1. This is illustrated in Figure 3(c).

4. THE PROOF OF OPTIMALITY OF THE ALTERNATING POLICY π^*

We now commence the proof of optimality of the Alternating Policy π^*. It uses a carefully chosen stochastic coupling in several places. We will consider the graph shown in Figure 3(c); the general case proceeds similarly. The edges meeting Rules 2 and 3 are the potential edges between C_S and C_1. Note that compared to the other C_i's, C_1 has the greatest chance of subsequently having an edge connected to C_D, if an edge is found between C_1 and C_S.

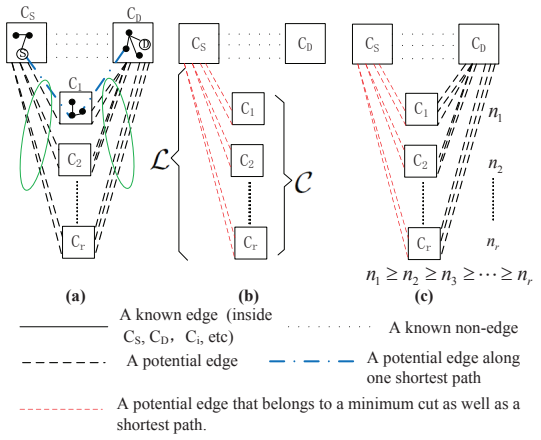

Figure 3: The illustration of Rules 2 and 3.

4.1 Proof of Optimality of Rules 1 and 2

The optimality of Rule 1 follows from the following:

LEMMA 1. *Suppose that there are direct potential edges between C_S and C_D. Then for any policy A that tests an edge other than such a direct potential edge, there is a policy $\widetilde{A}$ that does test a direct potential edge and has a lower expected cost than A.*

PROOF. See Appendix A. □

Next, assuming that Rule 1 is always followed, we prove the optimality of Rule 2. The following lemma establishes a preliminary property.

LEMMA 2. *When the policy follows both Rules 1 and 2, all the edges in the minimum cut at any step will be between $\cup_{i=1}^r C_i$ and C_S, or they will all be between $\cup_{i=1}^r C_i$ and C_D.*

PROOF. A minimum set partitions the connected components into two classes, with one class containing C_S and another class containing C_D. An equivalent claim of the lemma is that at any step t, all the components other than C_S or C_D will lie in one class. Suppose that not all the components lie in the same class. Then, in order to determine the S-D disconnectivity, the edges between components, in particular those that traverse two classes, say C_i and C_j will need to be tested, in addition to those between C_S and the components in C_i, and those between C_D and the components in C_j. Therefore, extra steps are wasted on testing the edges between components. However, those edges will never need to be tested when a cut is chosen to make all the components lie within one class. □

Therefore, to prove Rule 2 in our scenario we only need to prove Lemma 3 stated as follows.

LEMMA 3. *When there are no direct edges between C_S and C_D, listing all the potential shortest paths and*

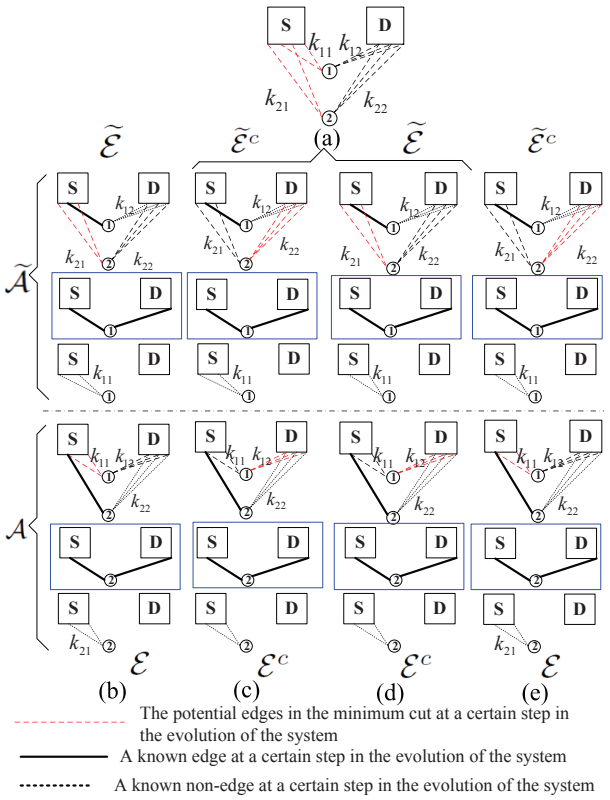

Figure 4: The illustration of the four cases $(\widetilde{\mathcal{E}},\mathcal{E})$, $(\widetilde{\mathcal{E}}^c,\mathcal{E}^c)$, $(\widetilde{\mathcal{E}},\mathcal{E}^c)$ and $(\widetilde{\mathcal{E}}^c,\mathcal{E})$.

sampling the edges in the minimum set on them will lead to smaller expected cost than sampling any other edge first.

PROOF. See Appendix B. □

4.2 Proof of Optimality of Rule 3

Proving Rule 3, it is equivalent to proving that in choosing between any two components from the set $\{C_1, \ldots, C_r\}$ at step t, the expected cost will be smaller if we first pick the component with a larger number of direct edges to the other side. Without loss of generality, let us suppose that there are only two components, C_1 and C_2. Based on Lemma 2, all the edges in the minimum cut $\mathscr{M}_{\mathcal{G}_t}$ are always on the same side. Let k_{11} and k_{21} be the number of edges from $\mathscr{M}_{\mathcal{G}_t}$ that connect to C_1 and C_2, respectively, as shown in Figure 4(a). Similarly, let k_{12} and k_{22} be the number of direct edges C_1 and C_2 have to C_D. Assuming that $k_{12} \geq k_{22}$ in Figure 4, we have the following lemma regarding Rule 3.

LEMMA 4. *Among all the potential edges that C_1 and C_2 connect to in $\mathscr{M}_{\mathcal{G}_t}$, it incurs smaller expected cost to first test the ones that C_1 is connected to.*

PROOF. We prove this by induction on the number of potential edges, in the graph of the type that results at each step from following Rules 1 and 2. Based on Lemma 2, such type of graph contains singleton components with the edges in the minimum cut all being between C_S and singletons, or all being between C_D and singletons. Clearly, Rules 1, 2 and 3 are all true for the graph with 2 potential edges. Assume that Rule 3 is true for graphs with k potential edges. We now consider the case where the graph has $k+1$ edges. Similar to the proof of Rule 1, we introduce two policies $\widetilde{\mathcal{A}}$ and $\mathcal{A}$, both of which always follow Rules 1 and 2. $\widetilde{\mathcal{A}}$ first tests an edge connecting C_S to C_1, as prescribed by Rule 3. If it finds an edge, then by Rule 1 it subsequently tests edges from C_1 to C_D. If not, according to Rule 2, the same cut continues to be minimum, and by induction, since the number of potential edges is reduced by one after the first test, it continues to test the remaining edges between C_1 and C_S. In contrast, the policy $\mathcal{A}$ violates Rule 3 on the first test by testing an edge connecting to C_2, and then follows the optimal policy for the graph with k edges. We have four possible cases, which are listed as follows:

$\widetilde{\mathcal{E}}$: $\widetilde{\mathcal{A}}$ finds an edge between C_S and C_1, and subsequently no edges between C_1 and C_D, and then the minimum cut turns out to be on the same side as C_S; or $\widetilde{\mathcal{A}}$ finds an edge between C_S and C_1 and subsequently an edge between C_1 and C_D; or $\widetilde{\mathcal{A}}$ finds no edges between C_S and C_1.

$\widetilde{\mathcal{E}}^c$ $\widetilde{\mathcal{A}}$: finds an edge between C_S and C_1, and subsequently no edges between C_1 and C_D, and the minimum cut therefore subsequently switches to the side of C_D.

$\mathcal{E}$: $\mathcal{A}$ finds an edge between C_S and C_2, and subsequently no edges between C_2 and C_D, and then the minimum cut turns out to be on the same side as C_S; or $\mathcal{A}$ finds an edge between C_S and C_2 and subsequently an edge between C_2 and C_D; or $\mathcal{A}$ finds no edges between C_S and C_2.

$\mathcal{E}^c$: $\mathcal{A}$ finds an edge between C_S and C_2, and subsequently no edges between C_2 and C_D, and the minimum cut therefore subsequently switches to the side of C_D.

The four cases, i.e., $(\widetilde{\mathcal{E}},\mathcal{E})$, $(\widetilde{\mathcal{E}}^c,\mathcal{E}^c)$, $(\widetilde{\mathcal{E}},\mathcal{E}^c)$ and $(\widetilde{\mathcal{E}}^c,\mathcal{E})$ are illustrated in Figures 4(b), (c), (d) and (e), respectively. Consider Figure 4(b), which is the case $(\widetilde{\mathcal{E}},\mathcal{E})$. We will prove using a stochastic coupling argument that testing C_1 first leads to smaller expected cost. We couple the edges labeled with the same symbol (e.g., β) tested under $\widetilde{\mathcal{A}}$ and $\mathcal{A}$, as shown in Figures 5 (a), (b), (c) and (d). Note in particular that, edges in α and β under policy $\widetilde{\mathcal{A}}$ are switched under policy $\mathcal{A}$, and edges in θ and ε under policy $\widetilde{\mathcal{A}}$ are also switched under policy $\mathcal{A}$, as shown in Figures 5 (a) and (b). The edges in δ, λ, γ and η under policy $\widetilde{\mathcal{A}}$ are the same as those in δ, λ, γ and η under policy $\mathcal{A}$, as shown in Figures 5 (c) and (d). The edges with different labels are defined in Table 2. Since testing each potential edge has two possible outcomes depending on whether it exists or not, the corresponding sample paths generated under $\widetilde{\mathcal{A}}$ and $\mathcal{A}$ are

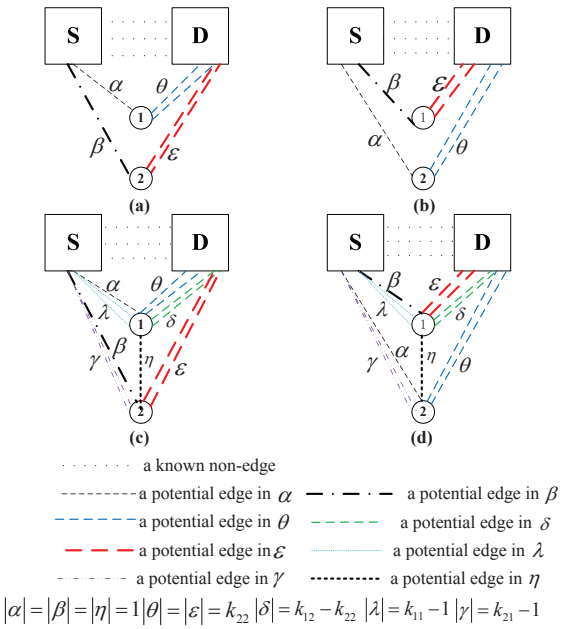

Figure 5: The stochastic coupling of the edges under policy $\widetilde{\mathcal{A}}$ (shown in (a)) and policy $\mathcal{A}$ (shown in (b)).

illustrated in Figures 6 (a) and (b), respectively. The nodes in the figures represent the tested edges, while the outcome of each tested edge is indicated by a label, 1 if it exists, or 0 otherwise. Let $P_{D,i}$ $(i = 1, 2, \ldots)$ be the ith path where C_S and C_D are found to be disconnected under $\widetilde{\mathcal{A}}$ $(\mathcal{A})$, and let $P_{C,i}$ be $(i = 1, 2, \ldots)$ the ith path where C_S and C_D are found to be connected under $\widetilde{\mathcal{A}}$ $(\mathcal{A})$. Note that the paths with the same label have the same probability. As an example, consider paths labeled as $P_{D,4}$ in both Figures 6 (a) and (b). We can see that the outcome 0 from node α in Figure 6(a) has the same probability as the outcome 0 from nodes β in Figure 6(b). The outcome 1 from node β in Figure 6(a) has the same probability as the outcome 1 from node α in Figure 6(b). Nodes α and β have the same number of edges, and the outcomes of the edges from nodes θ, δ, ε and λ are the same in both figures. Another example is the paths labeled with $P_{C,2}$ in both figures. It can be seen that the outcome 1 from node θ in Figure 6(a) has the same probability as the outcome 1 from node ε in Figure 6(b). Nodes θ and ε have the same number of edges, and the outcomes of the edges emanating from nodes α and λ are the same in both figures. Similar relationships hold for the remaining paths with the same labels in Figures 4 (a) and (b), and we therefore omit their explanations. As for the paths $\widetilde{P}_{D,1}$ and $\widetilde{P}_{D,2}$ in Figure 6(a) and paths $\widehat{P}_{D,1}$ and $\widehat{P}_{D,2}$ in Figure 6(b), it is trivially true that any terminating time belonging to the range $[k_{12} + 2, 1 + k_{11} + k_{12}]$ under the paths $\widetilde{P}_{D,1}$ and $\widetilde{P}_{D,2}$ can also be found under the paths $\widehat{P}_{D,1}$ and $\widehat{P}_{D,2}$. Therefore, the only paths that have different probabilities are shown in the form of bold lines under both $\widetilde{\mathcal{A}}$ and $\mathcal{A}$ (specifically, path $\alpha\theta^c\delta$ under $\widetilde{\mathcal{A}}$, and paths $\alpha\theta^c\beta\delta$, $\alpha\theta^c\beta^c\lambda\delta$ and $\alpha^c\lambda^c\beta\varepsilon^c\delta$ under A.). Obviously, $\alpha\theta^c\delta$ dominates the paths $\alpha\theta^c\beta\delta$, $\alpha\theta^c\beta^c\lambda\delta$ and $\alpha^c\lambda^c\beta\varepsilon^c\delta$, which indicates that testing C_1 first will lead to smaller expected cost. This completes the proof of the case in Figure 4(b).

Table 2: Definitions of the edges for coupling $\widetilde{\mathcal{A}}$ and $\mathcal{A}$

Notation	Definition
α	The first potential edge that policy $\widetilde{\mathcal{A}}$ $(\mathcal{A})$ tests.
β	The first potential edge that policy $\widetilde{\mathcal{A}}$ $(\mathcal{A})$ tests between C_S and C_2 (C_1).
θ	The first k_{22} potential edges that policy $\widetilde{\mathcal{A}}$ $(\mathcal{A})$ tests after an edge is found between C_1 (C_2) and C_S. Recall that $k_{12} \geq k_{22}$.
ε	k_{22} potential edges to be tested by policy $\widetilde{\mathcal{A}}$ $(\mathcal{A})$ after β is found to exist.
δ	The $k_{12} - k_{22}$ remaining potential edges yet to be tested by policy $\widetilde{\mathcal{A}}$ $(\mathcal{A})$ between C_D and C_1, after no edges are found to exist in θ (ε).
λ	The $k_{11} - 1$ remaining potential edges yet to be tested by policy $\widetilde{\mathcal{A}}$ $(\mathcal{A})$ between C_S and C_1, after the first edge between C_S and C_1 is found to not exist.
γ	The $k_{21} - 1$ remaining potential edges between C_S and C_2.
η	The potential edge between C_1 and C_2.

Next we turn to the case $(\widetilde{\mathcal{E}}^c,\mathcal{E}^c)$ shown in Figure 4(c). The paths generated under $\widetilde{\mathcal{A}}$ and $\mathcal{A}$ using coupling are illustrated in Figures 7 (a) and (b), respectively. Again, the paths with the same label have the same probability, and they can be checked as in the aforementioned case $(\widetilde{\mathcal{E}},\mathcal{E})$. As for the paths $\widetilde{P}_{D,1}$ and $\widetilde{P}_{D,2}$ in Figure 7(a) and paths $\widehat{P}_{D,1}$ and $\widehat{P}_{D,2}$ in Figure 7(b), it is trivially true that any terminating time belonging to the range $[k_{12} + k_{21} + k_{22} + 2, k_{11} + k_{12} + k_{21} + k_{22} + 1]$ under the paths $\widetilde{P}_{D,1}$ and $\widetilde{P}_{D,2}$ can also be found under the paths $\widehat{P}_{D,1}$ and $\widehat{P}_{D,2}$. The same conclusion also holds for the paths $\widetilde{P}_{C,1}$ and $\widetilde{P}_{C,2}$ in Figure 7(a), and paths $\widehat{P}_{C,1}$ and $\widehat{P}_{C,2}$ in Figure 7(b). Hence, we highlight the paths that will terminate in different time with bold lines under both $\widetilde{\mathcal{A}}$ and $\mathcal{A}$ (specifically path $\alpha\theta^c\delta$ under $\widetilde{\mathcal{A}}$ and paths $\alpha\theta^c\delta\beta$, $\alpha\theta^c\delta\beta^c\lambda$ and $\alpha^c\lambda^c\beta\varepsilon^c\delta$ under $\mathcal{A}$.). Obviously, $\alpha\theta^c\delta$ terminates earlier than $\alpha\theta^c\delta\beta$, $\alpha\theta^c\delta\beta^c\lambda$ as well as $\alpha^c\lambda^c\beta\varepsilon^c\delta$, which indicates that testing C_1 first will lead to smaller expected cost. This completes the proof of the case in Figure 4(c).

Now we proceed to prove cases $(\widetilde{\mathcal{E}},\mathcal{E}^c)$ and $(\widetilde{\mathcal{E}}^c,\mathcal{E})$, shown in Figures 4(d) and 4(e), respectively. We first consider case $(\widetilde{\mathcal{E}},\mathcal{E}^c)$. This case implies that the minimum cut turns out to be on the same side of C_S, after an edge is found between C_1 and C_S but no edges are found between C_1 and C_D by $\widetilde{\mathcal{A}}$. Due to Rule 2, it leads to smaller expected cost for $\widetilde{\mathcal{A}}$ to subsequently test the

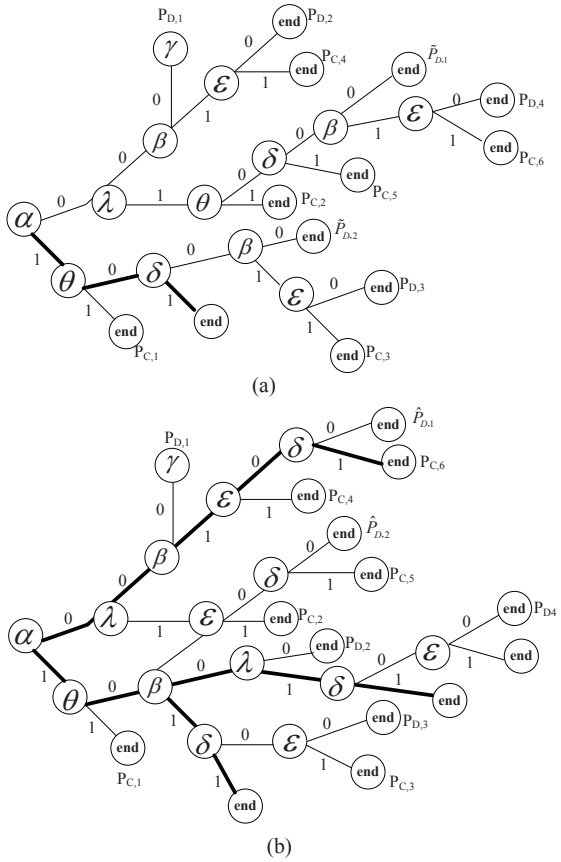

Figure 6: Paths generated by the coupling of $\widetilde{\mathcal{A}}$ (a) and $\mathcal{A}$ (b) for the case shown in Figure 4(b).

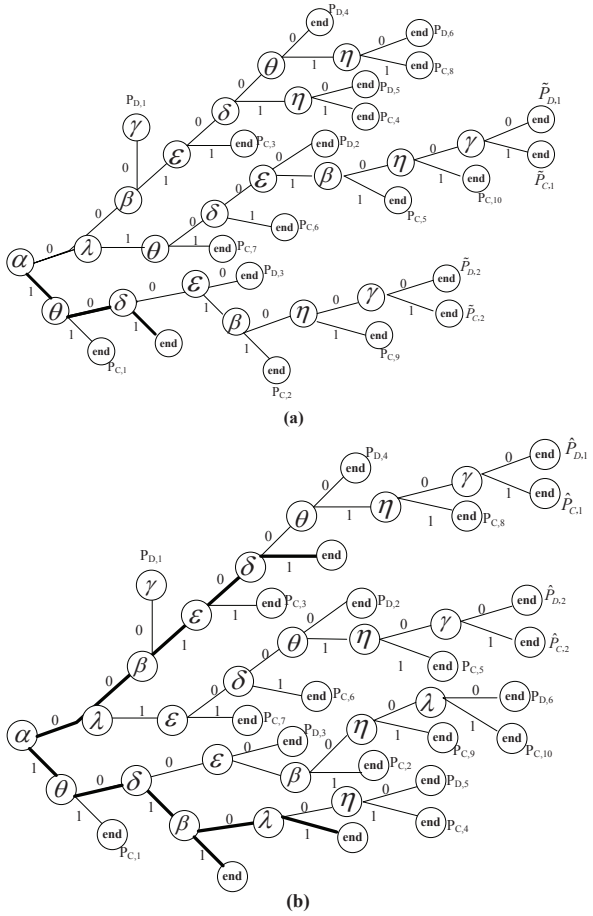

Figure 7: Paths generated by the coupling of $\widetilde{\mathcal{A}}$ (a) and $\mathcal{A}$ (b) for the case shown in Figure 4(c).

edges between C_2 and C_S rather than those between C_2 and C_D. This means the expected cost of case $(\widetilde{\mathcal{E}}, \mathcal{E}^c)$ is smaller than that of case $(\widetilde{\mathcal{E}}^c, \mathcal{E}^c)$. Moreover, we have already proved that the expected cost is smaller under policy $\widetilde{\mathcal{A}}$ in case $(\widetilde{\mathcal{E}}^c, \mathcal{E}^c)$. By transitivity, we conclude that $\widetilde{\mathcal{A}}$ leads to smaller cost than $\mathcal{A}$ does in case $(\widetilde{\mathcal{E}}, \mathcal{E}^c)$. Now consider the last case, $(\widetilde{\mathcal{E}}^c, \mathcal{E})$. This case implies that the minimum cut switches to the side of C_D, after an edge is found between C_1 and C_S but no edges are found between C_1 and C_D by $\widetilde{\mathcal{A}}$. According to Rule 2, it incurs smaller expected cost if $\widetilde{\mathcal{A}}$ subsequently tests the edges between C_2 and C_D rather than those between C_2 and C_S. Since we have already proved that the cost is smaller under policy $\widetilde{\mathcal{A}}$ in case $(\widetilde{\mathcal{E}}, \mathcal{E})$, it follows that by transitivity that $\widetilde{\mathcal{A}}$ leads to smaller expected cost than $\mathcal{A}$ in case $(\widetilde{\mathcal{E}}^c, \mathcal{E})$. □

The low complexity of the policy, of no more than $30 \log_2 n$ operations per step, follows from the fact that except for C_S and C_D, all other components are singletons, and moreover are only of three kinds, either having a known non-edge to S, or a known non-edge to D, or neither. Hence all computations involving these components are extremely simple.

5. EXTENSION TO GENERAL GRAPHS

In this section, we indicate extensions of the optimality results, without proofs, to certain more general classes of random graphs.

5.1 (1,p) Random Graphs

By a (1,p) random graph, we mean a graph that initially contains two types of edges, "1" edges and "p" edges. The former are edges that are known to exist, i.e., exist with probability one, while the latter are potential edges that exist with probability p. The Alternating Policy remains optimal under the condition that the size of each component, excluding C_S and C_D, is the same.

5.2 (1,0,p) Random Graphs

By a (1,0,p) random graph, we mean a graph that also initially contains "0" edges. A "0" edge is simply an edge that we already know to not exist. Unfortunately, determining the optimal policy for all types of (1,0,p) random graphs appears to be intractable. The Alternating Policy remains optimal for certain types of

(1,0,p) graph patterns – series graphs, parallel graphs, SP graphs, PS graphs, series of parallel of series (SPS) graphs and parallel of series of parallel (PSP) graphs.

1. Optimal policy for Series of Parallel (SP) Graphs

An SP graph consists of several parallel graphs arranged in series, as shown in Figure 8. It consists of n parallel graphs labeled $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$, with m_i potential edges (i.e., existing independently with equal probability p) in the ith parallel graph.

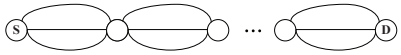

Figure 8: A series of parallel (SP) graph.

THEOREM 1. *The optimal policy is to test the parallel subgraph with the fewest number of potential edges.*

2. Optimal Policy for Parallel of Series (PS) Graphs

A PS graph consists of several series graphs arranged in parallel, as shown in Figure 9. It consists of n series graphs labeled $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$, with m_i potential edges in the ith series graph.

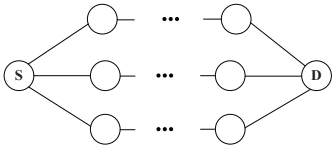

Figure 9: A parallel of series (PS) graph.

THEOREM 2. *The optimal policy is to test any edge on the series subgraph that contains the fewest number of potential edges.*

5.2.1 The Optimality Policy for SPS Graphs

An SPS graph consists of several PS graphs arranged in series, as shown in Figure 10. It consists of n PS graphs labeled $\mathcal{PS}_1, \mathcal{PS}_2, \ldots, \mathcal{PS}_n$, with m_i potential edges in the ith ($1 \leq i \leq n$) PS graph.

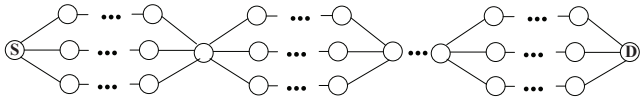

Figure 10: A series of parallel of series (SPS) graph.

THEOREM 3. *The optimal policy is to test any edge in that series graph which has the fewest number of edges in the PS graph that contains the minimum number of series graphs.*

5.2.2 The Optimal Policy for PSP Graphs

A PSP graph consists of several SP graphs arranged in parallel, as shown in Figure 11. It consists of n SP

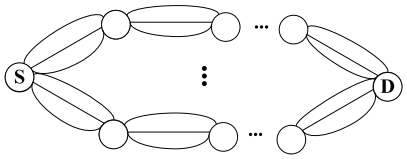

Figure 11: A parallel of series of parallel (PSP) graph.

graphs labeled as $\mathcal{SP}_1, \mathcal{SP}_2, \ldots, \mathcal{SP}_n$, with m_i potential edges in the ith ($1 \leq i \leq n$) SP graph.

THEOREM 4. *The optimal policy is to test any edge in that parallel graph which contains the fewest number of edges in the SP graph that contains the minimum number of parallel graphs.*

6. CONCLUSION

We have studied the problem of optimally determining S-D connectivity of random graphs. We have considered the classic ER Random Graphs with a finite number n of nodes, for any value of n. Assuming that each testing of an edge has the same unit cost, we have determined a policy for establishing whether the designated source and destination are connected with minimum expected cost. Interestingly, though ER graphs exhibit a sharp transition between disconnectivity of the entire graph and connectivity of the entire graph around $p = \ln n/n$, the optimal policy turns out to not depend on the specific value of p, or even of which side of the phase transition p lies. The policy simply contracts each known connected component to a single super node at each step, and in that condensation multigraph it simply tests an edge that is both on the shortest path containing the super nodes containing S and D, as well as on a minimum S-D cut, giving higher priority to sampling the edges that will lead to more opportunities for S-D connectivity. The policy is also extendable to certain types of more generalized graph structures consisting of additional deterministic structure.

Acknowledgement

This paper is partially based on work supported by NSF under Contract Nos. CNS-1302182 and CCF-0939370, and AFOSR under Contract No. FA-9550-13-1-0008, and NSF China (No. 61325012, 61271219).

7. REFERENCES

[1] E. N. Gilbert, "Random Graphs", in Annals of Mathematical Statistics 30 (4), pp. 1141-1144, 1959.
[2] P. Erdos, A. Renyi, "On Random Graphs. I", in Publicationes Mathematicae 6: pp. 290-297, 1959.
[3] P. Erdos, A. Renyi, "On the evolution of random graphs", in *the Mathematical Institute of the Hungarian Academy of Sciences* 5, pp. 17-61, 1960.
[4] P. Erdos, E. M. Palmer and R. W. Robinson, "Local Connectivity of A Random Graph", in *J. of Graph Theory*, Vol. 7, No. 4, pp. 411-417, 1983.

[5] B. Bollobas, S. Janson and O. Riordan, "The Phase Transition in Inhomogeneous Random Graphs", in *J. Random Structures and Algorithms*, Vol. 31, No. 1, pp. 3-122, 2007.
[6] G. Han and A. M. Makowski, "One-dimensional Geometric Random Graphs with Nonvanishing Densities: Part I: A Strong Zero-one Law for Connectivity", in *J. IEEE. Trans. on Inform. Theory*, Vol. 55, No. 12, pp. 5832-5839, 2009.
[7] M. D. Penrose. "On K-connectivity for a Geometric Random Graph," Random Structure and Algorithms, Vol. 15, No. 2, pp. 145-164, Sept. 1999.
[8]

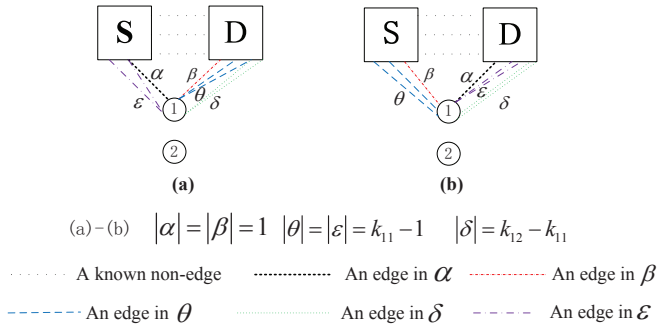

Figure 12: The edges coupled under policies $\widetilde{\mathcal{A}}$ (a) and $\mathcal{A}$ (b).

Now we consider the policy $\mathcal{A}^2$, and define T^2 similarly. Then we have

$$\begin{aligned}\mathbb{E}[T^1 - T^2] &= \mathbb{E}[\mathbb{1}\left(T^2 = 2\right)(1-2) + \mathbb{1}\left(T^2 < 2, e_d = 1\right)(1-1) \\ &\quad + \mathbb{1}\left(T^2 < 2, e_d = 0\right)|T|] \\ &= \mathbb{E}\left[\mathbb{1}(T^2 = 2)(-1) + \mathbb{1}(T^2 = 1, e_d = 0)\right] \\ &= -p + 0 = -p.\end{aligned}$$

Hence $\mathcal{A}^1$ has lower cost than $\mathcal{A}^2$. Similarly, we can prove that each $\mathcal{A}^i$ has lower cost than $\mathcal{A}^{i+1}$. Noting that $\mathcal{A}^N = \mathcal{A}$, we thus conclude that $\mathcal{A}^1$ has lower cost than $\mathcal{A}$, proving the lemma. □

Appendix B: Proof of Lemma 3

We prove by induction on the number of steps. Clearly, by Lemma 1, the conclusion holds at step 1. Suppose it is true till a certain step t. From Lemma 2, it follows that all the edges in the minimum cut at step t are either all between C_S and $\cup_{i=1}^{r} C_i$, or they are all between C_D and $\cup_{i=1}^{r} C_i$. Furthermore, since all those edges are also along all the potential shortest paths, they also belong to the set $\mathcal{L}$. Without loss of generality, suppose that all the edges in $\mathcal{L}$ at step t are between C_S and $\cup_{i=1}^{r} C_i$. Now we consider step $t+1$.

We introduce two policies $\widetilde{\mathcal{A}}$ and $\mathcal{A}$, where $\widetilde{\mathcal{A}}$ always follows Rule 2. Specially, at step $t+1$, $\widetilde{\mathcal{A}}$ will test the edges in $\mathscr{M}_{\mathcal{G}_t}$, starting from an edge between some component, say C_1, and C_S. If an edge is found between C_1 and C_S, then, due to Rule 1, $\widetilde{\mathcal{A}}$ subsequently tests all the edges between C_1 and C_D. If not, then the same cut continues to be the minimum cut, and $\widetilde{\mathcal{A}}$ continues to test the remaining edges between C_1 and C_S. In contrast, $\mathcal{A}$ violates Rule 2 at step $t+1$ by testing an edge between C_1 and then follows $\widetilde{\mathcal{A}}$.

We prove that testing the edges between C_1 and C_S leads to smaller expected cost , by again employing a stochastic coupling. Note that the number of edges between C_1 and C_S can be either larger or smaller than the number of the edges between C_1 and C_D. We only consider the former case; the latter is proved similarly. We couple the edges tested under $\widetilde{\mathcal{A}}$ and $\mathcal{A}$, as shown in Figures 12. Table 3 lists the permutations of the nodes labels involved in the coupling. The paths generated under both $\widetilde{\mathcal{A}}$ and $\mathcal{A}$ for the cases in Figures 12(a) and (b) are shown in Figure 13. The nodes in the figures represent the edges tested while the outcomes of the tested edges are indicated using lines with 1 (or 0) meaning that the edges turn out to exist (or not exist). The paths that terminate at the same time are marked with the same labels under $\widetilde{\mathcal{A}}$ and $\mathcal{A}$. The two paths highlighted in bold lines in Figure 13(b), i.e., $\hat{P}_{D,1}$ and $\hat{P}_{D,2}$, represent the graph states where none of the edges between C_1 and C_S exist while one edge between C_1 and C_D is found to exist in $\hat{P}_{D,2}$ and not to exist in $\hat{P}_{D,1}$ under policy $\mathcal{A}$. The path in bold line in Figure 13(a) represents the graph state where none of the edges between C_1 and C_S exist under policy $\widetilde{\mathcal{A}}$, with the circle containing dots being the possible edges that $\widetilde{\mathcal{A}}$ will subsequently test. Notice that if the circle represents an edge between C_1 and C_D, i.e., if $\widetilde{\mathcal{A}}$ chooses to test an edge between C_1 and C_D, then both Figures 13(a) and (b) will exhibit the same termination probability. However, the expected cost will be smaller if $\widetilde{\mathcal{A}}$ tests any edge between C_S (C_D) and one of the remaining components. Similarly, we can proceed to prove that the expected cost can be even smaller if the edge tested belongs to the remaining edges in $\mathcal{L}$. The proof is completed by checking all the components. □

Table 3: Definitions of the edges coupled under $\widetilde{\mathcal{A}}$ and $\mathcal{A}$

Notation	Definition
α	The first potential edge that policy $\widetilde{\mathcal{A}}$ ($\mathcal{A}$) tests.
β	The first potential edge that policy $\widetilde{\mathcal{A}}$ ($\mathcal{A}$) tests between C_1 and C_D (C_S).
θ	The $k_{11}-1$ potential edges that policy $\widetilde{\mathcal{A}}$ ($\mathcal{A}$) tests between C_1 and C_D (C_S) after β is found not to exist between C_1 and C_D (C_S).
δ	The remaining $k_{12}-k_{11}$ ($k_{11}-k_{12}$) potential edges to be tested by policy $\widetilde{\mathcal{A}}$ ($\mathcal{A}$) between C_D and C_1.
ε	The remaining $k_{11}-1$ potential edges between C_1 and C_S to be tested by policy $\widetilde{\mathcal{A}}$ and the $k_{11}-1$ potential edges between C_1 and C_D to be tested by policy $\mathcal{A}$.

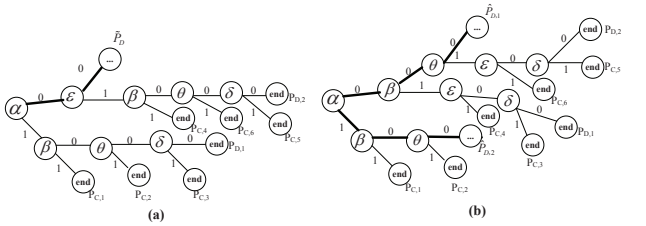

Figure 13: Paths generated by the coupling on $\widetilde{\mathcal{A}}$ (a) and $\mathcal{A}$ (b) for the case shown in Figures 12 (a)-(b).