An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 40 (2000) 359–372
www.elsevier.nl/locate/camwa

A Network Reduction Axiom for Efficient Computation of Terminal-Pair Reliability

S. J. HSU AND M. C. YUANG
Department of Computer Science and Information Engineering
National Chiao Tung University, Taiwan, R.O.C.

(Received May 1998; revised and accepted August 1998)

Abstract—Terminal-pair reliability (TR) in network management determines the probabilistic reliability between two nodes (the source and sink) of a network, given failure probabilities of all links. It has been shown that TR can be effectively computed by means of the network reduction technique. Existing reduction axioms, unfortunately, are limited to trivial rules such as valueless link removal and series-parallel link reduction. In this paper, we propose a novel reduction axiom, referred to as *triangle reduction*. The triangle reduction axiom transforms a graph containing a *triangle subgraph* to that excluding the base of the triangle. The computational complexity of the transformation is as low as $O(1)$. With triangle reduction, the number of subproblems generated by partition-based TR algorithms, for simplified grid networks, can be reduced to $O(((1+\sqrt{5})/2)^n)$. The paper further provides an assessment of the effectiveness of triangle reduction on partition-based TR algorithms with respect to the number of subproblems and computation time through experimenting on published benchmarks and random networks. Experimental results demonstrate that, incorporating triangle reduction, the path-based (cut-based) partition TR algorithm yields a substantially reduced number of subproblems and computation time for all (most of the) benchmarks and random networks.

Keywords—Terminal-pair reliability (TR), Path-based partition, Cut-based partition, Network reduction technique.

1. INTRODUCTION

The analysis of network reliability has been given considerable attention in network management. In particular, terminal-pair reliability (TR) [1–14] deals with the determination of the probabilistic reliability between two nodes (the source and sink) of a network, given failure probabilities of all links. Existing TR algorithms, which are based on the partition technique, such as the cut-based [2,6] and path-based algorithms [7], achieve efficient TR computation by means of simple network reduction rules [6,7,11], such as valueless link removal and series-parallel link reduction.

The goal of the paper is to propose a novel reduction axiom [9], referred to as *triangle reduction.* The triangle reduction axiom basically transforms a graph, in which the source is only adjacent to two one-way or two-way connected nodes, forming a triangle subgraph, to a simpler graph with the link(s) incident with the two nodes removed. The resulted success probabilities of the corresponding links, connecting the source to the two nodes, are reassigned via closed-form

PII: S0898-1221(00)00166-8

equations. The computational complexity of the transformation is as low as $O(1)$. Incorporating the triangle reduction axiom, we prove that the number of subproblems generated by partition-based TR algorithms, for simplified grid networks, is reduced to $O(((1+\sqrt{5})/2)^n)$. The paper further provides an assessment of the effectiveness of triangle reduction on partition-based TR algorithms with respect to the number of subproblems and computation time through experimenting on published benchmarks and random networks. Our experimental results demonstrate that, incorporating the triangle reduction, the path-based (cut-based) partition TR algorithm yields a substantially reduced number of subproblems and computation time for all (most of the) benchmarks and random networks.

This paper is organized as follows. Section 2 gives an overview of the two partition-based TR algorithms, namely the cut-based and path-based algorithms. The new triangle reduction axiom is proposed in Section 3. Section 4 analyzes the reduction efficiency with and without triangle reduction, for simplified grid network. Section 5 provides performance assessment via experiments on benchmarks and random networks. Finally, conclusion remarks are given in Section 6.

2. OVERVIEW OF PARTITION-BASED TR ALGORITHMS

Existing partition-based TR algorithms, employing the traditional reduction technique, can be categorized [2,6,7] as: path-based partition with reduction (PPR), and cut-based partition with reduction (CPR). In both algorithms, networks are modeled as directed graphs with each link associated with a failure probability. These failure probabilities are assumed to be statistically independent. While PPR and CPR have great similarity in nature, they differ in the selection of the partition basis. Each of them is further described in detail as follows.

2.1. Path-Based Partition with Reduction (PPR) Algorithm

The PPR algorithm [7] computes terminal-pair reliability, $\mathrm{Rel}(G)$, from source s to sink t in network G by Boolean algebra. First, the network is simplified by employing the network reduction technique [6,11], including removing valueless links (such as entering the source) and series-parallel link reduction, as shown in Figure 1. The path-based partition is in turn performed based on the shortest $s-t$ path, which is a set of links, $\{e_1, e_2, \dots, e_l\}$, constituting the shortest path from s to t. Based on the factoring theorem [11], the problem is decomposed into a set of subproblems. That is, $\mathrm{Rel}(G) = q_1 \times \mathrm{Rel}(G-e_1) + p_1 q_2 \times \mathrm{Rel}(G*e_1-e_2) + \cdots + p_1 p_2 \dots p_{l-1} q_l \times \mathrm{Rel}(G*e_1*e_2*\cdots*e_{l-1}-e_l) + p_1 p_2 \dots p_{l-1} p_l$, where $p_i(q_i)$ represents the success (failure) probability of link e_i, "$*$" ("$-$") represents the contracting (deleting) operation of links, and Rel()'s correspond to the subproblems. The same reduction and partition procedures are recurrently applied to each newly generated subproblem until the source and sink are disconnected.

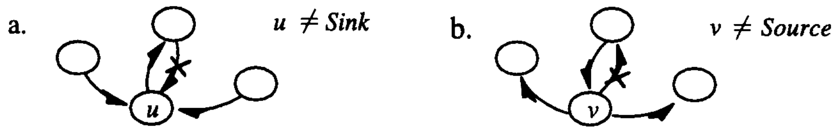
2.2. Cut-Based Partition with Reduction (CPR) Algorithm

Similar to PPR, CPR [2,6] initially simplifies the network by using the network reduction technique. Rather than partition based on the shortest $s-t$ path, CPR employs the cut-based partition by means of the source-cut consisting of all links emanating from the source. Given source-cut $\{e_1, e_2, \dots, e_l\}$, based on the factoring theorem, a number of subproblems are similarly generated. The same reduction and partition procedures are recursively applied to each newly generated subproblem until the source and sink are contracted or disconnected.

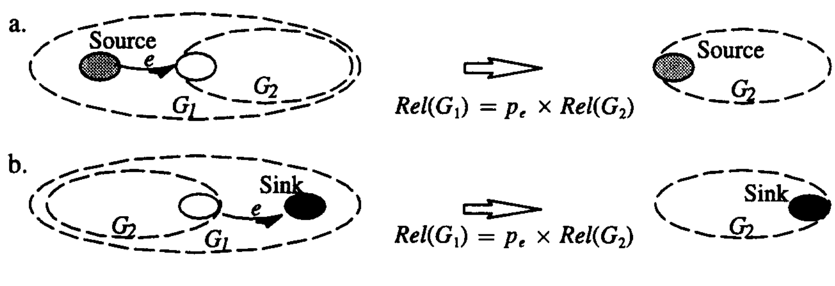
An example of how PPR and CPR algorithms perform is illustrated in Figure 2. Given a network (Figure 2a), based on the PPR algorithm (Figure 2b), according to reduction rule **r5**, serial links e_1 and e_2 can be first reduced to e_7 with success probability p_7, where $p_7 = p_1 p_2$. Then, after the shortest-path-based partition and factoring, $\mathrm{Rel}(G)$ is decomposed as $\mathrm{Rel}(G) = q_7 \times \mathrm{Rel}(G-e_7) + p_7 q_5 \times \mathrm{Rel}(G*e_7-e_5) + p_7 p_5$. In the case of CPR, the network is first simplified reducing serial links e_1 and e_2 to e_7. After the source-cut-based partition and factoring, $\mathrm{Rel}(G)$

r1. Links entering the source or exiting from the sink are valueless.

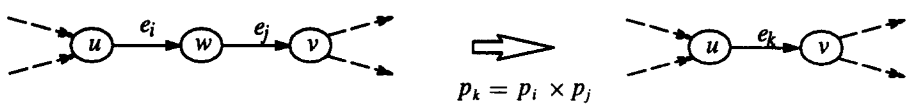
r2. Nodes (except source and sink) with no output links or input links are valueless.



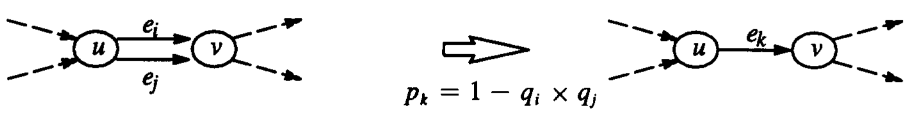
r3. Links antiparallel to node's single input link or output link are valueless.



r4. A single link going out of the source or into the sink could be contracted.



r5. Series link reduction.



r6. Parallel link reduction.

Figure 1. Existing reduction rules.

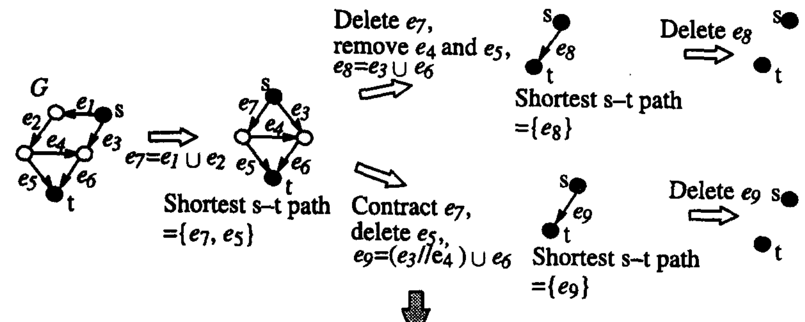
is decomposed as $\text{Rel}(G) = p_7 \times \text{Rel}(G * e_7) + q_7 p_3 \times \text{Rel}(G - e_7 * e_3)$. In both case, all newly generated subproblems are continuously processed until s and t are contracted or disconnected.

3. TRIANGLE REDUCTION AXIOM

The triangle reduction axiom [9] is applied to a *source-based triangle subgraph* of a graph representing the network under consideration. A subgraph is defined as a source-based triangle subgraph if it contains the source and two one-way or two-way connected nodes to which the source is only adjacent, forming a triangle, as shown in Figure 3a. Notice that the notion of the triangle subgraph can be similarly applied to a subgraph including the sink instead (sink-based), as shown in Figure 3b. For simplicity, without further declaration, the triangle subgraph referred throughout the rest of the paper is source-based. Notice that the concept of the triangle reduction cannot be applied to the cases in which the source (sink) is incident with more than two outgoing (incoming) edges due to exponentially increased complexity.

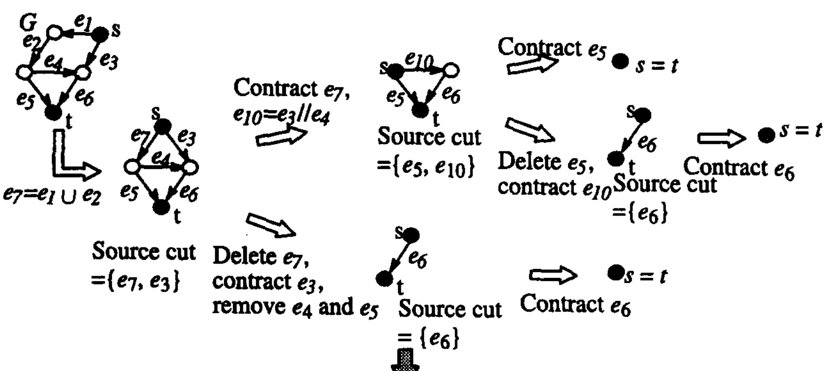
In Figure 3a, the two nodes to which the source is adjacent are denoted as n_1 and n_2. The two links connecting from s to n_1 and n_2, referred to as the *sides* of the triangle, are labeled as e_{s1}

(a) Network G.



$$Rel(G) = p_7p_5 + q_7p_8 + p_7q_5p_9 = p_1p_2p_5 + (1-p_1p_2)p_3p_6 + p_1p_2q_5(1-q_3q_4)p_6$$

(b) The PPR algorithm.



$$Rel(G) = p_7(p_5 + q_5p_{10}p_6) + q_7p_3p_6 = p_1p_2(p_5 + q_5(1-q_3q_4)p_6) + (1-p_1p_2)p_3p_6$$

(c) The CPR algorithm.

Legend:
- $p_i(q_i)$: the success (failure) probability of link e_i.
- "$\cup$": the operation of combining series links.
- $p_j = p_kp_l$, if $e_j = e_k \cup e_l$.
- "$//$": the operation of combining parallel links.
- $p_j = 1 - q_kq_l$, if $e_j = e_k // e_l$.

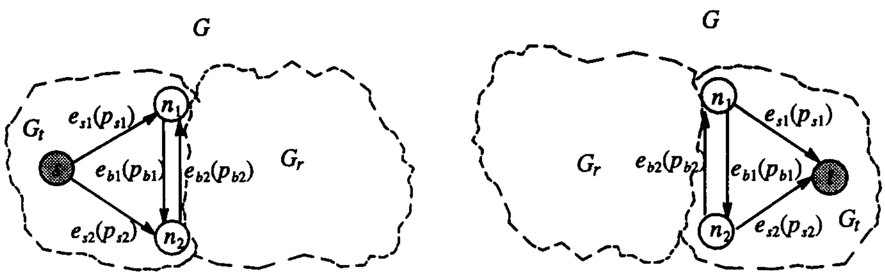
Figure 2. PPR and CPR algorithms—an example.

and e_{s2} with success probabilities p_{s1} and p_{s2}, respectively. The link connecting $n_1(n_2)$ to $n_2(n_1)$, referred to as the *base* of the triangle, is labeled as $e_{b1}(e_{b2})$ with success probability $p_{b1}(p_{b2})$. Notice that, if n_1 and n_2 are two-way connected, the base of the triangle is comprised of two links. As a result, the three nodes (s, n_1, and n_2), the sides (e_{s1} and e_{s2}), and the base (e_{b1} and/or e_{b2}), constitute a triangle subgraph, denoted as G_t.

Basically, the triangle reduction axiom transforms a graph containing a triangle subgraph to a simpler graph with the base of the triangle deleted. In the following, the axiom for the two-link base is formally stated and proved. In the case of the one-link base, similar results can be obtained by replacing p_{b1} or p_{b2} with zero.

Triangle Reduction Axiom

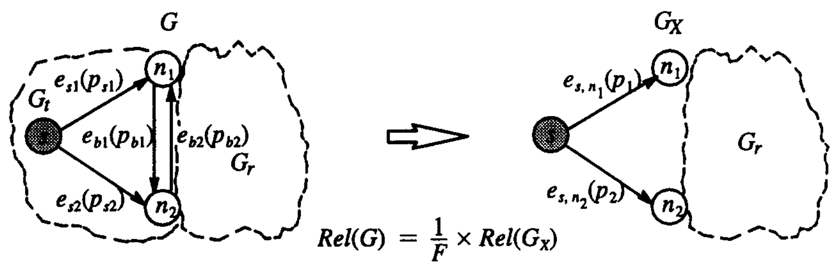
In a given graph G, as shown in Figure 4, if there exists a triangle subgraph with three nodes (s, n_1, and n_2), two sides (e_{s1} and e_{s2}), and the base (e_{b1} and/or e_{b2}), G can be transformed



(a) A source-based triangle subgraph. (b) A sink-based triangle subgraph.

Legend:
- $e_l(p_l)$: link l with success probability p_l.
- G_t: the triangle subgraph of graph G.
- G_r: graph $G - G_t$.
- s: the source node.
- t: the sink node.

Figure 3. Triangle subgraphs.



Legend:
- $e_l(p_l)$: link l with success probability p_l.
- G_t: the triangle subgraph of graph G.
- G_r: graph $G - G_t$.

Figure 4. Triangle reduction axiom.

to G_X with the base removed. The new probability p_1 of link e_{s,n_1} of G_X connecting s to n_1, and probability p_2 of link e_{s,n_2} of G_X connecting s to n_2, are reassigned as

$$p_1 = \frac{q_{s1}p_{s2}p_{b2} + p_{s1}q_{s2}p_{b1} + p_{s1}p_{s2}}{q_{s1}p_{s2}q_{b2} + q_{s1}p_{s2}p_{b2} + p_{s1}q_{s2}p_{b1} + p_{s1}p_{s2}} \tag{1}$$

and

$$p_2 = \frac{q_{s1}p_{s2}p_{b2} + p_{s1}q_{s2}p_{b1} + p_{s1}p_{s2}}{p_{s1}q_{s2}q_{b1} + q_{s1}p_{s2}p_{b2} + p_{s1}q_{s2}p_{b1} + p_{s1}p_{s2}}. \tag{2}$$

Moreover, the terminal-pair reliability of the transformed graph G_X, $\mathrm{Rel}(G_X)$, becomes the product of $\mathrm{Rel}(G)$ and the reduction factor, F

$$\mathrm{Rel}\,(G_X) = \mathrm{Rel}(G) \times F, \tag{3}$$

where

$$F = \frac{q_{s1}p_{s2}p_{b2}+p_{s1}q_{s2}p_{b1}+p_{s1}p_{s2}}{(p_{s1}q_{s2}q_{b1}+q_{s1}p_{s2}p_{b2}+p_{s1}q_{s2}p_{b1}+p_{s1}p_{s2})(q_{s1}p_{s2}q_{b2}+q_{s1}p_{s2}p_{b2}+p_{s1}q_{s2}p_{b1}+p_{s1}p_{s2})}. \tag{4}$$

PROOF. According to the factoring theorem, $\mathrm{Rel}(G)$ can be partitioned to 16 subproblems, as given in Figure 5, corresponding to four graphs, G_a, G_b, G_c, and G_d. In the figure, for example, graph G_b is related to graph G by the presence of link e_{s1} and the absence of links e_{s2} and e_{b1}. Namely, $G_b = G*e_{s1} - e_{s2} - e_{b1}$. According to reduction rules **r4**(a) and **r1**, s is contracted to n_1, and valueless link e_{b2} is removed, resulting in two equal-valued subproblems, $\mathrm{Rel}(G_b) = \mathrm{Rel}(G * e_{s1} - e_{s2} - e_{b1} - e_{b2}) = \mathrm{Rel}(G * e_{s1} - e_{s2} - e_{b1} * e_{b2})$. As a result, G_X can be associated with G_b by the presence of link e_{s,n_1} and the absence of link e_{s,n_2}, and thus, $\mathrm{Rel}(G_b) = \mathrm{Rel}(G_X * e_{s,n_1} - e_{s,n_2})$.

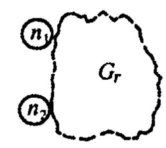
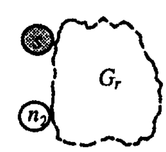
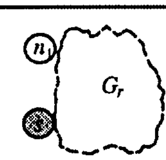
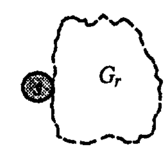
Decomposition	Subproblems for G	Corresponding graph after factoring	Subproblems for G_X
$G_a = G - e_{s1} - e_{s2}$	4 subproblems : $Rel(G_a) = Rel(G - e_{s1} - e_{s2} - e_{b1} - e_{b2})$ $= Rel(G - e_{s1} - e_{s2} - e_{b1} * e_{b2})$ $= Rel(G - e_{s1} - e_{s2} * e_{b1} - e_{b2})$ $= Rel(G - e_{s1} - e_{s2} * e_{b1} * e_{b2})$	G_a 	$Rel(G_a)$ $= Rel(G_X - e_{s,n_1} - e_{s,n_2})$ $= 0$
$G_b = G * e_{s1} - e_{s2} - e_{b1}$	2 subproblems : $Rel(G_b) = Rel(G * e_{s1} - e_{s2} - e_{b1} - e_{b2})$ $= Rel(G * e_{s1} - e_{s2} - e_{b1} * e_{b2})$	G_b 	$Rel(G_b)$ $= Rel(G_X * e_{s,n_1} - e_{s,n_2})$
$G_c = G - e_{s1} * e_{s2} - e_{b2}$	2 subproblems : $Rel(G_c) = Rel(G - e_{s1} * e_{s2} - e_{b1} - e_{b2})$ $= Rel(G - e_{s1} * e_{s2} * e_{b1} - e_{b2})$	G_c 	$Rel(G_c)$ $= Rel(G_X - e_{s,n_1} * e_{s,n_2})$
$G_d = G * e_{s1} * e_{s2}$	8 subproblems : $Rel(G_d) = Rel(G * e_{s1} - e_{s2} * e_{b1} - e_{b2})$ $= Rel(G * e_{s1} - e_{s2} * e_{b1} * e_{b2})$ $= Rel(G - e_{s1} * e_{s2} - e_{b1} * e_{b2})$ $= Rel(G - e_{s1} * e_{s2} * e_{b1} * e_{b2})$ $= Rel(G * e_{s1} * e_{s2} - e_{b1} - e_{b2})$ $= Rel(G * e_{s1} * e_{s2} - e_{b1} * e_{b2})$ $= Rel(G * e_{s1} * e_{s2} * e_{b1} - e_{b2})$ $= Rel(G * e_{s1} * e_{s2} * e_{b1} * e_{b2})$	G_d 	$Rel(G_d)$ $= Rel(G_X * e_{s,n_1} * e_{s,n_2})$
$Rel(G) = p_{s1}q_{s2}q_{b1} \times Rel(G_b) + q_{s1}p_{s2}q_{b2} \times Rel(G_c)$ $+ (p_{s1}q_{s2}p_{b1} + q_{s1}p_{s2}p_{b2} + p_{s1}p_{s2}) \times Rel(G_d)$			$Rel(G_X) =$ $p_1q_2 \times Rel(G_b)$ $+ q_1p_2 \times Rel(G_c)$ $+ p_1p_2 \times Rel(G_d)$

Figure 5. Association of Rel(G) and Rel(G_X).

Applying the same logic of relating other graphs (G_a, G_c, and G_d) to G_X, we attain

$$\begin{aligned}\mathrm{Rel}\,(G_X) &= p_1q_2 \times \mathrm{Rel}\,(G_X * e_{s,n_1} - e_{s,n_2}) + q_1p_2 \times \mathrm{Rel}\,(G_X - e_{s,n_1} * e_{s,n_2}) + p_1p_2 \\ &\quad \times \mathrm{Rel}\,(G_X * e_{s,n_1} * e_{s,n_2}) \\ &= p_1q_2 \times \mathrm{Rel}(G_b) + q_1p_2 \times \mathrm{Rel}(G_c) + p_1p_2 \times \mathrm{Rel}(G_d).\end{aligned} \tag{5}$$

In addition, notice that Rel(G) can be expressed as

$$\begin{aligned}\mathrm{Rel}(G) &= p_{s1}q_{s2}q_{b1} \times \mathrm{Rel}(G_b) + q_{s1}p_{s2}q_{b2} \times \mathrm{Rel}(G_c) \\ &\quad + (p_{s1}q_{s2}p_{b1} + q_{s1}p_{s2}p_{b2} + p_{s1}p_{s2}) \times \mathrm{Rel}(G_d).\end{aligned} \tag{6}$$

Dividing equation (5) by a reduction factor F, we obtain

$$\frac{1}{F} \times \mathrm{Rel}\,(G_X) = \frac{p_1q_2}{F} \times \mathrm{Rel}(G_b) + \frac{q_1p_2}{F} \times \mathrm{Rel}(G_c) + \frac{p_1p_2}{F} \times \mathrm{Rel}(G_d). \tag{7}$$

Equating equations (6) and (7), we attain

$$\mathrm{Rel}\,(G_X) = \mathrm{Rel}(G) \times F \tag{8}$$

and

$$\frac{p_1q_2}{F} = p_{s1}q_{s2}q_{b1}, \quad \frac{q_1p_2}{F} = q_{s1}p_{s2}q_{b2}, \quad \text{and} \quad \frac{p_1p_2}{F} = p_{s1}q_{s2}p_{b1} + q_{s1}p_{s2}p_{b2} + p_{s1}p_{s2}. \tag{9}$$

Rearranging equation (9), we directly derive equations (1), (2), and (4) and thus, prove the theorem. ∎

The computational complexity of triangle reduction rests on the examination of the existence of triangle subgraphs and the transformation. Clearly, examining the existence of triangle subgraphs, namely an output-degree of the source of two and the adjacency of the source with two one-way or two-way connected nodes, only requires computational complexity of a constant time. With the closed formulas given in equations (1)–(4), the computational complexity of the transformation is apparent $O(1)$.

4. REDUCTION EFFICIENCY ANALYSIS FOR SIMPLIFIED GRID NETWORK

To exhibit the effectiveness of triangle reduction, we analyze the numbers of subproblems generated by PPR and CPR, with and without the triangle reduction axiom, for a *simplified grid network*. A network with $n+2$ nodes including s and t (numbered from 0 to $n+1$) is defined as an n-level simplified grid network, denoted by SG_n, if any three consecutive nodes of the network form a complete graph, as shown in Figure 6. For ease of description, the partition basis in PPR or CPR is selected in an increasing node number manner. The number of subproblems generated by PPR (CPR) for SG_n is denoted as NS_n^P (NS_n^C). In SG_n, the link incident from i to j is denoted as $e_{i,j}$. In addition, the PPR and CPR algorithms with triangle reduction applied are denoted as PPR$^+$ and CPR$^+$, respectively.

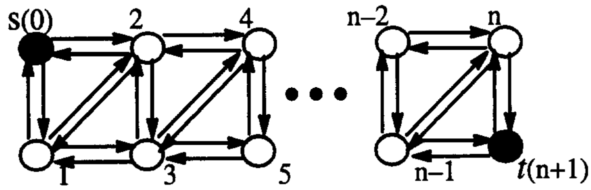


Figure 6. An n-level simplified grid network—SG_n.

LEMMA 1.

$$NS_n^P = \left(\frac{5+3\sqrt{5}}{10}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{5-3\sqrt{5}}{10}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n, \qquad \text{for } n \geq 2 \text{ and } NS_1^P = 1.$$

PROOF. According to reduction rules **r1**, **r5**, and **r6**, the fact that TR of SG_1 can be directly computed without any partition leads to $NS_n^P = 1$, for $n = 1$. Through simple derivation, one can simply get that $NS_2^P = 3$ and $NS_3^P = 5$. For $n \geq 4$, the derivation can be discussed in the following two cases, as illustrated in Figure 7.

CASE (a). n IS ODD. According to reduction rule **r1**, valueless links $e_{1,s}$, $e_{2,s}$, $e_{t,n-1}$, and $e_{t,n}$ can be immediately removed, leading to a new shortest $s-t$ path of SG_n, $s \to 2 \to 4 \to \cdots \to n-1 \to t$. Based on the path-based partition and factoring, $\text{Rel}(SG_n)$ is decomposed to $((n-1)/2+1)$ newly generated subproblems, namely $\text{Rel}(SG_n - e_{s,2})$, $\text{Rel}(SG_n * e_{s,2} - e_{2,4}), \ldots, \text{Rel}(SG_n * e_{s,2} * e_{2,4} * \cdots * e_{n-5,n-3} - e_{n-3,n-1})$, and $\text{Rel}(SG_n * e_{s,2} * e_{2,4} * \cdots * e_{n-3,n-1} - e_{n-1,t})$. According to reduction rules **r1**, **r3** to **r6**, the first $(n-1)/2$ subproblems can be reduced to lower-level simplified grid networks, as shown in Case (a) of Figure 7. The last subproblem $(SG_n * e_{s,2} * e_{2,4} * \cdots * e_{n-3,n-1} - e_{n-1,t})$ can be repeatedly reduced to the simplest network with only source and sink, resulting in the generation of one subproblem. Accordingly,

$$\begin{aligned} NS_n^P &= 1 + \left(\sum_{k=1}^{(n-1)/2} NS_{n-2k+1}^P + 1\right) \\ &= NS_{n-1}^P + NS_{n-2}^P, \qquad \text{for } n > 4. \end{aligned} \tag{10}$$

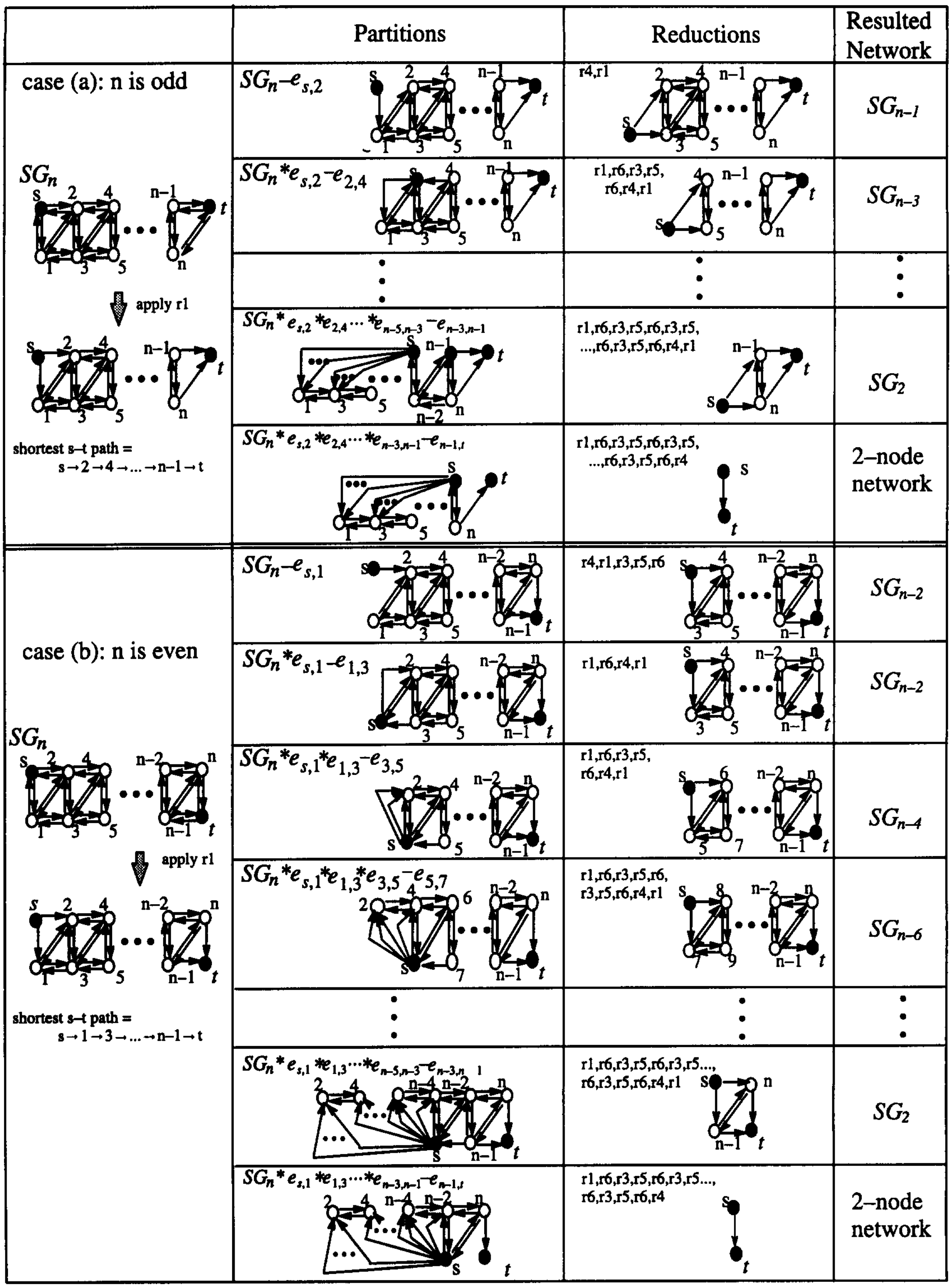


Figure 7. PPR algorithm for an n-level simplified grid network.

CASE (b). n IS EVEN. According to reduction rule **r1**, valueless links $e_{1,s}$, $e_{2,s}$, $e_{t,n-1}$, and $e_{t,n}$ can also be removed, leading to a shortest $s - t$ path of SG_n, $s \to 1 \to 3 \to \cdots \to n-1 \to t$. Based on the path-based partition and factoring, $\mathrm{Rel}(SG_n)$ is decomposed to $(n/2+1)$ newly generated subproblems, namely $\mathrm{Rel}(SG_n - e_{s,1}), \mathrm{Rel}(SG_n * e_{s,1} - e_{1,3}), \ldots, \mathrm{Rel}(SG_n * e_{s,1} * e_{1,3} * \cdots * e_{n-5,n-3} - e_{n-3,n-1})$, and $\mathrm{Rel}(SG_n * e_{s,1} * e_{1,3} * \cdots * e_{n-3,n-1} - e_{n-1,t})$. According to reduction rules **r1**, and **r3** to **r6**, the first $n/2$ subproblems can be reduced to $SG_{n-2}, SG_{n-2}, SG_{n-4}, \ldots$, and SG_2, respectively, as shown in Case (b) of Figure 7. The last subproblem $(SG_n * e_{s,1} * e_{1,3} *$

$\cdots * e_{n-3,n-1} - e_{n-1,t})$ can be repeatedly reduced to the simplest network with only source and sink, resulting in the generation of one subproblem. Accordingly,

$$\begin{aligned} NS_n^P &= 1 + \left(NS_{n-2}^P + \sum_{k=1}^{(n-2)/2} NS_{n-2k}^P + 1 \right) \\ &= NS_{n-1}^P + NS_{n-2}^P, \qquad \text{for } n > 4. \end{aligned} \tag{11}$$

From equations (10) and (11), we obtain the recurrence relation

$$NS_n^P = NS_{n-1}^P + NS_{n-2}^P, \qquad \text{for } n > 4. \tag{12}$$

Solving equation (12), the lemma can be directly proved. ∎

LEMMA 2.

$$NS_n^C = \left(\frac{5+\sqrt{5}}{5}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{5-\sqrt{5}}{5}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n - 1, \qquad \text{for } n \geq 1.$$

PROOF. Through simple derivation, one can get $NS_1^C = 1$ and $NS_2^C = 3$. For $n > 2$, according to reduction rule **r1**, valueless links $e_{1,s}$, $e_{2,s}$, $e_{t,n-1}$, and $e_{t,n}$ are removed, as shown in Figure 8. Based on the source-cut-based partition and factoring, $\mathrm{Rel}(SG_n)$ is further decomposed to two new subproblems, $\mathrm{Rel}(SG_n * e_{s,1})$ and $\mathrm{Rel}(SG_n - e_{s,1} * e_{s,2})$. The former can be reduced to SG_{n-1}, and the latter can be reduced to SG_{n-2}. Thus, we obtain

$$NS_n^C = NS_{n-1}^C + NS_{n-2}^C + 1, \qquad \text{for } n > 2. \tag{13}$$

Solving the equation, the lemma is directly proved. ∎

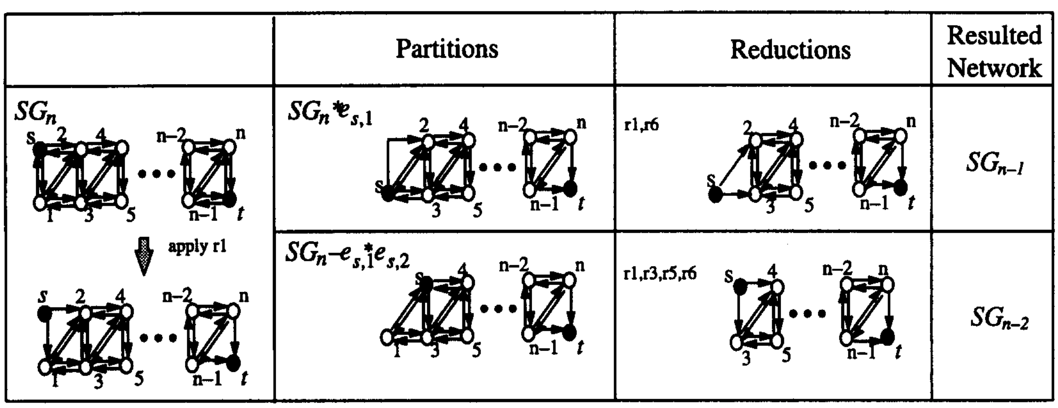


Figure 8. CPR algorithm for an n-level simplified grid network.

LEMMA 3. *With triangle reduction augmented, both PPR$^+$ and CPR$^+$ result in the generation of only one subproblem for SG_n.*

PROOF. According to reduction rule **r1**(a) and triangle reduction, links $e_{1,s}$, $e_{2,s}$, $e_{1,2}$, and $e_{2,1}$, are first removed, as shown in Figure 9. Through reduction and contraction, SG_n is further reduced to $SG_{n-1}, \ldots, SG_2$, and ultimately to the simplest network with only source and sink. ∎

THEOREM 4. *The reduction efficiency ratios of PPR to PPR$^+$ and CPR to CPR$^+$ are $O(((1+\sqrt{5})/2)^n)$, for SG_n, $n \geq 2$.*

PROOF. Based on Lemmas 1 and 3, the reduction efficiency ratio of PPR to PPR$^+$ is NS_n^P to one, for all $n \geq 2$. The reduction efficiency ratio of CPR to CPR$^+$, by Lemmas 2 and 3, is NS_n^C

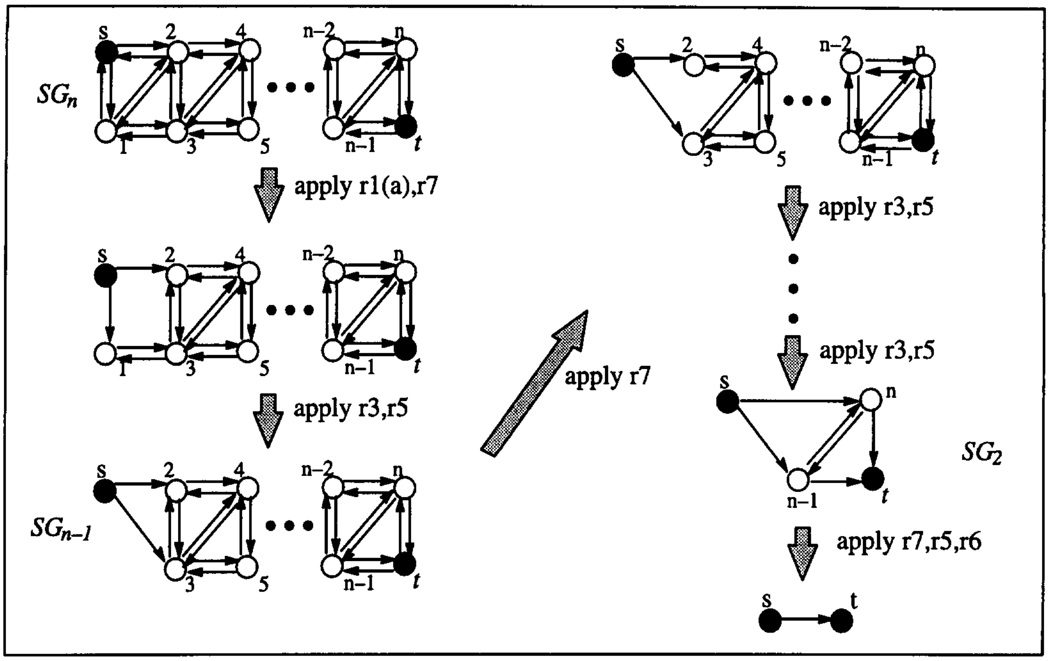


Figure 9. The reduction procedures of PPR$^+$ and CPR$^+$ for an n-level triangle network.

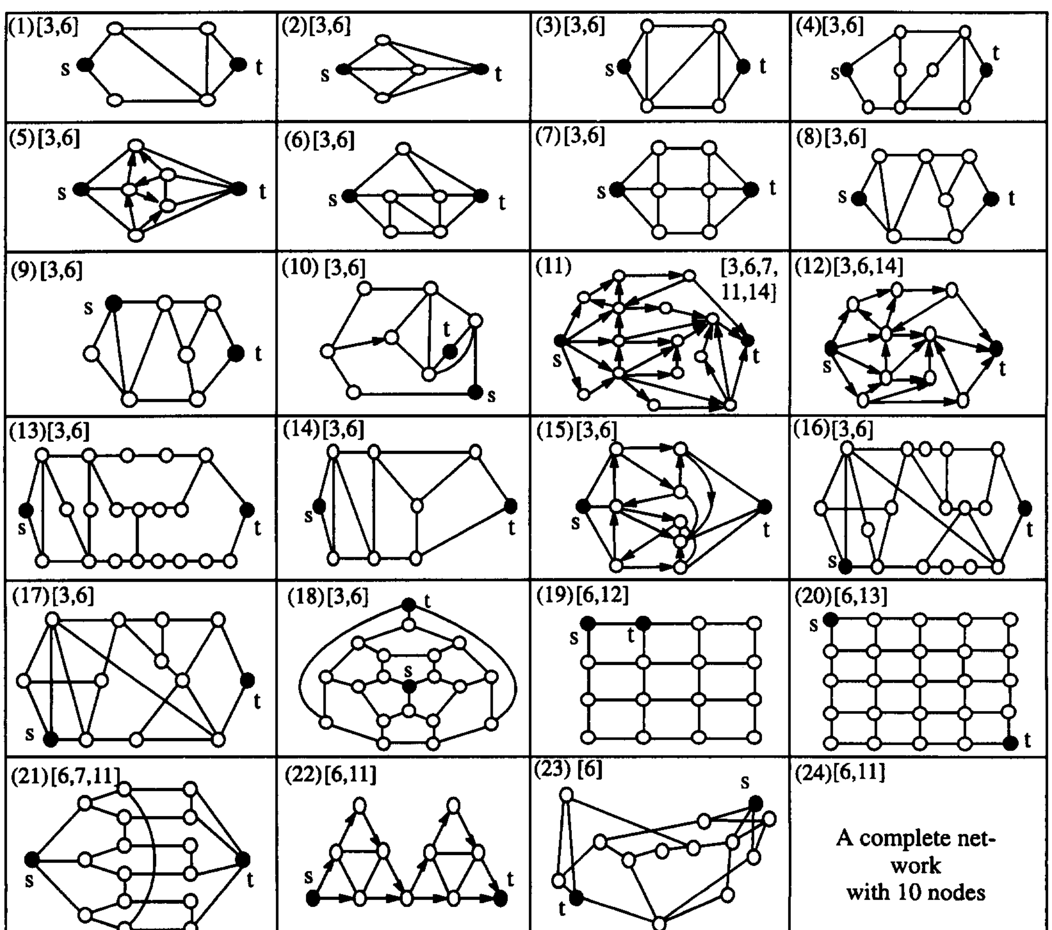


Figure 10. Benchmarks.

to one, for all $n \geq 1$. Thus, we attain

$$\begin{aligned} NS_n^P &= \left(\frac{5+3\sqrt{5}}{10}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{5-3\sqrt{5}}{10}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n \\ &\leq NS_n^C = \left(\frac{5+\sqrt{5}}{5}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{5-\sqrt{5}}{5}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n - 1 \qquad (14) \\ &= O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right), \qquad \text{for } n \geq 2. \end{aligned}$$

∎

5. PERFORMANCE COMPARISONS

To demonstrate the effectiveness of triangle reduction, we experimented on various networks using four algorithms, PPR, CPR, PPR$^+$, and CPR$^+$, which were implemented in C language and executed on Sun ServexStation 5. The experimented networks include the benchmarks [3,6,7,11–14], as summarized in Figure 10, and randomly generated networks with various link degrees. In all experiments, two performance metrics, the number of subproblems and computation time, have been observed.

Figures 11 and 12 show performance comparisons among these four algorithms under published benchmarks. In Figure 11, as was expected, the number of subproblems generated by either the PPR$^+$ or the CPR$^+$ algorithm is lower than that of both the PPR and CPR algorithms for all benchmarks. The performance superiority is particularly prominent under Benchmarks 1, 3, and 22, owing to the existence of higher numbers of triangle subgraphs. As for computation time, PPR$^+$ (CPR$^+$) also outperforms PPR (CPR) algorithm in all (most of the) benchmarks, as shown in Figure 12.

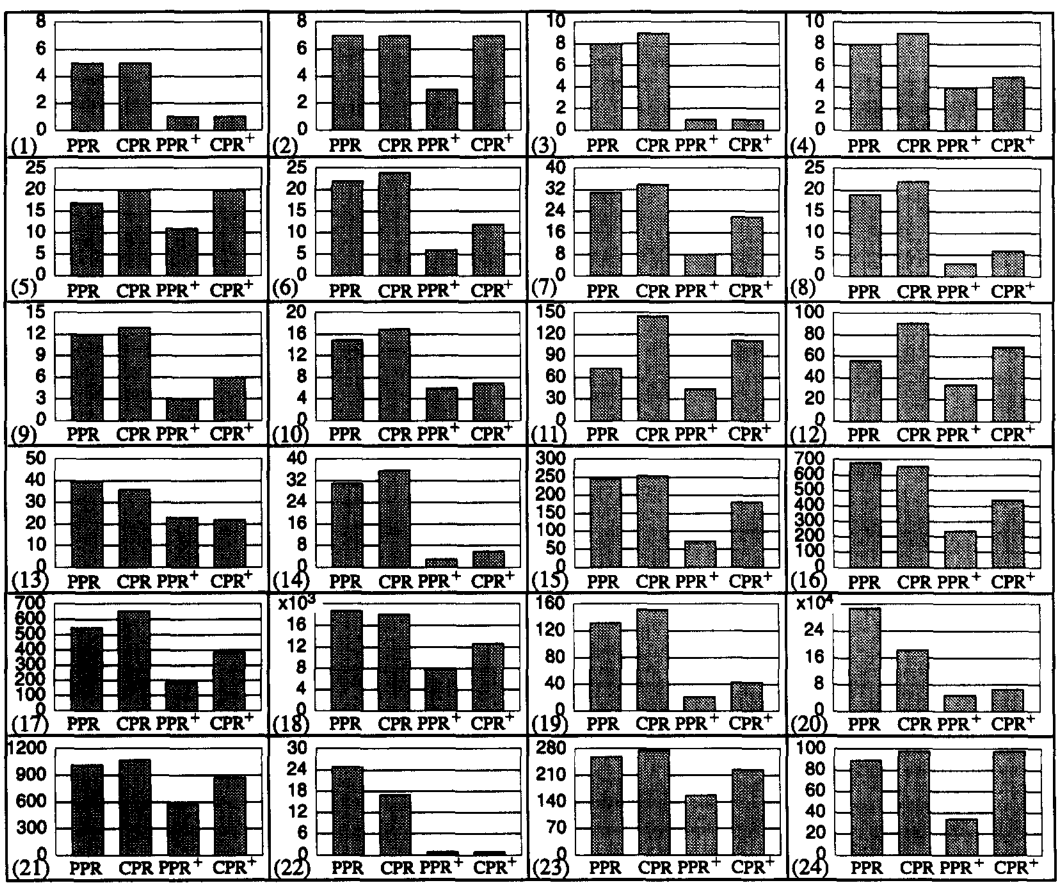


Figure 11. Comparisons of the number of subproblems under benchmarks.

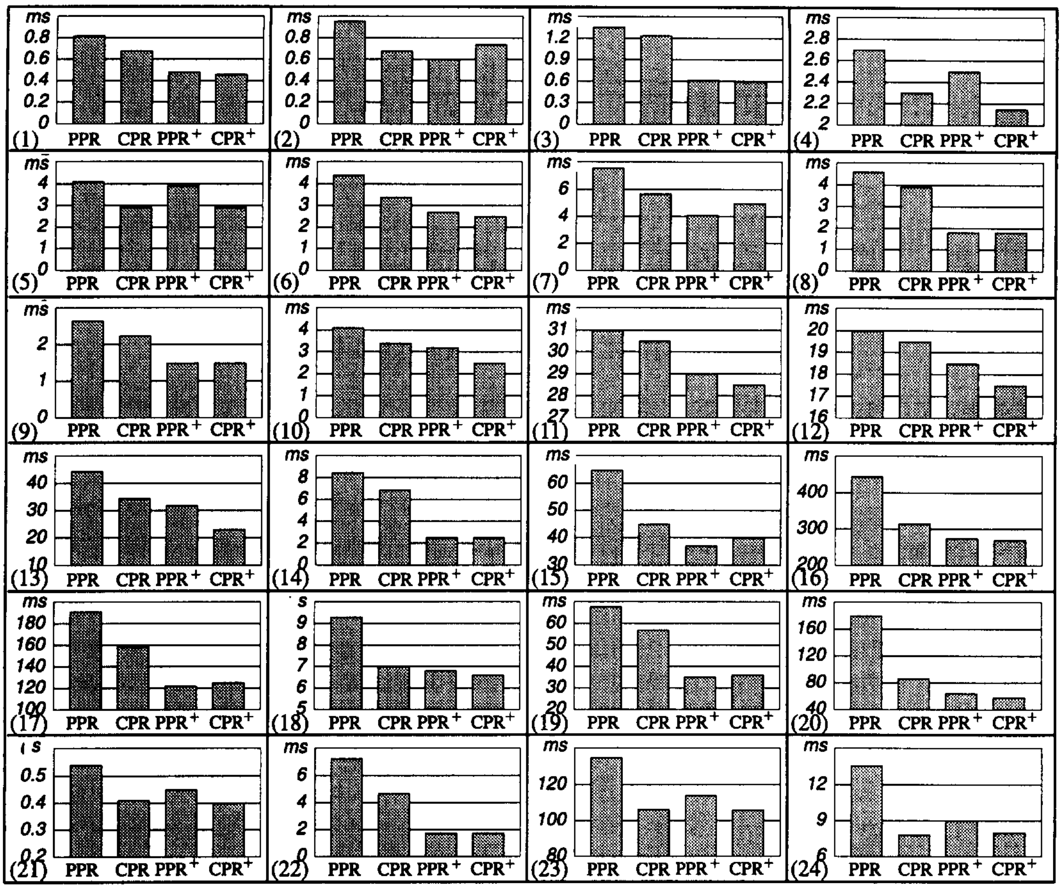
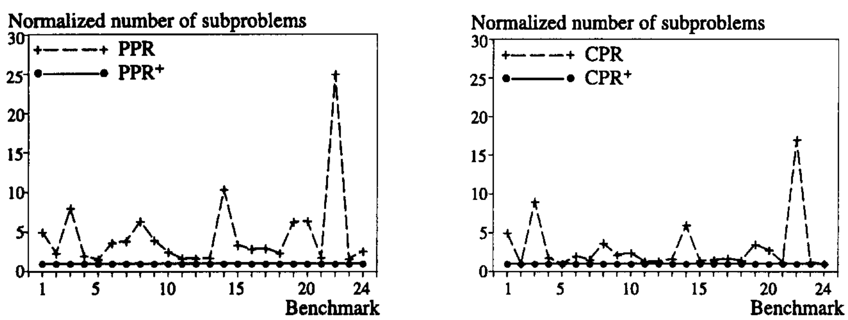
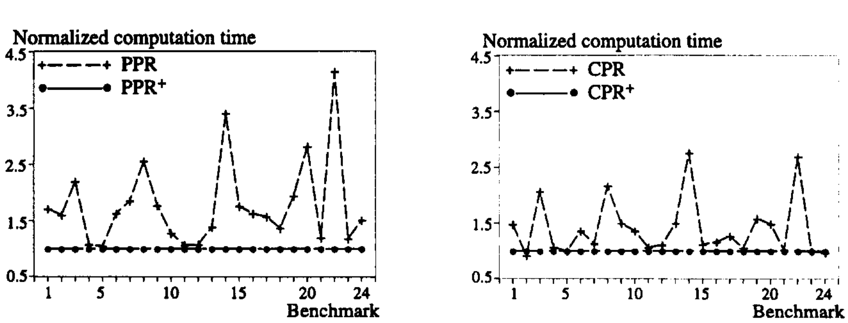


Figure 12. Comparisons of computation time under benchmarks.



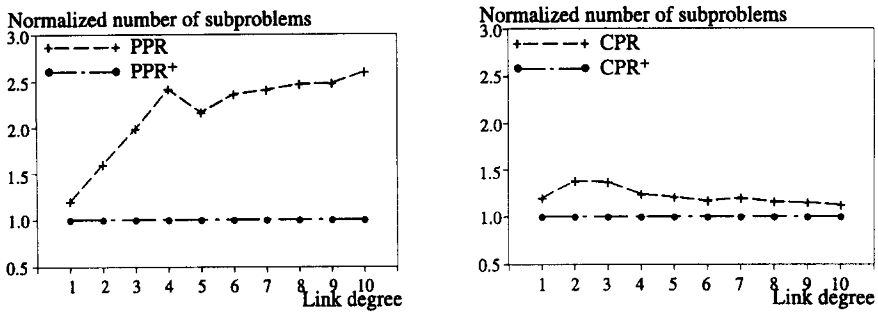
(a) Comparisons between PPR and PPR+. (b) Comparisons between CPR and CPR+.

Figure 13. Comparisons of the number of subproblems under benchmarks.



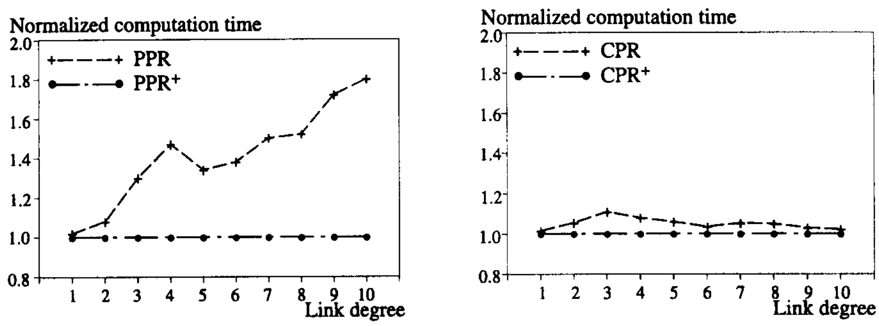
(a) Comparisons between PPR and PPR+. (b) Comparisons between CPR and CPR+.

Figure 14. Comparisons of computation time under benchmarks.



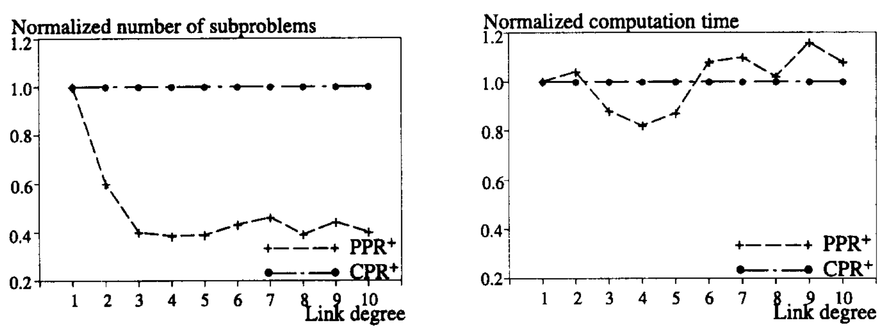
(a) Comparisons between PPR and PPR$^+$. (b) Comparisons between CPR and CPR$^+$.

Figure 15. Comparisons of the number of subproblems under randomly generated networks.



(a) Comparisons between PPR and PPR$^+$. (b) Comparisons between CPR and CPR$^+$.

Figure 16. Comparisons of computation time under randomly generated networks.



(a) Comparisons between PPR and PPR$^+$. (b) Comparisons between CPR and CPR$^+$.

Figure 17. Performance comparisons under randomly generated networks.

Figures 13 and 14 show the performance improvement of PPR$^+$/CPR$^+$ compared to PPR/CPR, under all benchmarks. In Figure 13a, the number of subproblems generated by PPR$^+$ is improved by a magnitude of four. As shown in Figure 13b, while the improvement ratio of CPR$^+$ to CPR is less significant than that of PPR$^+$ to PPR, CPR$^+$ still outperforms CPR by a magnitude of two. In Figure 14, we have observed that the contribution of the triangle reduction to the computation time is more significant in PPR$^+$ than in CPR$^+$ as well.

Figures 15 and 16 display the performance improvement of PPR$^+$ and CPR$^+$ under a set of randomly generated networks, from sparse to dense, with 15 nodes in each network. As shown in both figures, the improvement of PPR$^+$ in both performance metrics increases with the link degree of the network. In contrast, the improvement of CPR$^+$ is almost irrelevant to the link degree. By drawing direct comparisons between PPR$^+$ and CPR$^+$ in Figure 17, we have learned that, while PPR yields poorer performance [6] than CPR, PPR$^+$ with triangle reduction augmented achieves surprisingly better performance under sparse networks. As for denser networks, CPR$^+$ still outperforms PPR$^+$ due to its simplicity in determining the partition basis [6].

6. CONCLUSIONS

This paper proposed a triangle reduction which transforms a graph containing a triangle subgraph to that excluding the base of the triangle, with constant complexity. The paper also proved that both the reduction efficiency ratios of PPR to PPR$^+$ (i.e., NS_n^P to one) and CPR to CPR$^+$ (i.e., NS_n^C to one) are $O(((1+\sqrt{5})/2)^n)$, for simplified grid networks. The paper further provided an assessment of the effectiveness of triangle reduction on partition-based TR algorithms with respect to the number of subproblems and computation time through published benchmarks and randomly generated networks. Experimental results revealed that, PPR$^+$ and CPR$^+$ outperform PPR and CPR algorithms under most of the benchmarks and randomly generated networks. The improvement of PPR$^+$ in both performance metrics increases with the link degree of the network, while the improvement of CPR$^+$ is almost irrelevant to the link degree. In addition, even though PPR was shown in literature to exhibit much poorer performance than CPR, PPR$^+$ achieves surprisingly better performance under sparse networks.

REFERENCES

1. S. Rai, A. Kumar and E.V. Prasad, Computer terminal reliability of computer network, *Reliability Engineering* **16**, 109–119, (1986).
2. S. Rai and A. Kumar, Recursive technique for computing system reliability, *IEEE Trans. Reliability* **R-36**, 38–44, (April 1987).
3. S. Soh and S. Rai, CAREL: Computer aided reliability evaluator for distributed computing networks, *IEEE Trans. Parallel & Distributed Systems* **2**, 199–213, (April 1991).
4. V.A. Netes and B.P. Filin, Consideration of node failures in network-reliability calculation, *IEEE Trans. Reliability* **R-45**, 127–128, (March 1996).
5. W. Ke and S. Wang, Reliability evaluation for distributed computing networks with imperfect nodes, *IEEE Trans. Reliability* **R-46**, 342–349, (Sep. 1997).
6. Y.G. Chen and M.C. Yuang, A cut-based method for terminal-pair reliability, *IEEE Trans. Reliability* **R-45** (3), 413–416, (September 1996).
7. N. Deo and M. Medidi, Parallel algorithm for terminal-pair reliability, *IEEE Trans. Reliability* **R-41**, 201–209, (June 1992).
8. S. Hariri and C.S. Raghavendra, SYREL: A symbolic reliability algorithm based on path and cutset methods, *IEEE Trans. Computers* **C-36**, 1224–1232, (Oct. 1987).
9. S.J. Hsu and M.C. Yuang, Efficient computation of terminal-pair reliability using triangle reduction in network management, In *Proc. ICC*, pp. S8.5.1–S8.5.5, (1998).
10. M. Macgregor, W.D. Grover and U.M. Maydell, Connectability: A performance metric for reconfigurable transport networks, *IEEE J. Select. Areas Commun.* **11**, 1461–1469, (Dec. 1993).
11. L.B. Page and J.E. Perry, Reliability of directed networks using the factoring theorem, *IEEE Trans. Reliability* **R-38**, 556–562, (Dec. 1989).
12. D. Torrieri, An efficient algorithm for the calculation of node-pair reliability, *Proc. IEEE MILCOM 91*, 187–192, (Nov. 1991).
13. D. Torrieri, Calculation of node-pair reliability in large networks with unreliable nodes, *IEEE Trans. Reliability* **R-43**, 375–377, (Sept. 1994).
14. Y.B. Yoo and N. Deo, A comparison of algorithms for terminal-pair reliability, *IEEE Trans. Reliability* **R-37**, 210–215, (June 1988).